REGULARIZATION

In Machine Learning, we train our data and let it learn from it. When fitting our data points, our goal is to find the best fit when it can find all necessary patterns in our data and avoid random data points and unnecessary patterns called Noise. While training a machine learning model, the model can easily be overfitted or under fitted. To avoid this, we use regularization in machine learning to properly fit a model onto our test set.

Regularization: This is the technique that helps in avoiding overfitting and also it increasese model interpretability. This is a form of regression, that constrains / regularizes or shrinks the coefficient estimates towards zero. In addition, this technique discourages learning a more complex or flexible model, to avoid the risk of overfitting. Further, Regularization refers to techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting.

Overfitted Model: The model is not able to predict the output or target column for the unseen data by introducing noise in the output. When a Machine learning model tries to learn from the details along with the noise in the data and tries to fit each data point on the curve.

Underfitting model: When a Machine learning model can neither learn the relationship between variables in the testing data nor predict or classify a new data point.

"Noise" - this means those data points in the dataset which don't really represent the true properties of your data, but only due to a random chance.

Types of Regularization Techniques

1. Ridge Regression

Ridge Regression is also known as L-2 norm. It is one of the Regularization technique that is used to avoid overfitting or underfitting models by adding the penalty equivalent to the sum of the squares of the magnitude of coefficients. By changing the values of the penalty function, we are controlling the penalty term. The higher the penalty, it reduces the magnitude of coefficients. This technique shrinks the estimated coefficients towards zero. It seeks coefficient estimates that fit the data well, by making the RSS (Residual Sum of Squares) small.

1. Lasso Regression

Lasso Regression is also known as L-1 norm. It's a technique used to shrink all of the coefficients to exactly zero when the tuning parameter lambda is sufficiently large. Lasso Regression yields sparse models, modles that involve only a subset of the variables.

Application on Dataset

We will use dataset that describes First year students GPA. The data set contains information about 219 randomly selected first-year students at a midwestern college. This dataset contains variables such as GPA(first year GPA), HSGPA(high school GPA), SATV, SATM, Male(an indicator variable for Male vs Female), HU, SS, White(an indicator variable for white student vs non white), FirstGen(an indicator variable for first-generation students), and CollegeBound. We will be using GPA as our response vearlable and other variables as predictors. Our main goals are;

1. Standardize independent variables if necessary.
2. Pull off some of the data to use as training data and some to use as test data. Set up a grid for λ (but here we will use alpha instead).
3. Fit a ridge regression to the training data, then use cross-validation to choose the optimal value of λ. For this value of λ, calculate the test MSE on our test (validation) set. Record this test MSE, the value of λ, and the coefficients from the fitted ridge regression. Are any of the estimated coefficients 0? Is this what you would expect?
4. Repeat the previous bullet point for lasso regression.
5. Determine which variables are most important

Note: Lower RMSE and higher R-Squared values are indicative of a good model.

In [79]:
```
# Checking version for the numpy
import numpy as np
print(np.__version__)
```

```
1.21.5
```

In [80]:
```
# Checking version for pandas
import pandas as pd
print(pd.__version__)
```

```
1.3.5
```

In [81]:
```
# updating pandas and numpy
# !pip install pandas --upgrade
# !pip install numpy --upgrade
```

In [82]:
```python
# importing libraries
%matplotlib inline

import seaborn as sns
import matplotlib.pyplot as plt


from sklearn.model_selection import train_test_split
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.linear_model import LinearRegression
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.linear_model import RidgeCV
from sklearn.linear_model import LassoCV
from sklearn.model_selection import KFold
```

In [83]:
```python
# Loading dataset
GPA = pd.read_csv('FirstYearGPA.csv')
```

In [84]:
```python
# Looking at the first 5 column of the FirstYearGPA dataset
GPA.head()
```

Out[84]:

	GPA	HSGPA	SATV	SATM	Male	HU	SS	FirstGen	White	CollegeBound
0	3.06	3.83	680	770	1	3.0	9.0	1	1	1
1	4.15	4.00	740	720	0	9.0	3.0	0	1	1
2	3.41	3.70	640	570	0	16.0	13.0	0	0	1
3	3.21	3.51	740	700	0	22.0	0.0	0	1	1
4	3.48	3.83	610	610	0	30.5	1.5	0	1	1

In [85]:
```
# Looking at the dtype
GPA.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 219 entries, 0 to 218
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   GPA          219 non-null    float64
 1   HSGPA        219 non-null    float64
 2   SATV         219 non-null    int64  
 3   SATM         219 non-null    int64  
 4   Male         219 non-null    int64  
 5   HU           219 non-null    float64
 6   SS           219 non-null    float64
 7   FirstGen     219 non-null    int64  
 8   White        219 non-null    int64  
 9   CollegeBound 219 non-null    int64  
dtypes: float64(4), int64(6)
memory usage: 17.2 KB
```

In [86]:
```
# statistical summary
GPA.describe()
```

Out[86]:

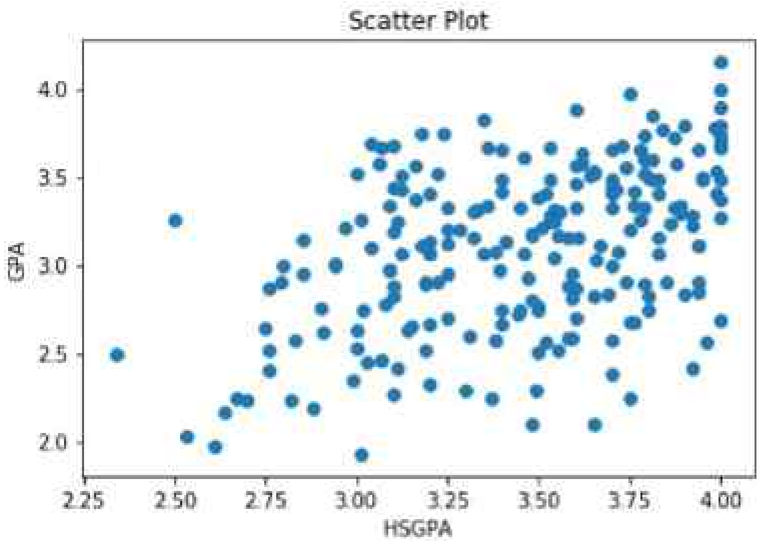
	GPA	HSGPA	SATV	SATM	Male	HU	SS	F
count	219.000000	219.000000	219.000000	219.000000	219.000000	219.000000	219.000000	219
mean	3.096164	3.452740	605.068493	634.292237	0.465753	13.108219	7.248858	0
std	0.465476	0.374794	83.393452	75.235572	0.499969	7.224647	5.000315	0
min	1.930000	2.340000	260.000000	430.000000	0.000000	0.000000	0.000000	0
25%	2.745000	3.170000	565.000000	580.000000	0.000000	8.000000	3.000000	0
50%	3.150000	3.500000	610.000000	640.000000	0.000000	13.000000	6.000000	0
75%	3.480000	3.760000	670.000000	690.000000	1.000000	17.000000	11.000000	0
max	4.150000	4.000000	740.000000	800.000000	1.000000	40.000000	21.000000	1

DATA VISUALIZATION

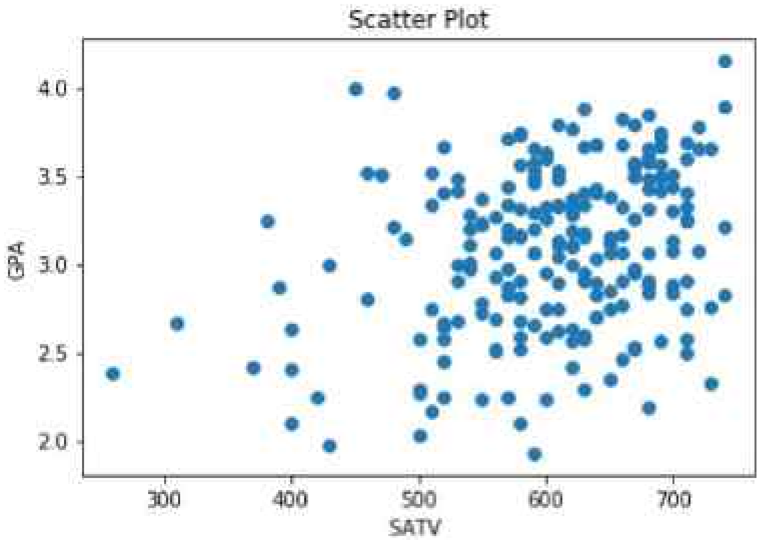
Data visualization helps us to see if there is any correlation between our predictor variables and the response variable. In our dataset, GPA is our response varibale and the rest are predictor variables. In our predictor variables, we can observe that some of them have dummy variables.

