


```
import seaborn as sns
import pandas as pd
import numpy as np

df_iris=pd.read_csv('Iris.csv')
```

```
print(df_iris)
```



	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	\
	0	1	5.1	3.5	1.4	0.2
	1	2	4.9	3.0	1.4	0.2
	2	3	4.7	3.2	1.3	0.2
	3	4	4.6	3.1	1.5	0.2
	4	5	5.0	3.6	1.4	0.2
	..	...	...	...	...	...
	145	146	6.7	3.0	5.2	2.3
	146	147	6.3	2.5	5.0	1.9
	147	148	6.5	3.0	5.2	2.0
	148	149	6.2	3.4	5.4	2.3
	149	150	5.9	3.0	5.1	1.8

Species

0	Iris-setosa
1	Iris-setosa
2	Iris-setosa
3	Iris-setosa
4	Iris-setosa
..	...
145	Iris-virginica
146	Iris-virginica
147	Iris-virginica
148	Iris-virginica
149	Iris-virginica


[150 rows x 6 columns]

```
df_iris['Species'].unique()
```



```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```


```
df_iris.isnull().sum()
```



	0
Id	0
SepalLengthCm	0
SepalWidthCm	0
PetalLengthCm	0
PetalWidthCm	0
Species	0

dtype: int64

```
df_iris = df_iris[df_iris['Species']!='Iris-setosa']
df_iris.head()
```



	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
50	51	7.0	3.2	4.7	1.4	Iris-versicolor
51	52	6.4	3.2	4.5	1.5	Iris-versicolor
52	53	6.9	3.1	4.9	1.5	Iris-versicolor
53	54	5.5	2.3	4.0	1.3	Iris-versicolor
54	55	6.5	2.8	4.6	1.5	Iris-versicolor

Next steps:

Generate code with df\_iris

 View recommended plots

New interactive sheet


```
df_iris['Species']=df_iris['Species'].map({'Iris-versicolor':0,'Iris-virginica':1})
```





```
df_iris.head()
```

Next steps: [Generate code with df\\_iris](#) [View recommended plots](#) [New interactive sheet](#)

```
X=df_iris.iloc[:, :-1]
Y=df_iris.iloc[:, -1]
```

X




	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm		
	50	51	7.0	3.2	4.7	1.4	
	51	52	6.4	3.2	4.5	1.5	
	52	53	6.9	3.1	4.9	1.5	
	53	54	5.5	2.3	4.0	1.3	
	54	55	6.5	2.8	4.6	1.5	
	...	...	...	...	...	...	
	145	146	6.7	3.0	5.2	2.3	
	146	147	6.3	2.5	5.0	1.9	
	147	148	6.5	3.0	5.2	2.0	
	148	149	6.2	3.4	5.4	2.3	
	149	150	5.9	3.0	5.1	1.8	

100 rows x 5 columns

Next steps: [Generate code with X](#) [View recommended plots](#) [New interactive sheet](#)

Y



	Species
50	0
51	0
52	0
53	0
54	0
...	...
145	1
146	1
147	1
148	1
149	1

100 rows x 1 columns

dtype: int64

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42)
```

```
from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression()
```

```
from sklearn.model_selection import GridSearchCV
parameter={'penalty':['l1','l2','elasticnet'],'C':[1,2,3,4,5,6,10,20,30,40,50],'max_iter':[100,200,300]}

classifier_regressor=GridSearchCV(classifier,param_grid=parameter,scoring='accuracy',cv=5)

classifier_regressor.fit(X_train,y_train)
```

```

/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed to c
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```

n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed to c
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```

n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed to c
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```

n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed to c
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```

n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed to c
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```

n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed to c
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```

n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed to c
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```

n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed to c
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```

n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed to c
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```

n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed to c
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```

n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed to c
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```

n_iter_i = _check_optimize_result(

```

```
n_iter_i = _check_optimize_result(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:400: ConvergenceWarning: lbfgs failed to converge. Increase the number of iterations (max_iter) or scale the data as shown in:
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_validation.py:425: FitFailedWarning:
330 fits failed out of a total of 495.
```

The score on these train-test partitions for these parameters will be set to nan.

If these failures are not expected, you can try to debug them by setting error\_score='raise'.

Below are more details about the failures:

165 fits failed with the following error:

Traceback (most recent call last):

```
File "/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_validation.py", line 729, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
File "/usr/local/lib/python3.10/dist-packages/sklearn/base.py", line 1152, in wrapper
    return fit_method(estimator, *args, **kwargs)
File "/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py", line 1169, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
File "/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py", line 56, in _check_solver
    raise ValueError(
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.
```

165 fits failed with the following error:

Traceback (most recent call last):

```
File "/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_validation.py", line 729, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
File "/usr/local/lib/python3.10/dist-packages/sklearn/base.py", line 1152, in wrapper
    return fit_method(estimator, *args, **kwargs)
File "/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py", line 1169, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
File "/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py", line 56, in _check_solver
    raise ValueError(
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got elasticnet penalty.
```

```
warnings.warn(some_fits_failed_message, FitFailedWarning)
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.py:979: UserWarning: One or more of the test
nan 1. nan nan 1. nan nan 1. nan nan 1. nan nan 1. nan nan 1. nan
nan 1. nan nan 1. nan nan 1. nan nan 1. nan nan 1. nan nan 1. nan
nan 1. nan nan 1. nan nan 1. nan nan 1. nan nan 1. nan nan 1. nan
nan 1. nan nan 1. nan nan 1. nan nan 1. nan nan 1. nan nan 1. nan
nan 1. nan nan 1. nan nan 1. nan]
warnings.warn(
```

```

> GridSearchCV
  estimator: LogisticRegression
  LogisticRegression()
    LogisticRegression

```

```
print(classifier_regressor.best_params_)
```

```
{'C': 1, 'max_iter': 100, 'penalty': 'l2'}
```

```
print(classifier_regressor.best_score_)
```

```
1.0
```


```
y_pred=classifier_regressor.predict(X_test)
```

```
from sklearn.metrics import accuracy_score,classification_report
```

```
score=accuracy_score(y_pred,y_test)
print(score)
```


```
1.0
```

```
print(classification_report(y_pred,y_test))
```



	precision	recall	f1-score	support
0	1.00	1.00	1.00	14
1	1.00	1.00	1.00	11
accuracy			1.00	25
macro avg	1.00	1.00	1.00	25
weighted avg	1.00	1.00	1.00	25

```
sns.pairplot(df_iris,hue='Species')
```



```
<seaborn.axisgrid.PairGrid at 0x7a2468fc5c30>
```