In [87]:

```
# Scatter plot with HSGPA against GPA
plt.scatter(GPA['HSGPA'], GPA['GPA'])

# Adding Title to the Plot
plt.title("Scatter Plot")

# Setting the X and Y labels
plt.xlabel('HSGPA')
plt.ylabel('GPA')

plt.show()
```

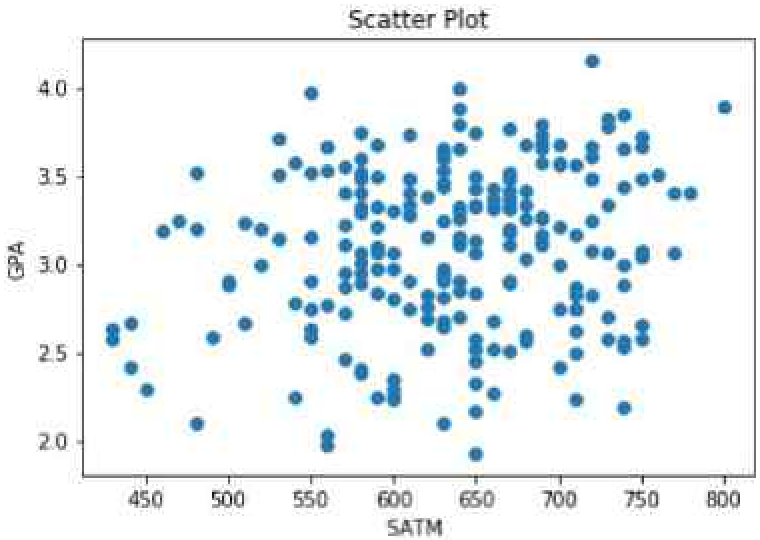


In [88]:
```python
# Scatter plot with SATV and GPA
plt.scatter(GPA['SATV'], GPA['GPA'])

# Adding Title to the Plot
plt.title("Scatter Plot")

# Setting the X and Y labels
plt.xlabel('SATV ')
plt.ylabel('GPA')

plt.show()
```

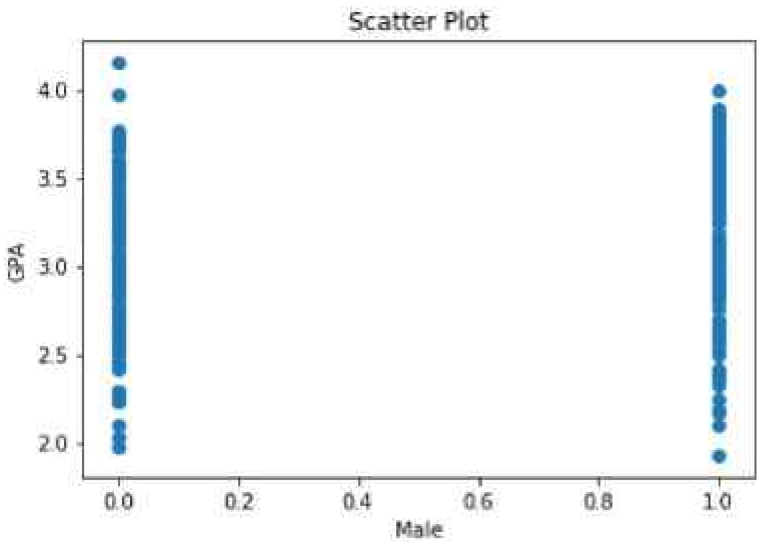


In [89]:

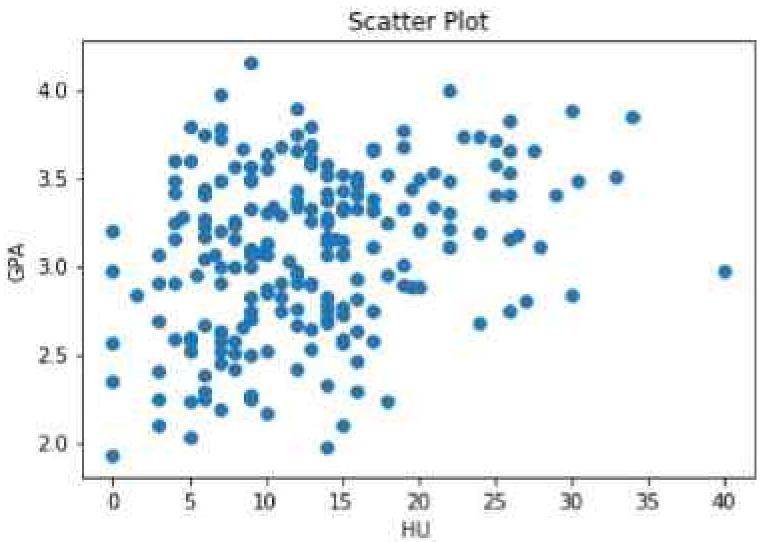
```python
# Scatter plot with SATM and GPA
plt.scatter(GPA['SATM'], GPA['GPA'])

# Adding Title to the Plot
plt.title("Scatter Plot")

# Setting the X and Y labels
plt.xlabel('SATM ')
plt.ylabel('GPA')

plt.show()
```



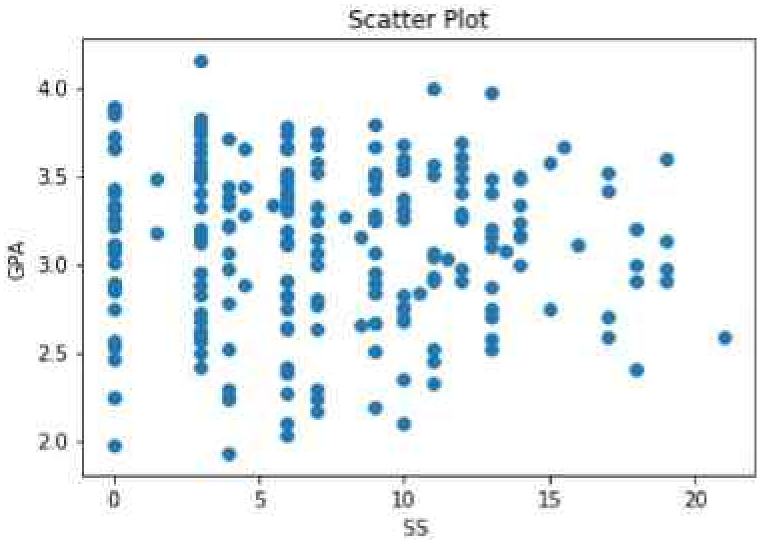
In [90]:
```
# Scatter plot with Male and GPA
plt.scatter(GPA['Male'], GPA['GPA'])

# Adding Title to the Plot
plt.title("Scatter Plot")

# Setting the X and Y labels
plt.xlabel('Male')
plt.ylabel('GPA')

plt.show()
```



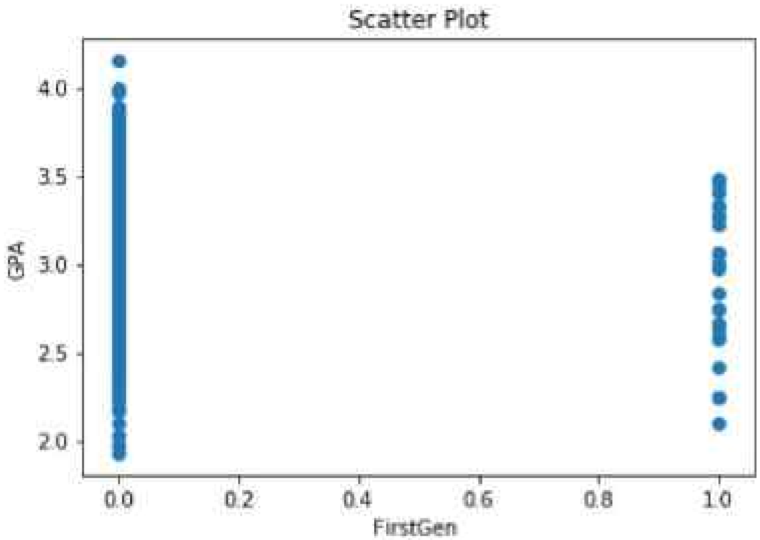
In [91]:
```
# Scatter plot with HU and GPA
plt.scatter(GPA['HU'], GPA['GPA'])

# Adding Title to the Plot
plt.title("Scatter Plot")

# Setting the X and Y labels
plt.xlabel('HU ')
plt.ylabel('GPA')

plt.show()
```



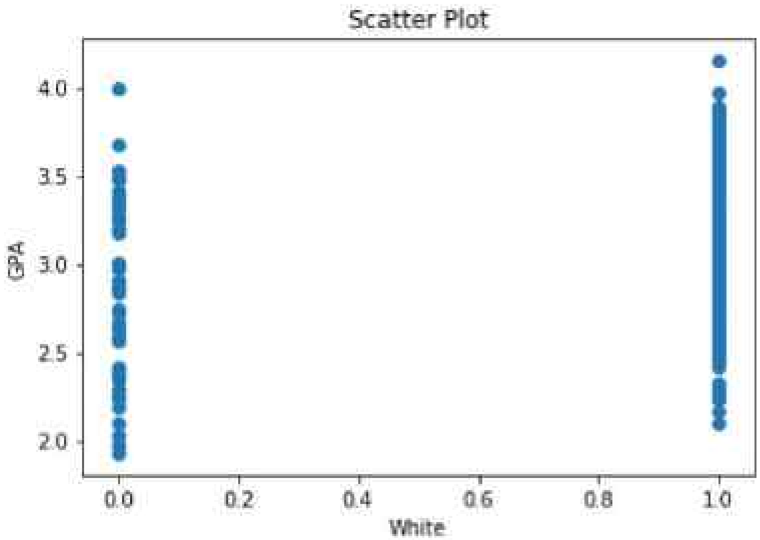
In [92]:
```
# Scatter plot with SS and GPA
plt.scatter(GPA['SS'], GPA['GPA'])

# Adding Title to the Plot
plt.title("Scatter Plot")

# Setting the X and Y labels
plt.xlabel('SS ')
plt.ylabel('GPA')

plt.show()
```



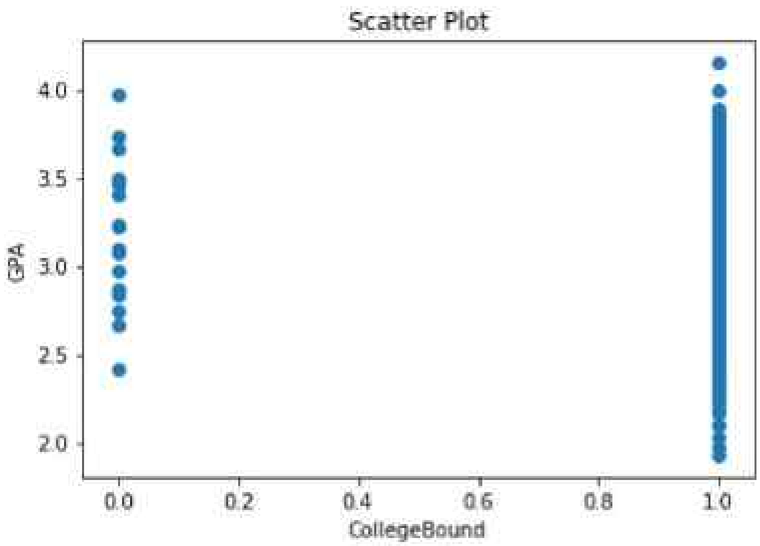
In [93]:
```
# Scatter plot with FirstGen and GPA
plt.scatter(GPA['FirstGen'], GPA['GPA'])

# Adding Title to the Plot
plt.title("Scatter Plot")

# Setting the X and Y labels
plt.xlabel('FirstGen ')
plt.ylabel('GPA')

plt.show()
```



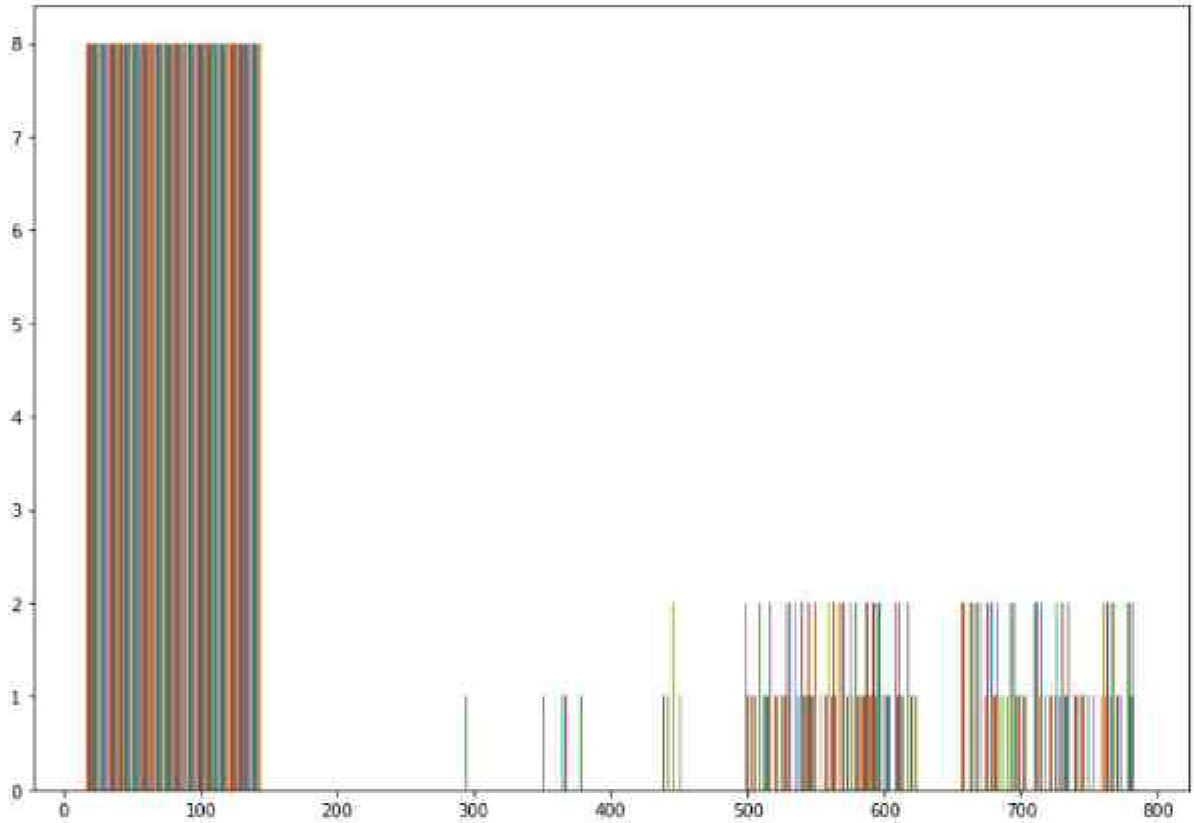
In [94]:
```
# Scatter plot with White and GPA
plt.scatter(GPA['White'], GPA['GPA'])

# Adding Title to the Plot
plt.title("Scatter Plot")

# Setting the X and Y labels
plt.xlabel('White')
plt.ylabel('GPA')

plt.show()
```



In [95]:
```
# Scatter plot with CollegeBound and GPA
plt.scatter(GPA['CollegeBound'], GPA['GPA'])

# Adding Title to the Plot
plt.title("Scatter Plot")

# Setting the X and Y labels
plt.xlabel('CollegeBound')
plt.ylabel('GPA')

plt.show()
```



In [96]:
```
# Creating histogram
fig, axs = plt.subplots(1, 1,
                        figsize =(10, 7),
                        tight_layout = True)

axs.hist(GPA, bins = 5)

# Show plot
plt.show()
```



Standardizing Variables

This data set might have variables of different scale. In order for our models to perform well, we would need to have our variables scaled to avoid bias in the outcome. By Standardizing we tend to make the mean of the dataset as 0 and the standard deviation to 1.

In [97]:
```
#scale the data to have mean 0 stdev 1
GPA = preprocessing.scale(GPA)
GPA = pd.DataFrame(GPA)
```

In [98]:
```
# Preparing dataset
#target_column = ['GPA']
#predictors = list(set(list(GPA.columns))-set(target_column))

# Normalizing by scaling the predictors between 0 and 1. This is done when the
units of the varibalrs differ significantly and may influence the modeling pro
cess.
#GPA[predictors] = GPA[predictors]/GPA[predictors].max()
```

We will build our model on the training set and evaluate its performance on the test set. This is called the holdout-validation approach for evaluating model performance.

X : independent variables (Other variables except GPA)

y : dependent variables (GPA)

The output shows that the shape of the training set has 153 observation of 9 variables and test set has 66 observation of 9 variables.

In [99]:
```
#List of predictors for Legend
predictors = list(GPA.columns.values)[1:10]

# Creating the training and Test datasets
X = GPA.iloc[:,1:10]
y = GPA.iloc[:,9]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, rand
om_state=40)

print(X_train.shape)
print(X_test.shape)

def MAPE(Y_actual,Y_Predicted):
    mape = np.mean(np.abs((Y_actual - Y_Predicted)/Y_actual))*100
    return mape
```

```
(153, 9)
(66, 9)
```

In [100]:
```
# Creating the training and Test datasets
#X = GPA[predictors].values
#y = GPA[target_column].values


#X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, ran
dom_state=40)

#print(X_train.shape)
#print(X_test.shape)

#def MAPE(Y_actual,Y_Predicted):
    #mape = np.mean(np.abs((Y_actual - Y_Predicted)/Y_actual))*100
    #return mape
```

Ridge regression is to slove the problems in the regression caused by multicollinearity. We have to check for multicollinearity first. Here we will check for high variance inflation factors (VIFs). The rule of thumb is that a VIF>10 indicates multicollinearity.

In [101]:
```
#VIF Calculations
###Calculate VIF Factors
vif=pd.DataFrame()
vif["VIF Factor"] = [variance_inflation_factor(X.values,i) for i in range(X.sh
ape[1])]
vif["features"]=X.columns

print (vif.round(2))
```

```
   VIF Factor  features
0        1.16         1
1        1.61         2
2        1.66         3
3        1.21         4
4        1.22         5
5        1.15         6
6        1.19         7
7        1.21         8
8        1.07         9
```

Linear Regression

The simplest form of regression is linear regression, which assumes that the predictors have a linear relationship with the target variable. The input variables are assumed to have a Gaussian distribution and are not correlated with each other (a problem called multi-collinearity).

The linear regression equation can be expressed in the following form: y = a_1x_1 + a_2x_2 + a_3x_3 + ... + a_x_n + b, where y is the target variable, x1, x2, x3, ... xn are the features, a1, a2, a3, ... an are the coefficients, and b is the parameter of the model.

The parameters a and b in the model are selected through the ordinary least squares (OLS) method. This method works by minimizing the sum of squares of residuals (actual value - predicted value).

Linear regression algorithm works by selecting coefficients for each independent variable that minimizes a loss function. However, if the coefficients are large, they can lead to over-fitting on the training dataset, and such a model will not generalize well on the unseen test data. To overcome this shortcoming, we will do regularization as shown below, which penalizes large coefficients.

In [102]:
```
# Multiple Linear regression model
lm = LinearRegression()
lm.fit(X_train, y_train)
```

Out[102]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=Fals
e)
```

In [103]:
```
# Genereating prediction on test set
lm_train_pred = lm.predict(X_train)
```

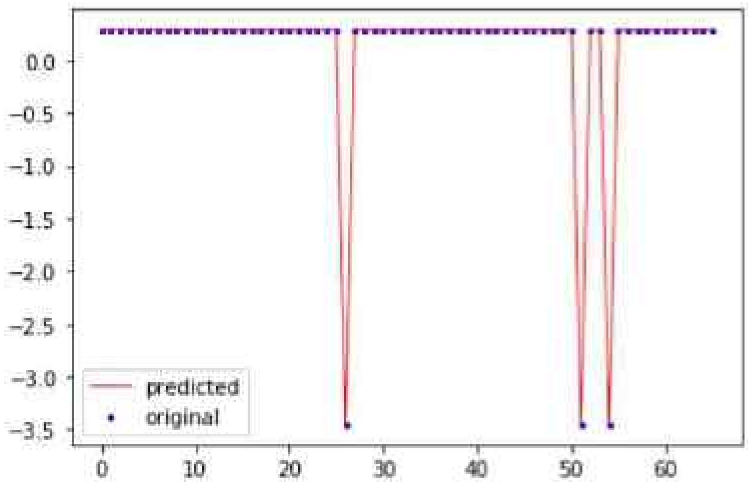
In [104]:
```
# prints the evaluation metrics (RMSE and R-squared) on the train set
print('RMSE of train set:            ',np.sqrt(mean_squared_error(y_train,lm_t
rain_pred)))
print('R-Squared of train set:       ',r2_score(y_train, lm_train_pred)*100,'%'
)

# prints the evaluation metrics (RMSE and R-squared) on the test set
lm_test_pred = lm.predict(X_test)
print('RMSE of est set:              ',np.sqrt(mean_squared_error(y_test,lm_te
st_pred)))
print('R-Squared of test set:        ',r2_score(y_test, lm_test_pred)*100,"%")

Lm_MAPE = MAPE(y_test,lm_test_pred)
print("MAPE value:                   ",Lm_MAPE)
Accuracy = 100 - Lm_MAPE
print('Accuracy of Linear Regression: {:0.2f}%.'.format(Accuracy))
```

```
RMSE of train set:             1.4049714454594176e-15
R-Squared of train set:       100.0 %
RMSE of est set:               1.543127709506172e-15
R-Squared of test set:        100.0 %
MAPE value:                   4.082448508711048e-13
Accuracy of Linear Regression: 100.00%.
```

In [105]:
```
# retrieving the intercept and the coefficients
print(lm.intercept_,lm.coef_)
```

```
6.938893903907228e-17 [ 1.90708999e-16 -4.99600361e-16  8.88178420e-16  3.885
78059e-16
 -9.15933995e-16 -3.22658567e-16 -2.49800181e-16  4.92661467e-16
  1.00000000e+00]
```

In [106]:
```
x_ax = range(len(X_test))
plt.scatter(x_ax, y_test, s=5, color="blue", label="original")
plt.plot(x_ax, lm_test_pred, lw=0.8, color="red", label="predicted")
plt.legend()
plt.show()
```



RIDGE REGRESSION

Ridge Regression is a model that analyses any data that seems to have multicollinearity. In ridge regression, we have a term lambda. This term is denoted by an alpha parameter in the ridge function. When the values of alpha changes then the penalty term is controlled. By using FirstYearGPA dataset, our y value will be the GPA of the student (response variable) and x will be the predictor variables. We expect coefficients that do not contribute to the model to be shrinked by letting them have smaller value.

The main goal is to obtain the mean squared error and observe if the estimated coefficients are 0. The mean Square error will show us the model error; thus, we want our MSE to be as small as possible. Also,we will calculate R-squared which measures the goodness of fit. R-squared measures the strength of the relationship between Ridge Regression model and the predictor variables.

R-squared of 0% represents a model that does not explain any of the variation in the response variable around its mean. The mean of the dependent variable predicts the dependent variable as well as the regression model. While, R-squared of 100% represents a model that explains all the variation in the response variable around its mean. For R-square we would want our result to have higher value.

Here we used different alpha values to check and see which model would give us a smaller RMSE and higher R-squared. Also, our models were normalized to have mean zero to avoid regularization biases of the model away from the data. In addition, we checked for the accuracy of the model.

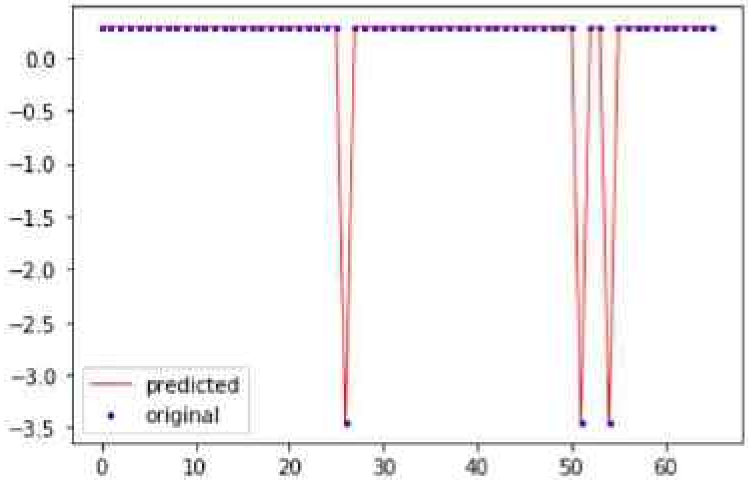
In [107]:
```
# Ridge regression model with an alpha value of 0.01
rr1 = Ridge(alpha=0.01,solver="cholesky")

# fitting the model to the training data
rr1.fit(X_train, y_train)

# predicting
pred_train_rr1= rr1.predict(X_train)

# prints the evaluation metrics (RMSE and R-squared) on the train set
print('RMSE of train set:        ', np.sqrt(mean_squared_error(y_train,pred_tra
in_rr1)))
print('R-Squared of train set: ',r2_score(y_train, pred_train_rr1)*100,'%')

# prints the evaluation metrics (RMSE and R-squared) on the test set
pred_test_rr1= rr1.predict(X_test)
print('RMSE of test set:         ',np.sqrt(mean_squared_error(y_test,pred_test_
rr1)))
print('R-Squared of test set:  ',r2_score(y_test, pred_test_rr1)*100,'%')

Ridge_MAPE = MAPE(y_test,pred_test_rr1)
print("MAPE value:              ",Ridge_MAPE)
Accuracy = 100 - Ridge_MAPE
print('Accuracy of Ridge Regression: {:0.2f}%.'.format(Accuracy))
```

```
RMSE of train set:         6.356818603729847e-05
R-Squared of train set:  99.99999965195795 %
RMSE of test set:          5.167067629930698e-05
R-Squared of test set:   99.99999955941834 %
MAPE value:              0.006268135906189343
Accuracy of Ridge Regression: 99.99%.
```

In [108]:
```
# retrieving the intercept and the coefficients
print(rr1.intercept_,rr1.coef_)
```

```
-1.2820771127103336e-06 [-1.21078606e-05  8.45213588e-06  4.00349536e-06  5.6
4408032e-06
  2.62094123e-06 -2.53791915e-06 -5.91359896e-06 -8.63077311e-06
  9.99938170e-01]
```

In [109]:
```
x_ax = range(len(X_test))
plt.scatter(x_ax, y_test, s=5, color="blue", label="original")
plt.plot(x_ax, pred_test_rr1, lw=0.8, color="red", label="predicted")
plt.legend()
plt.show()
```



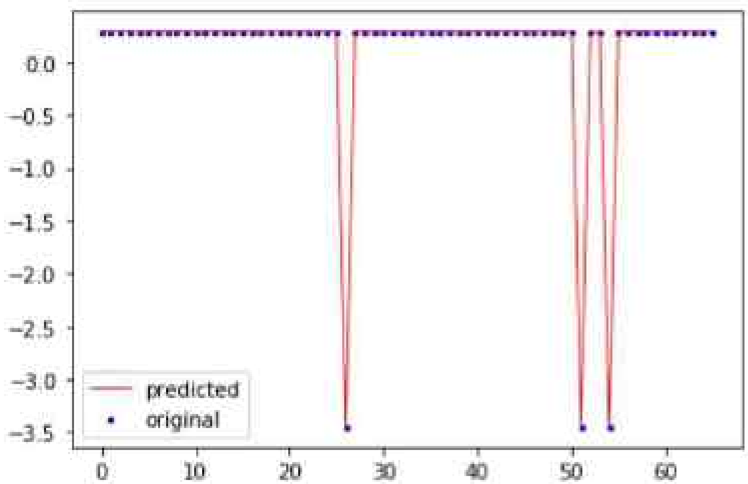
In [110]:
```
# Ridge regression model with an alpha value of 1
rr3 = Ridge(alpha=1)

# fitting the model to the training data
rr3.fit(X_train, y_train)

# predicting
pred_train_rr3= rr3.predict(X_train)

# prints the evaluation metrics (RMSE and R-squared) on the train set
print('RMSE of train set:          ',np.sqrt(mean_squared_error(y_train,pred_t
rain_rr3)))
print('R-Squared of train set:    ',r2_score(y_train, pred_train_rr3)*100,'%')

# prints the evaluation metrics (RMSE and R-squared) on the test set
pred_test_rr3= rr3.predict(X_test)
print('RMSE of test set:           ',np.sqrt(mean_squared_error(y_test,pred_te
st_rr3)))
print('R-Squared of test set:     ',r2_score(y_test, pred_test_rr3)*100,'%')

Ridge_MAPE = MAPE(y_test,pred_test_rr3)
print("MAPE value:                 ",Ridge_MAPE)
Accuracy = 100 - Ridge_MAPE
print('Accuracy of Ridge Regression: {:0.2f}%.'.format(Accuracy))
```

```
RMSE of train set:           0.006314283670278182
R-Squared of train set:     99.99656600020937 %
RMSE of test set:            0.005128883098501061
R-Squared of test set:      99.99565906054228 %
MAPE value:                 0.6212457410574677
Accuracy of Ridge Regression: 99.38%.
```

In [111]:
```
# retrieving the intercept and the coefficients
print(rr3.intercept_,rr3.coef_)
```

```
-0.00012843137528020798 [-1.19383640e-03  8.29953903e-04  3.96936720e-04  5.5
9200670e-04
  2.57021487e-04 -2.52028605e-04 -5.86464759e-04 -8.47531918e-04
  9.93858367e-01]
```

In [112]:
```
x_ax = range(len(X_test))
plt.scatter(x_ax, y_test, s=5, color="blue", label="original")
plt.plot(x_ax, pred_test_rr3, lw=0.8, color="red", label="predicted")
plt.legend()
plt.show()
```



Looking at the reults of estimated coefficients, we can see that, as we increase the value of alpha, the magnitude of the coefficients decreases, where the values reaches to zero but not absolute zero. On the other hand, if we look at the results of the R-Squared, we see that the value of R-Squared is also at higher value for an increased alpha of 1. In addition, when alpha was increases RSME increases. This is not what we want in our model. By changing the values of alpha, we are basically controlling the penalty term. The higher the values of alpha, bigger is the penalty and therefore the magnitude of coefficients are reduced. Thus, Ridge Regression model with alpha 0.01 performs better than the one with alpha 1. However, this doesn't mean that it is the best model overall. Since we want our estimated coefficients to go to zero, we will look into corss validation of 10 folds and Lasso Regression to see if they will performs better than Ridge Regression by sending estimated coefficients to zero and showing us the most important variables.

In [131]:

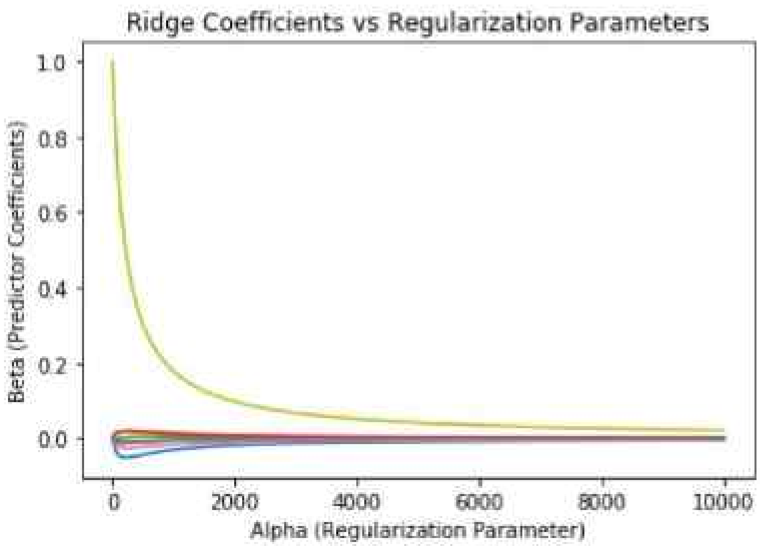
```
alphas = range(0,100)
for a in alphas:
 model = Ridge(alpha=a, normalize=True).fit(X,y)
 score = model.score(X, y)
 pred_y = model.predict(X)
 mse = mean_squared_error(y, pred_y)
 print("Alpha:{0:.6f}, R2:{1:.3f}, MSE:{2:.2f}, RMSE:{3:.2f}"
    .format(a, score, mse, np.sqrt(mse)))
```

```
Alpha:0.000000, R2:1.000, MSE:0.00, RMSE:0.00
Alpha:1.000000, R2:0.754, MSE:0.25, RMSE:0.50
Alpha:2.000000, R2:0.566, MSE:0.43, RMSE:0.66
Alpha:3.000000, R2:0.450, MSE:0.55, RMSE:0.74
Alpha:4.000000, R2:0.372, MSE:0.63, RMSE:0.79
Alpha:5.000000, R2:0.318, MSE:0.68, RMSE:0.83
Alpha:6.000000, R2:0.277, MSE:0.72, RMSE:0.85
Alpha:7.000000, R2:0.245, MSE:0.75, RMSE:0.87
Alpha:8.000000, R2:0.220, MSE:0.78, RMSE:0.88
Alpha:9.000000, R2:0.200, MSE:0.80, RMSE:0.89
Alpha:10.000000, R2:0.183, MSE:0.82, RMSE:0.90
Alpha:11.000000, R2:0.168, MSE:0.83, RMSE:0.91
Alpha:12.000000, R2:0.156, MSE:0.84, RMSE:0.92
Alpha:13.000000, R2:0.145, MSE:0.85, RMSE:0.92
Alpha:14.000000, R2:0.136, MSE:0.86, RMSE:0.93
Alpha:15.000000, R2:0.128, MSE:0.87, RMSE:0.93
Alpha:16.000000, R2:0.121, MSE:0.88, RMSE:0.94
Alpha:17.000000, R2:0.114, MSE:0.89, RMSE:0.94
Alpha:18.000000, R2:0.109, MSE:0.89, RMSE:0.94
Alpha:19.000000, R2:0.103, MSE:0.90, RMSE:0.95
Alpha:20.000000, R2:0.099, MSE:0.90, RMSE:0.95
Alpha:21.000000, R2:0.094, MSE:0.91, RMSE:0.95
Alpha:22.000000, R2:0.090, MSE:0.91, RMSE:0.95
Alpha:23.000000, R2:0.087, MSE:0.91, RMSE:0.96
Alpha:24.000000, R2:0.083, MSE:0.92, RMSE:0.96
Alpha:25.000000, R2:0.080, MSE:0.92, RMSE:0.96
Alpha:26.000000, R2:0.077, MSE:0.92, RMSE:0.96
Alpha:27.000000, R2:0.075, MSE:0.93, RMSE:0.96
Alpha:28.000000, R2:0.072, MSE:0.93, RMSE:0.96
Alpha:29.000000, R2:0.070, MSE:0.93, RMSE:0.96
Alpha:30.000000, R2:0.068, MSE:0.93, RMSE:0.97
Alpha:31.000000, R2:0.065, MSE:0.93, RMSE:0.97
Alpha:32.000000, R2:0.063, MSE:0.94, RMSE:0.97
Alpha:33.000000, R2:0.062, MSE:0.94, RMSE:0.97
Alpha:34.000000, R2:0.060, MSE:0.94, RMSE:0.97
Alpha:35.000000, R2:0.058, MSE:0.94, RMSE:0.97
Alpha:36.000000, R2:0.057, MSE:0.94, RMSE:0.97
Alpha:37.000000, R2:0.055, MSE:0.94, RMSE:0.97
Alpha:38.000000, R2:0.054, MSE:0.95, RMSE:0.97
Alpha:39.000000, R2:0.053, MSE:0.95, RMSE:0.97
Alpha:40.000000, R2:0.051, MSE:0.95, RMSE:0.97
Alpha:41.000000, R2:0.050, MSE:0.95, RMSE:0.97
Alpha:42.000000, R2:0.049, MSE:0.95, RMSE:0.98
Alpha:43.000000, R2:0.048, MSE:0.95, RMSE:0.98
Alpha:44.000000, R2:0.047, MSE:0.95, RMSE:0.98
Alpha:45.000000, R2:0.046, MSE:0.95, RMSE:0.98
Alpha:46.000000, R2:0.045, MSE:0.96, RMSE:0.98
Alpha:47.000000, R2:0.044, MSE:0.96, RMSE:0.98
Alpha:48.000000, R2:0.043, MSE:0.96, RMSE:0.98
Alpha:49.000000, R2:0.042, MSE:0.96, RMSE:0.98
Alpha:50.000000, R2:0.041, MSE:0.96, RMSE:0.98
Alpha:51.000000, R2:0.041, MSE:0.96, RMSE:0.98
Alpha:52.000000, R2:0.040, MSE:0.96, RMSE:0.98
Alpha:53.000000, R2:0.039, MSE:0.96, RMSE:0.98
Alpha:54.000000, R2:0.038, MSE:0.96, RMSE:0.98
Alpha:55.000000, R2:0.038, MSE:0.96, RMSE:0.98
Alpha:56.000000, R2:0.037, MSE:0.96, RMSE:0.98
```

```
Alpha:57.000000, R2:0.036, MSE:0.96, RMSE:0.98
Alpha:58.000000, R2:0.036, MSE:0.96, RMSE:0.98
Alpha:59.000000, R2:0.035, MSE:0.96, RMSE:0.98
Alpha:60.000000, R2:0.035, MSE:0.97, RMSE:0.98
Alpha:61.000000, R2:0.034, MSE:0.97, RMSE:0.98
Alpha:62.000000, R2:0.034, MSE:0.97, RMSE:0.98
Alpha:63.000000, R2:0.033, MSE:0.97, RMSE:0.98
Alpha:64.000000, R2:0.033, MSE:0.97, RMSE:0.98
Alpha:65.000000, R2:0.032, MSE:0.97, RMSE:0.98
Alpha:66.000000, R2:0.032, MSE:0.97, RMSE:0.98
Alpha:67.000000, R2:0.031, MSE:0.97, RMSE:0.98
Alpha:68.000000, R2:0.031, MSE:0.97, RMSE:0.98
Alpha:69.000000, R2:0.030, MSE:0.97, RMSE:0.98
Alpha:70.000000, R2:0.030, MSE:0.97, RMSE:0.98
Alpha:71.000000, R2:0.029, MSE:0.97, RMSE:0.99
Alpha:72.000000, R2:0.029, MSE:0.97, RMSE:0.99
Alpha:73.000000, R2:0.029, MSE:0.97, RMSE:0.99
Alpha:74.000000, R2:0.028, MSE:0.97, RMSE:0.99
Alpha:75.000000, R2:0.028, MSE:0.97, RMSE:0.99
Alpha:76.000000, R2:0.028, MSE:0.97, RMSE:0.99
Alpha:77.000000, R2:0.027, MSE:0.97, RMSE:0.99
Alpha:78.000000, R2:0.027, MSE:0.97, RMSE:0.99
Alpha:79.000000, R2:0.027, MSE:0.97, RMSE:0.99
Alpha:80.000000, R2:0.026, MSE:0.97, RMSE:0.99
Alpha:81.000000, R2:0.026, MSE:0.97, RMSE:0.99
Alpha:82.000000, R2:0.026, MSE:0.97, RMSE:0.99
Alpha:83.000000, R2:0.025, MSE:0.97, RMSE:0.99
Alpha:84.000000, R2:0.025, MSE:0.98, RMSE:0.99
Alpha:85.000000, R2:0.025, MSE:0.98, RMSE:0.99
Alpha:86.000000, R2:0.024, MSE:0.98, RMSE:0.99
Alpha:87.000000, R2:0.024, MSE:0.98, RMSE:0.99
Alpha:88.000000, R2:0.024, MSE:0.98, RMSE:0.99
Alpha:89.000000, R2:0.024, MSE:0.98, RMSE:0.99
Alpha:90.000000, R2:0.023, MSE:0.98, RMSE:0.99
Alpha:91.000000, R2:0.023, MSE:0.98, RMSE:0.99
Alpha:92.000000, R2:0.023, MSE:0.98, RMSE:0.99
Alpha:93.000000, R2:0.023, MSE:0.98, RMSE:0.99
Alpha:94.000000, R2:0.022, MSE:0.98, RMSE:0.99
Alpha:95.000000, R2:0.022, MSE:0.98, RMSE:0.99
Alpha:96.000000, R2:0.022, MSE:0.98, RMSE:0.99
Alpha:97.000000, R2:0.022, MSE:0.98, RMSE:0.99
Alpha:98.000000, R2:0.021, MSE:0.98, RMSE:0.99
Alpha:99.000000, R2:0.021, MSE:0.98, RMSE:0.99
```

In [114]:
```
###initialize list to store coefficient values
coef=[]
alphas = range(0,10000)

for a in alphas:
  ridgereg=Ridge(alpha=a,solver="cholesky")
  ridgereg.fit(X,y)
  coef.append(ridgereg.coef_)

###Make plot of Beta as a function of Alpha
fig=plt.figure()
ax=fig.add_subplot(111)
ax.plot(alphas,coef)
ax.set_xlabel('Alpha (Regularization Parameter)')
ax.set_ylabel('Beta (Predictor Coefficients)')
ax.set_title('Ridge Coefficients vs Regularization Parameters')
ax.axis('tight')
###ax.legend(loc='best')
#fig.savefig('coef_vs_alpha.png')
```

Out[114]:
```
(-499.95000000000005, 10498.95, -0.10322114230825868, 1.0525343401099163)
```



From the plot as alpha increases the coefficients convert to smaller values of their original. This is the power of ridge regression, making the coefficients smaller to limit the collinearity between predictors.

USING CROSS VALIDATION IN RIDGE REGRESSION

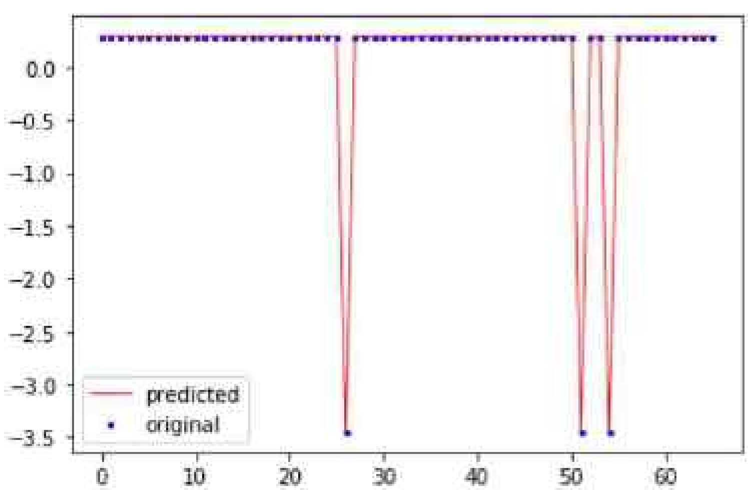
In [137]:
```
###Selecting lambda
scaler=StandardScaler()
X_std=scaler.fit_transform(X)

###Fit Ridge regression through cross validation
regr_cv=RidgeCV(alphas=range(1,40),store_cv_values=True)
model_cv=regr_cv.fit(X,y)

print (model_cv.alpha_)

print(np.mean(model_cv.cv_values_, axis=0))
```

```
1
[2.71568602e-05 1.07386978e-04 2.38880665e-04 4.19893874e-04
 6.48745377e-04 9.23814080e-04 1.24353648e-03 1.60640427e-03
 2.01096201e-03 2.45580500e-03 2.93957716e-03 3.46096909e-03
 4.01871620e-03 4.61159689e-03 5.23843091e-03 5.89807768e-03
 6.58943476e-03 7.31143641e-03 8.06305214e-03 8.84328540e-03
 9.65117226e-03 1.04857802e-02 1.13462071e-02 1.22315796e-02
 1.31410528e-02 1.40738686e-02 1.50290549e-02 1.60060250e-02
 1.70039760e-02 1.80221887e-02 1.90599662e-02 2.01166334e-02
 2.11915361e-02 2.22840404e-02 2.33935319e-02 2.45194150e-02
 2.56611125e-02 2.68180646e-02 2.79897286e-02]
```

In [116]:
```
ypred = model_cv.predict(X_test)
score = model_cv.score(X_test,y_test)
mse = mean_squared_error(y_test,ypred)
print("R2:{0:.3f}, MSE:{1:.2f}, RMSE:{2:.2f}"
    .format(score, mse, np.sqrt(mse)))
```

```
R2:1.000, MSE:0.00, RMSE:0.00
```

In [133]:
```
x_ax = range(len(X_test))
plt.scatter(x_ax, y_test, s=5, color="blue", label="original")
plt.plot(x_ax, ypred, lw=0.8, color="red", label="predicted")
plt.legend()
plt.show()
```



In [136]:

```
w = list()
for a in alphas:
    ridge_clf = RidgeCV(alphas=[a],cv=10).fit(X, y)
    w.append(ridge_clf.coef_)
w = np.array(w)
plt.semilogx(alphas,w)
plt.title('Ridge coefficients as function of the regularization')
plt.xlabel('alpha')
plt.ylabel('weights')
plt.legend(X.keys())
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
any behavior and is safe. When replacing `np.int`, you may wish to use e.g. `
np.int64` or `np.int32` to specify the precision. If you wish to review your
current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
any behavior and is safe. When replacing `np.int`, you may wish to use e.g. `
np.int64` or `np.int32` to specify the precision. If you wish to review your
current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
any behavior and is safe. When replacing `np.int`, you may wish to use e.g. `
np.int64` or `np.int32` to specify the precision. If you wish to review your
current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
```

```
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
any behavior and is safe. When replacing `np.int`, you may wish to use e.g. `
np.int64` or `np.int32` to specify the precision. If you wish to review your
current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
```

```
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
```

```
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
```

```
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
```

```
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
```

```
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
```

```
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
```

```
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
```

```
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
```

```
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
```

```
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
```

```
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
```

```
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
```

```
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
```

```
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
```

```
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
```

```
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
```

```
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
```

```
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
```

```
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
```

```
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
```

```
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
```

```
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
```

```
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
```

```
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
```

```
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
```

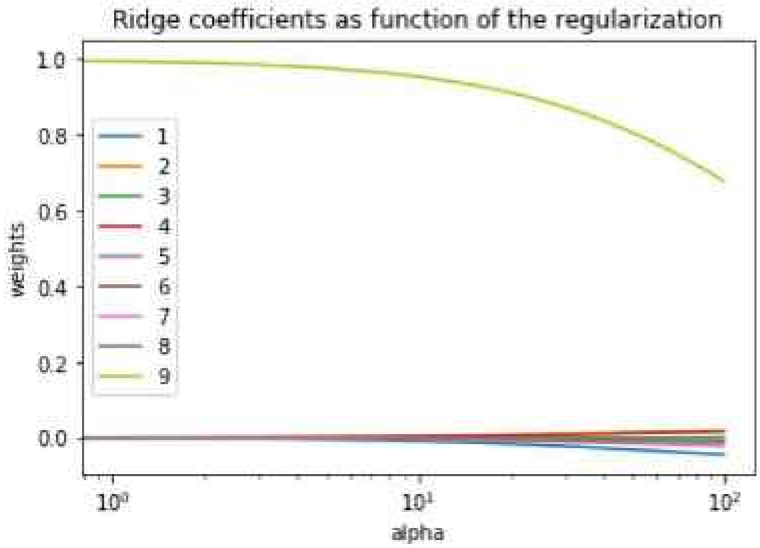
```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
```

```
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:794: DeprecationWarning: `np.int` is a deprecated alias for the builtin `in
t`. To silence this warning, use `int` by itself. Doing this will not modify
 any behavior and is safe. When replacing `np.int`, you may wish to use e.g.
 `np.int64` or `np.int32` to specify the precision. If you wish to review you
r current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_search.p
y:814: DeprecationWarning: The default of the `iid` parameter will change fro
m True to False in version 0.22 and will be removed in 0.24. This will change
numeric results when test-set sizes are unequal.
  DeprecationWarning)
```

```
Out[136]: <matplotlib.legend.Legend at 0x25b13560cc8>
```



Above shows the results of cross validatio of 10 folds. We can observe that the results shows that R-Sqaured is at it's higher value and RMSE is at it's low value. In addition we can see that most of the estimated coefficients are zero only 3 are below. Here we can concluded that these 3 coefficients tells us that there are 6 variables that are of most important.

LASSO REGRESSION

Lasso stands for Least Absolute and Selection Operator which is know as the L-1 norm. Lasso regression is used to reduce the complexity of the model. Lasso regression is similar to Ridge regression except that the penalty term incudes the absolute weights instead of a square weights. In this technique, the L1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero which means there is a complete removal of some of the features for model evaluation when the tuning parameter λ is sufficiently large. Therefore, the lasso method also performs Feature selection and is said to yield sparse models.

In Lasso Regression, we will perfom similar analysis as Ridge Regression. Here our main goal is to see if Lasso Regression out perfoms Ridge Regression.

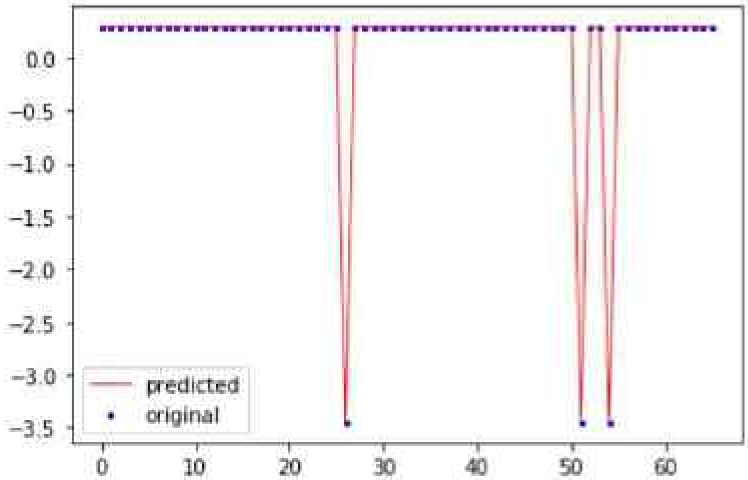
In [139]:
```
# Lasso Regression Model with an alpha value of 0.01
model_lasso1 = Lasso(alpha=0.01)

# fitting the model to the training data
model_lasso1.fit(X_train, y_train)

# predicting
pred_train_lasso1= model_lasso1.predict(X_train)

# prints the evaluation metrics (RMSE and R-squared) on the train set
print('RMSE of train set:           ',np.sqrt(mean_squared_error(y_train,pred_
train_lasso1)))
print('R-Squared of train set:      ',r2_score(y_train, pred_train_lasso1)*100,
'%')

# prints the evaluation metrics (RMSE and R-squared) on the test set
pred_test_lasso1= model_lasso1.predict(X_test)
print('RMSE of test set:            ',np.sqrt(mean_squared_error(y_test,pred_t
est_lasso1)))
print('R-Squared of test set:       ',r2_score(y_test, pred_test_lasso1)*100,
"%")

Lasso_MAPE = MAPE(y_test,pred_test_lasso1)
print("MAPE value:                  ",Lasso_MAPE)
Accuracy = 100 - Lasso_MAPE
print('Accuracy of Lasso Regression: {:0.2f}%.'.format(Accuracy))
```

```
RMSE of train set:             0.00928059975912435
R-Squared of train set:      99.99258170373658 %
RMSE of test set:              0.006866640252862525
R-Squared of test set:       99.99221915582007 %
MAPE value:                  1.0076873210558026
Accuracy of Lasso Regression: 98.99%.
```

In [119]:
```
# retrieving the intercept and the coefficients
print(model_lasso1.intercept_,model_lasso1.coef_)
```

```
-0.0004466974399462559 [-0.          0.          0.          0.          0.
-0.
 -0.         -0.          0.99138705]
```

In [120]:
```
x_ax = range(len(X_test))
plt.scatter(x_ax, y_test, s=5, color="blue", label="original")
plt.plot(x_ax, pred_test_lasso1, lw=0.8, color="red", label="predicted")
plt.legend()
plt.show()
```



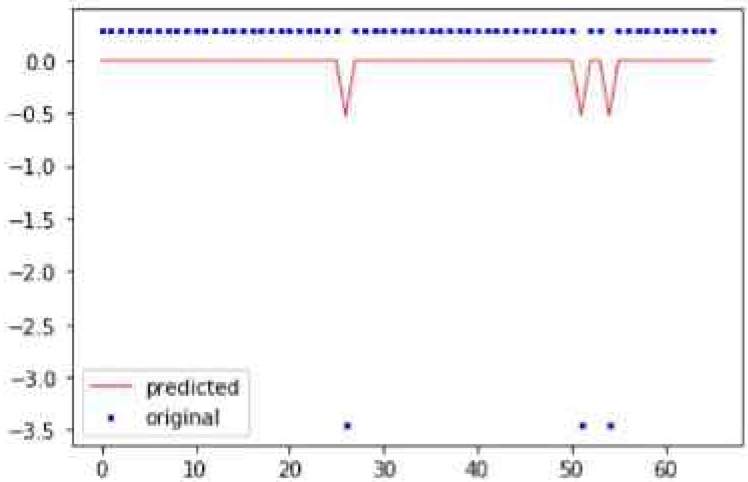
In [121]:
```
# Lasso Regression Model with an alpha value of 1
model_lasso3 = Lasso(alpha=1)

# fitting the model to the training data
model_lasso3.fit(X_train, y_train)

# predicting
pred_train_lasso3= model_lasso3.predict(X_train)

# prints the evaluation metrics (RMSE and R-squared) on the train set
print('RMSE of train set:            ',np.sqrt(mean_squared_error(y_train,pred_
train_lasso3)))
print('R-Squared of train set:      ',r2_score(y_train, pred_train_lasso3)*100
,"%")

# prints the evaluation metrics (RMSE and R-squared) on the test set
pred_test_lasso3= model_lasso3.predict(X_test)
print('RMSE of test set:             ',np.sqrt(mean_squared_error(y_test,pred_t
est_lasso3)))
print('R-Squared of test set:      ',r2_score(y_test, pred_test_lasso3)*100,
"%")

Lasso_MAPE = MAPE(y_test,pred_test_lasso3)
print("MAPE value:                  ",Lasso_MAPE)
Accuracy = 100 - Lasso_MAPE
print('Accuracy of Lasso Regression: {:0.2f}%.'.format(Accuracy))
```

```
RMSE of train set:            0.9280599759124335
R-Squared of train set:       25.817037365716324 %
RMSE of test set:             0.6866640252862516
R-Squared of test set:      22.191558200623817 %
MAPE value:                 100.76873210558048
Accuracy of Lasso Regression: -0.77%.
```

In [122]:
```
# retrieving the intercept and the coefficients
print(model_lasso3.intercept_,model_lasso3.coef_)
```

```
-0.04466974399462546 [-0.          0.          0.          0.          0.
-0.
 -0.         -0.          0.13870468]
```

In [123]:
```
x_ax = range(len(X_test))
plt.scatter(x_ax, y_test, s=5, color="blue", label="original")
plt.plot(x_ax, pred_test_lasso3, lw=0.8, color="red", label="predicted")
plt.legend()
plt.show()
```



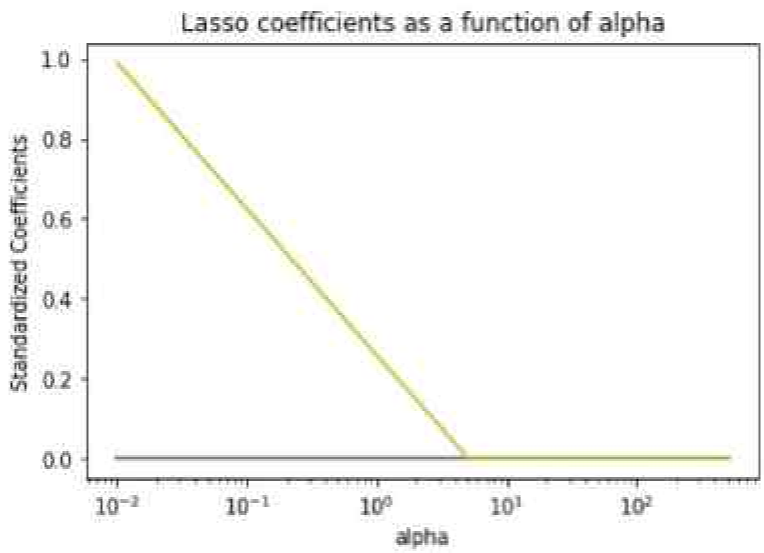
We can see that in Ridge Regression as we increased the value of alpha, coefficients were approaching towards zero, but in the case of Lasso Regression, even at smaller alpha's, our coefficients are reducing to absolute zeroes. Therefore, lasso selects some features while reduces the coefficients of others to zero. This property is known as feature selection and which is absent in Ridge Regression.

On the other hand, we can observe that our R-Squared are higher and RSME are smaller but not as in Ridge Regression. Overall, I observe that Lasso Regression has the best estimates for the coefficients where they are reduced to zero. Since this is what Lasso Regression does; thus, it is the best model.

In [124]:
```python
alphas = range(0,10)
for a in alphas:
 model = Lasso(alpha=a, normalize=True).fit(X,y)
 score = model.score(X, y)
 pred_y = model.predict(X)
 mse = mean_squared_error(y, pred_y)
 print("Alpha:{0:.6f}, R2:{1:.3f}, MSE:{2:.2f}, RMSE:{3:.2f}"
    .format(a, score, mse, np.sqrt(mse)))
```

```
Alpha:0.000000, R2:1.000, MSE:0.00, RMSE:0.00
Alpha:1.000000, R2:0.000, MSE:1.00, RMSE:1.00
Alpha:2.000000, R2:0.000, MSE:1.00, RMSE:1.00
Alpha:3.000000, R2:0.000, MSE:1.00, RMSE:1.00
Alpha:4.000000, R2:0.000, MSE:1.00, RMSE:1.00
Alpha:5.000000, R2:0.000, MSE:1.00, RMSE:1.00
Alpha:6.000000, R2:0.000, MSE:1.00, RMSE:1.00
Alpha:7.000000, R2:0.000, MSE:1.00, RMSE:1.00
Alpha:8.000000, R2:0.000, MSE:1.00, RMSE:1.00
Alpha:9.000000, R2:0.000, MSE:1.00, RMSE:1.00

C:\Users\vivia\Anaconda3\lib\site-packages\ipykernel_launcher.py:3: UserWarni
ng: With alpha=0, this algorithm does not converge well. You are advised to u
se the LinearRegression estimator
  This is separate from the ipykernel package so we can avoid doing imports u
ntil
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_de
scent.py:475: UserWarning: Coordinate descent with no regularization may lead
to unexpected results and is discouraged.
  positive)
```

In [140]:
```
alphas = np.linspace(0.01,500,100)
lasso = Lasso(max_iter=10000)
coefs = []

for a in alphas:
    lasso.set_params(alpha=a)
    lasso.fit(X_train, y_train)
    coefs.append(lasso.coef_)

ax = plt.gca()

ax.plot(alphas, coefs)
ax.set_xscale('log')
plt.axis('tight')
plt.xlabel('alpha')
plt.ylabel('Standardized Coefficients')
plt.title('Lasso coefficients as a function of alpha');
```



CROSS VALIDATION

In [147]:

```
# Lasso with 5 fold cross-validation
model = LassoCV(cv=10, random_state=0, max_iter=10000)

# Fit model
model.fit(X_train, y_train)

LassoCV(cv=10, max_iter=10000, random_state=0)

model.alpha_

# Set best alpha
lasso_best = Lasso(alpha=model.alpha_)
lasso_best.fit(X_train, y_train)

Lasso(alpha=1)

print(list(zip(lasso_best.coef_, X)))

print('R squared training set', round(lasso_best.score(X_train, y_train)*100,
2))
print('R squared test set', round(lasso_best.score(X_test, y_test)*100, 2))

mean_squared_error(y_test, lasso_best.predict(X_test))
```

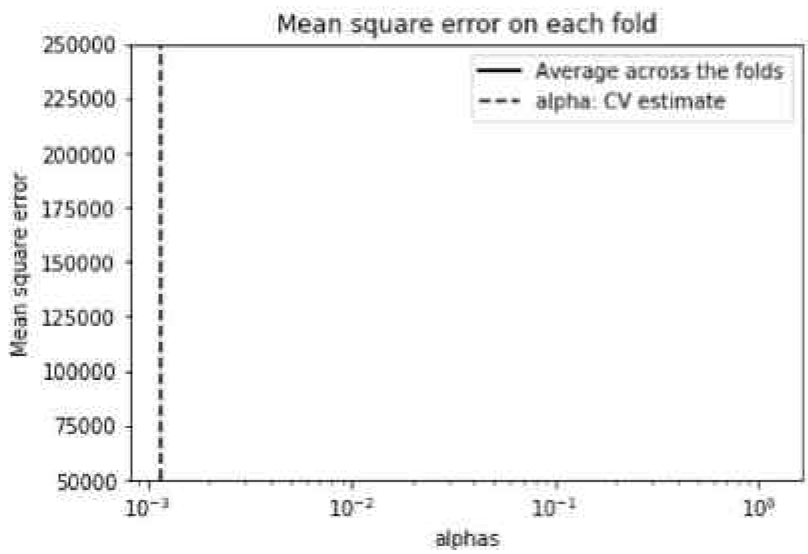
```
[(-0.0, 1), (0.0, 2), (0.0, 3), (0.0, 4), (0.0, 5), (-0.0, 6), (-0.0, 7), (-
0.0, 8), (0.9989999999999999, 9)]
R squared training set 100.0
R squared test set 100.0
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
442: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int
`. To silence this warning, use `int` by itself. Doing this will not modify a
ny behavior and is safe. When replacing `np.int`, you may wish to use e.g. `n
p.int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

```
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\vivia\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
```

Out[147]:
```
6.356007725749039e-07
```

In [148]:
```
plt.semilogx(model.alphas_, model.mse_path_, ":")
plt.plot(
    model.alphas_ ,
    model.mse_path_.mean(axis=-1),
    "k",
    label="Average across the folds",
    linewidth=2,
)
plt.axvline(
    model.alpha_, linestyle="--", color="k", label="alpha: CV estimate"
)

plt.legend()
plt.xlabel("alphas")
plt.ylabel("Mean square error")
plt.title("Mean square error on each fold")
plt.axis("tight")

ymin, ymax = 50000, 250000
plt.ylim(ymin, ymax);
```



To sum up, we can see that our results shows us that Linear regression model performed well. However, if we want to determine variables that are of most important then Cross vlidation and Lasso Regression gives us the best results as they make estimated coefficients to be zeros. We can say that CollegeBound is a variables that is not of the most important.

In [ ]: