Problem 1: Support Vector Machines

Instructions:

1. Please use this q1.ipynb file to complete hw5-q1 about SVMs
2. You may create new cells for discussions or visualizations

In [1]:
```
# Import modules
import numpy as np
import matplotlib.pyplot as plt
import cvxopt
from cvxopt import matrix, solvers
from numpy import linalg
```

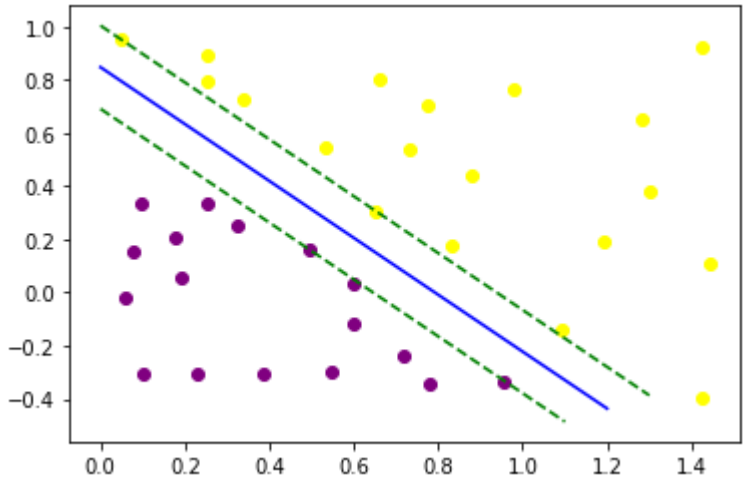
a): Linearly Separable Dataset

In [10]:

```
data = np.loadtxt('clean_lin.txt', delimiter='\t')
X = data[:, 0:2]
y = data[:, 2]

def kernel(x1, x2):
        return np.dot(x1, x2)

class SVM(object):

    def __init__(self, kernel=kernel):
        self.kernel = kernel

    def fit(self, X, y):
        n_samp, n_feat = X.shape

        K = np.zeros((n_samp, n_samp))
        for i in range(n_samp):
            for j in range(n_samp):
                K[i,j] = self.kernel(X[i], X[j])

        P = cvxopt.matrix(np.outer(y,y) * K)
        q = cvxopt.matrix(np.ones(n_samp) * -1)
        A = cvxopt.matrix(y, (1,n_samp))
        b = cvxopt.matrix(0.0)
        G = cvxopt.matrix(np.diag(np.ones(n_samp) * -1))
        h = cvxopt.matrix(np.zeros(n_samp))
        solution = cvxopt.solvers.qp(P, q, G, h, A, b)

        a = np.ravel(solution['x'])

        sv = a > 1e-5
        ind = np.arange(len(a))[sv]
        self.a = a[sv]
        self.sv = X[sv]
        self.sv_y = y[sv]
        print(sv)
        print("This is sv_y")
        print(self.sv_y)

        self.b = 0
        for n in range(len(self.a)):
            self.b += self.sv_y[n]
            self.b -= np.sum(self.a * self.sv_y * K[ind[n],sv])
        self.b /= len(self.a)

        if self.kernel == kernel:
            self.w = np.zeros(n_feat)
            for n in range(len(self.a)):
                self.w += self.a[n] * self.sv_y[n] * self.sv[n]
        else:
            self.w = None

    def Input(self, X):
        if self.w is not None:
            return np.dot(X, self.w) + self.b
        else:
```

```
            y_predict = np.zeros(len(X))
            for i in range(len(X)):
                s = 0
                for a, sv_y, sv in zip(self.a, self.sv_y, self.sv):
                    s += a * sv_y * self.kernel(X[i], sv)
                y_predict[i] = s
            return y_predict + self.b

    def predict(self, X):
        return np.sign(self.Input(X))

if __name__ == "__main__":

    def SVM_Plot(X1, X2, model):
        def f(x, w, b, c=0):
            return (-w[0] * x - b + c) / w[1]
        print("Move graph down for PDF")
        print("Move graph down for PDF")
        print("Move graph down for PDF")
        print("Move graph down for PDF")
        print("Move graph down for PDF")
        print("Move graph down for PDF")
        print("Move graph down for PDF")
        print("Move graph down for PDF")
        print("Move graph down for PDF")
        print("Move graph down for PDF")
        plt.plot(X1[:,0], X1[:,1], "o", color="yellow")
        plt.plot(X2[:,0], X2[:,1], "o", color="purple")

        a0 = 0; a1 = f(a0, model.w, model.b)
        b0 = 1.2; b1 = f(b0, model.w, model.b)
        plt.plot([a0,b0], [a1,b1], "k", c="blue")

        a0 = 0; a1 = f(a0, model.w, model.b, 1)
        b0 = 1.3; b1 = f(b0, model.w, model.b, 1)
        plt.plot([a0,b0], [a1,b1], "k--", c="green")

        a0 = 0; a1 = f(a0, model.w, model.b, -1)
        b0 = 1.1; b1 = f(b0, model.w, model.b, -1)
        plt.plot([a0,b0], [a1,b1], "k--", c="green")

        plt.show()

    def SVM_Model():

        model = SVM()
        model.fit(X, y)

        SVM_Plot(X[y==1], X[y==-1], model)


SVM_Model()
```

```
Move graph down for PDF
Move graph down for PDF
Move graph down for PDF
Move graph down for PDF
```



b) and c) : Linearly Non-separable Dataset

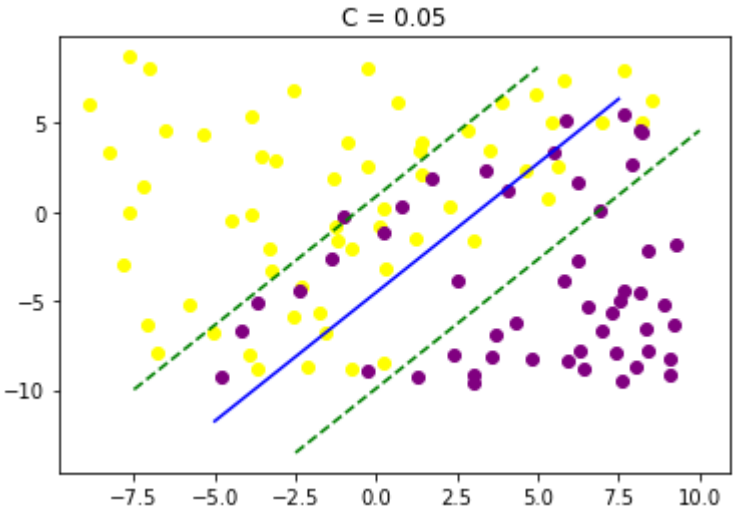
In [5]:

```
# Load the data set that is not linearly separable
data = np.loadtxt('dirty_nonlin.txt', delimiter='\t')
X = data[:, 0:2]
y = data[:, 2]

class SVM(object):

    def __init__(self, kernel=kernel, C=None):
        self.kernel = kernel
        self.C = C
        if self.C is not None: self.C = float(self.C)

    def kernel(x1, x2):
        return np.dot(x1, x2)

    def fit(self, X, y):
        n_samples, n_features = X.shape

        K = np.zeros((n_samples, n_samples))
        for i in range(n_samples):
            for j in range(n_samples):
                K[i,j] = self.kernel(X[i], X[j])

        P = cvxopt.matrix(np.outer(y,y) * K)
        q = cvxopt.matrix(np.ones(n_samples) * -1)
        A = cvxopt.matrix(y, (1,n_samples))
        b = cvxopt.matrix(0.0)

        In = np.diag(np.ones(n_samples) * -1)
        In2 = np.identity(n_samples)
        G = cvxopt.matrix(np.vstack((In, In2)))
        In = np.zeros(n_samples)
        In2 = np.ones(n_samples) * self.C
        h = cvxopt.matrix(np.hstack((In, In2)))

        solution = cvxopt.solvers.qp(P, q, G, h, A, b)

        a = np.ravel(solution['x'])

        sv = a > 1e-5
        ind = np.arange(len(a))[sv]
        self.a = a[sv]
        self.sv = X[sv]
        self.sv_y = y[sv]
        print(sv)
        print("This is sv_y")
        print(self.sv_y)

        self.b = 0
        for n in range(len(self.a)):
            self.b += self.sv_y[n]
            self.b -= np.sum(self.a * self.sv_y * K[ind[n],sv])
        self.b /= len(self.a)

        if self.kernel == kernel:
            self.w = np.zeros(n_features)
```

```
            for n in range(len(self.a)):
                self.w += self.a[n] * self.sv_y[n] * self.sv[n]
        else:
            self.w = None

    def Input(self, X):
        if self.w is not None:
            return np.dot(X, self.w) + self.b
        else:
            y_predict = np.zeros(len(X))
            for i in range(len(X)):
                s = 0
                for a, sv_y, sv in zip(self.a, self.sv_y, self.sv):
                    s += a * sv_y * self.kernel(X[i], sv)
                y_predict[i] = s
            return y_predict + self.b

    def predict(self, X):
        return np.sign(self.Input(X))

if __name__ == "__main__":
    def SVM_Plot(X1, X2, model, CVal):
        def f(x, w, b, c=0):
            return (-w[0] * x - b + c) / w[1]

        plt.plot(X1[:,0], X1[:,1], "o", c="yellow")
        plt.plot(X2[:,0], X2[:,1], "o", c="purple")

        a0 = -5; a1 = f(a0, model.w, model.b)
        b0 = 7.5; b1 = f(b0, model.w, model.b)
        plt.plot([a0,b0], [a1,b1], "k", c="blue")

        a0 = -7.5; a1 = f(a0, model.w, model.b, 1)
        b0 = 5; b1 = f(b0, model.w, model.b, 1)
        plt.plot([a0,b0], [a1,b1], "k--", c="green")

        a0 = -2.5; a1 = f(a0, model.w, model.b, -1)
        b0 = 10; b1 = f(b0, model.w, model.b, -1)
        plt.plot([a0,b0], [a1,b1], "k--", c="green")
        plt.title("C = " + str(CVal))
        plt.show()

    def SVM_Model(CVal=None):
        model = SVM(C=CVal)
        model.fit(X, y)
        SVM_Plot(X[y==1], X[y==-1], model, CVal)

SVM_Model(CVal=0.05)
SVM_Model(CVal=0.1)
SVM_Model(CVal=1)
SVM_Model(CVal=100)
SVM_Model(CVal=1000000)
```

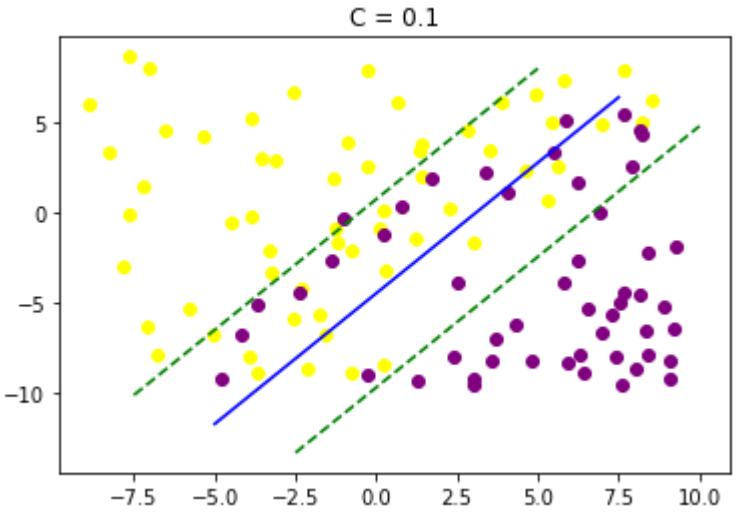
```
     pcost       dcost       gap    pres   dres
 0: -3.0198e+01 -1.1038e+01  6e+02  3e+01  6e-14
 1: -3.0101e+00 -1.0669e+01  3e+01  8e-01  6e-14
 2: -2.1035e+00 -6.2126e+00  6e+00  1e-01  8e-15
```

```
 3: -2.0595e+00 -2.6797e+00  7e-01  1e-02  4e-15
 4: -2.1895e+00 -2.4041e+00  2e-01  3e-03  4e-15
 5: -2.2354e+00 -2.3269e+00  1e-01  1e-03  4e-15
 6: -2.2630e+00 -2.2889e+00  3e-02  2e-04  4e-15
 7: -2.2726e+00 -2.2755e+00  3e-03  2e-16  4e-15
 8: -2.2739e+00 -2.2740e+00  7e-05  9e-17  4e-15
 9: -2.2739e+00 -2.2739e+00  7e-07  2e-16  4e-15
Optimal solution found.
[ True  True  True  True  True  True False False False False False False
 False False False False False False False False  True False False  True
  True  True False False False False False False False False False False
 False False False False False  True  True  True  True  True  True  True
 False  True  True  True  True  True  True  True False False False False
  True  True  True  True  True  True  True  True  True  True  True  True
  True False False False False False False False False False  True False
  True  True  True  True  True  True  True  True False False False False
 False False False False  True  True False False False False False False
 False False False]
This is sv_y
[ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.
 -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.]
```

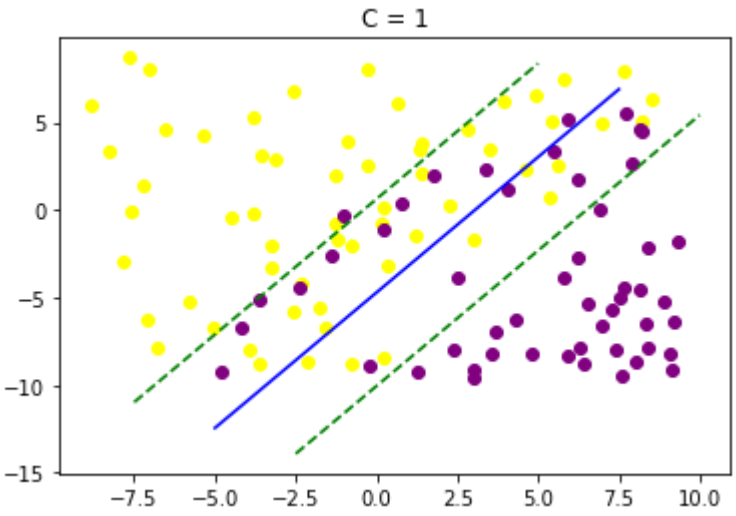



```
    pcost       dcost       gap    pres   dres
 0: -3.1636e+01 -2.2062e+01  7e+02  2e+01  5e-14
 1: -5.6226e+00 -2.0649e+01  5e+01  1e+00  7e-14
 2: -4.1036e+00 -1.1247e+01  1e+01  2e-01  1e-14
 3: -4.1158e+00 -5.1806e+00  1e+00  2e-02  8e-15
 4: -4.3486e+00 -4.7028e+00  4e-01  5e-03  7e-15
 5: -4.4340e+00 -4.5760e+00  2e-01  1e-03  8e-15
 6: -4.4708e+00 -4.5230e+00  5e-02  3e-04  9e-15
 7: -4.4851e+00 -4.5034e+00  2e-02  8e-05  6e-15
 8: -4.4920e+00 -4.4945e+00  3e-03  6e-06  8e-15
 9: -4.4931e+00 -4.4932e+00  3e-05  8e-08  8e-15
10: -4.4931e+00 -4.4931e+00  3e-07  8e-10  8e-15
Optimal solution found.
[ True  True  True  True  True  True False False False False False False
 False False False False False False False False  True False False  True
  True  True False False False False False False False False False False
 False False False False False  True  True  True  True  True  True False
 False  True  True  True  True  True  True  True False False False False
```

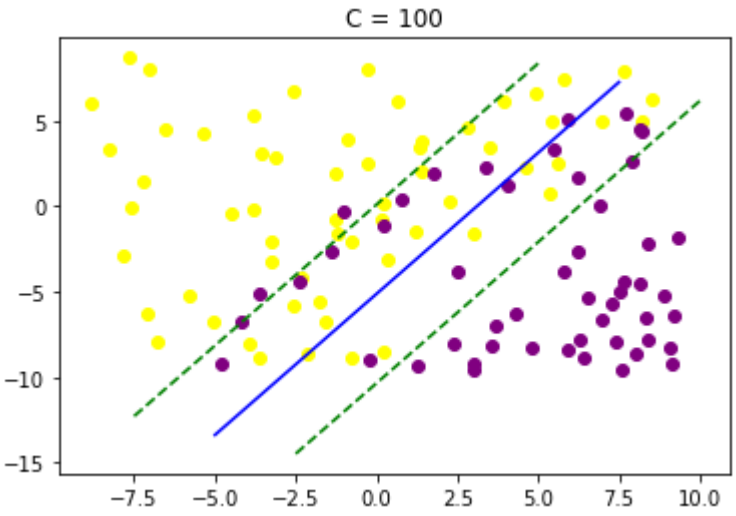
```
  True  True  True  True  True  True  True  True  True  True  True  True
  True False False False False False False False False False  True False
 False  True  True  True  True  True  True  True False False False False
 False False False False  True  True False False False False False False
 False False False]
This is sv_y
[ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.
 -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.]
```



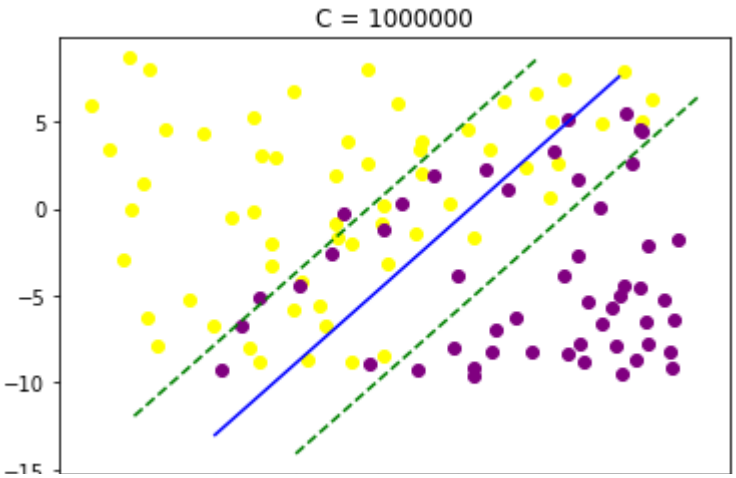

```
     pcost       dcost       gap    pres   dres
 0: -5.7505e+01 -2.5873e+02  1e+03  3e+00  1e-13
 1: -4.0125e+01 -1.6918e+02  2e+02  3e-01  8e-14
 2: -3.7704e+01 -5.8598e+01  2e+01  2e-02  6e-14
 3: -4.1093e+01 -5.0709e+01  1e+01  8e-03  7e-14
 4: -4.2959e+01 -4.6523e+01  4e+00  2e-03  7e-14
 5: -4.3779e+01 -4.5280e+01  2e+00  7e-04  8e-14
 6: -4.4015e+01 -4.4900e+01  9e-01  2e-04  7e-14
 7: -4.4311e+01 -4.4534e+01  2e-01  5e-05  8e-14
 8: -4.4341e+01 -4.4483e+01  1e-01  3e-05  8e-14
 9: -4.4388e+01 -4.4429e+01  4e-02  5e-15  8e-14
10: -4.4407e+01 -4.4408e+01  8e-04  7e-15  9e-14
11: -4.4407e+01 -4.4407e+01  8e-06  1e-14  9e-14
Optimal solution found.
[ True  True  True  True  True  True False False False False False False
 False False False False False False False False  True False False  True
  True  True False False False False False False False False False False
 False False False False  True  True  True  True  True  True  True False
 False  True  True  True  True  True  True  True False False False False
  True  True  True  True  True  True  True  True  True  True  True  True
  True False False False False False False False False False  True False
 False  True  True  True  True  True  True  True False False False False
 False False False False  True  True False False False False False False
 False False False]
This is sv_y
[ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.
 -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.]
```




```
     pcost       dcost       gap    pres   dres
 0: -2.8226e+03 -7.5705e+05  2e+06  6e-01  6e-12
 1: -2.4424e+03 -1.7816e+05  2e+05  1e-02  5e-12
 2: -2.6870e+03 -8.8257e+03  6e+03  3e-04  4e-12
 3: -3.5792e+03 -5.5403e+03  2e+03  8e-05  4e-12
 4: -3.8293e+03 -5.3116e+03  1e+03  5e-05  5e-12
 5: -4.0971e+03 -4.8435e+03  7e+02  2e-05  6e-12
 6: -4.2526e+03 -4.6780e+03  4e+02  1e-05  6e-12
 7: -4.3960e+03 -4.4924e+03  1e+02  4e-07  8e-12
 8: -4.4074e+03 -4.4687e+03  6e+01  1e-07  8e-12
 9: -4.4342e+03 -4.4356e+03  1e+00  2e-09  8e-12
10: -4.4348e+03 -4.4348e+03  6e-02  1e-10  7e-12
11: -4.4348e+03 -4.4348e+03  1e-03  2e-12  9e-12
Optimal solution found.
[ True  True  True  True  True  True False False False False False False
  True  True False False False False False False  True False False  True
  True  True  True False False False False False False False False False
 False False False  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True False False False False
  True  True  True  True  True  True  True  True  True  True  True  True
  True False False False False False False False False False  True  True
  True  True  True  True  True  True  True False False False False
 False False False False  True  True False False False False False False
 False False False]
This is sv_y
[ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1. -1. -1. -1. -1. -1. -1.
 -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.
 -1.]
```




```
     pcost       dcost       gap    pres   dres
 0:  8.1127e+09 -7.3857e+13  2e+14  6e-01  7e-08
 1:  1.8686e+10 -1.6606e+13  2e+13  9e-03  3e-05
 2: -1.4230e+07 -2.8241e+11  3e+11  1e-04  4e-07
 3: -2.6039e+07 -2.9005e+09  3e+09  1e-06  4e-08
 4: -2.6381e+07 -1.0339e+08  8e+07  3e-08  4e-08
 5: -3.4041e+07 -5.7464e+07  2e+07  8e-09  5e-08
 6: -3.7746e+07 -5.3491e+07  2e+07  5e-09  6e-08
 7: -4.0548e+07 -4.9490e+07  9e+06  2e-09  7e-08
 8: -4.2701e+07 -4.6567e+07  4e+06  6e-10  6e-08
 9: -4.3915e+07 -4.5004e+07  1e+06  5e-09  8e-08
10: -4.4016e+07 -4.4780e+07  8e+05  3e-09  7e-08
11: -4.4290e+07 -4.4418e+07  1e+05  3e-09  9e-08
12: -4.4346e+07 -4.4349e+07  3e+03  2e-09  8e-08
13: -4.4347e+07 -4.4347e+07  2e+02  7e-09  8e-08
14: -4.4347e+07 -4.4347e+07  2e+00  4e-09  8e-08
Optimal solution found.
[ True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True]
This is sv_y
[ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.
 -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.
 -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.
 -1. -1. -1.]
```



Explain your observations here:

In [4]:
```
#Higher values of C require more computing power and
#Higher values of C punish points outside the margins
#harsher because of C(Summation)*error
```

In [ ]:
```

```

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import KFold
```

a) load/merge data and visualize logerror

In [2]:
```
# load data into DataFrames
train = pd.read_csv('train.csv')
prop = pd.read_csv('properties.csv')
train_prop = pd.merge(train,prop, on='id')
```

In [3]:
```
train_prop_np = train_prop.to_numpy()
train_prop_np[0][1]
```

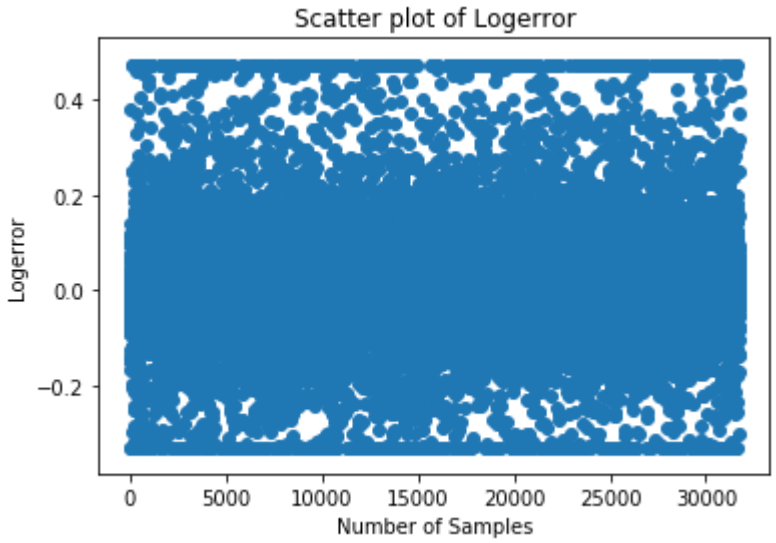
Out[3]:
```
-0.1684
```

In [4]:

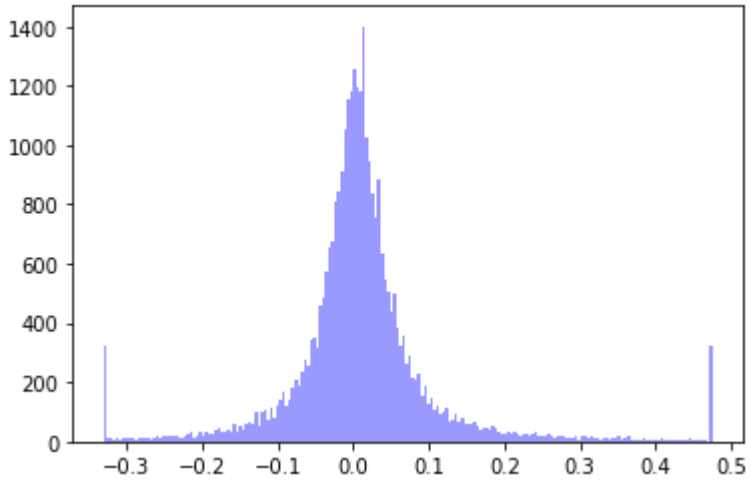
```
# eliminate outliers
train_prop_np = train_prop.to_numpy()
samples,catagories = train_prop_np.shape
listall = []
for logerror in range(samples):
    listall.append(train_prop_np[logerror][1])
print(len(listall))
alllog = np.array(listall)
lowerbound = np.percentile(alllog,q=1)
upperbound = np.percentile(alllog,q=99)

listclean = []
#samples,catagories=trainnp.shape
for logerror in range(samples):
    if train_prop_np[logerror][1] > upperbound:
        print("Outlier too high")
        listclean.append(upperbound)
    elif train_prop_np[logerror][1] < lowerbound:
        print("Outlier too low")
        listclean.append(lowerbound)
    else:
        listclean.append(train_prop_np[logerror][1])
print(len(listclean))
```

```
Outlier too low
Outlier too low
Outlier too low
Outlier too high
Outlier too low
Outlier too low
Outlier too low
Outlier too high
Outlier too low
Outlier too high
Outlier too low
Outlier too high
Outlier too low
Outlier too high
Outlier too low
Outlier too low
Outlier too high
Outlier too high
Outlier too low
31725
```

In [5]:
```
# scatter of logerr
plt.scatter(np.arange(len(listclean)),listclean)
plt.xlabel('Number of Samples')
plt.ylabel('Logerror')
plt.title('Scatter plot of Logerror')
plt.show()
```



In [6]:
```
# histogram of logerr
# histogram of logerr
x = listclean
#What to do with outliers?
num_bins = 100
plt.hist(x,bins="auto",facecolor="blue",alpha=0.4)
plt.show()
```



b) data cleaning

In [7]:
```
# build new data frame
NANData = train_prop.isna().sum()
MissingInfo = {'column_name':train_prop.columns,'missing_count':NANData}
Fixed = pd.DataFrame(data=MissingInfo)
Fixed['missing_ratio'] =  (train_prop.isna().sum())/(len(train_prop))
```

In [8]:
```
# fill missing data
CompleteData = train_prop.fillna(train_prop.mean())
print(CompleteData.isna().sum())
#CompleteData.head()
```

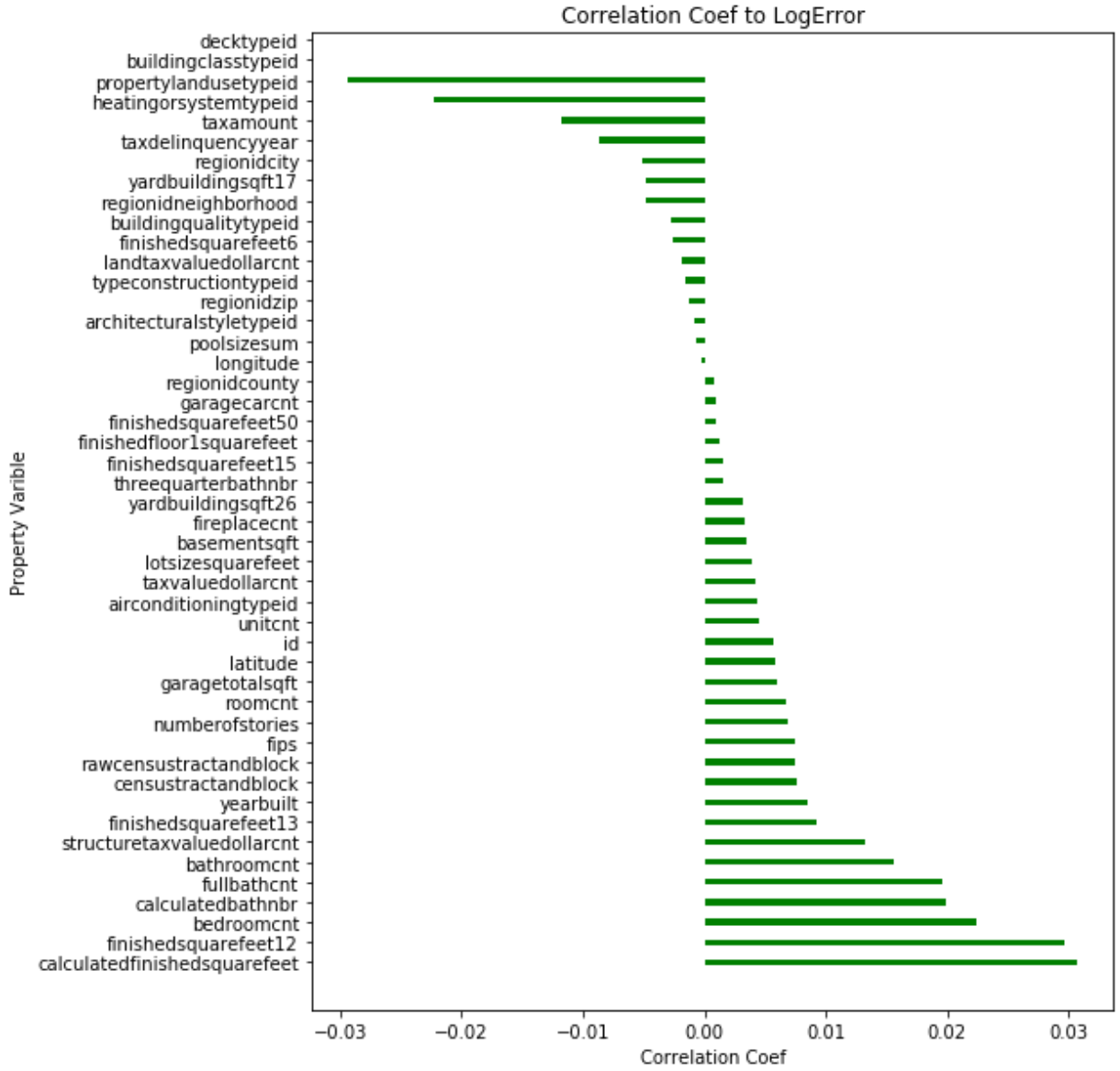
```
propertyzoningdesc              11135
rawcensustractandblock              0
regionidcity                        0
regionidcounty                      0
regionidneighborhood                0
regionidzip                         0
roomcnt                             0
storytypeid                         0
threequarterbathnbr                 0
typeconstructiontypeid              0
unitcnt                             0
yardbuildingsqft17                  0
yardbuildingsqft26                  0
yearbuilt                           0
numberofstories                     0

fireplaceflag                       0
structuretaxvaluedollarcnt          0
taxvaluedollarcnt                   0
assessmentyear                      0
landtaxvaluedollarcnt               0
```

In [9]:
```
print(train_prop["propertyzoningdesc"])
print(train_prop["taxdelinquencyflag"])
#taxdelinquencyflag and propertyzoningdesc are strings and
#do not have averages which is why NaN remain
```

```
0              NaN
1              NaN
2            LARE9
3             LAR1
4            LBR4R
            ...
31720       NOR1YY
31721    LRRPD75002
31722      LRR7000*
31723          LAR1
31724         CARS*
Name: propertyzoningdesc, Length: 31725, dtype: object
0        NaN
1        NaN
2        NaN
3        NaN
4        NaN
        ...
31720    NaN
31721      Y
31722    NaN
31723    NaN
31724    NaN
Name: taxdelinquencyflag, Length: 31725, dtype: object
```

c) univariate analysis

In [10]:
```
# make bar chart
DataCor = CompleteData.corrwith(CompleteData['logerror'])
SortedData = DataCor.sort_values(ascending=False)
SortedData_NoLog = SortedData.drop(['logerror'])
plt.figure(figsize=(8, 10))
plt.title('Correlation Coef to LogError')
plt.barh(SortedData_NoLog.index,SortedData_NoLog.values,0.3, color="green")
plt.xlabel('Correlation Coef')
plt.ylabel('Property Varible')
plt.show()
```



In [11]:
```
print(CompleteData["decktypeid"])
print(CompleteData["buildingclasstypeid"])
```

```
0        66.0
1        66.0
2        66.0
3        66.0
4        66.0
         ...
31720    66.0
31721    66.0
31722    66.0
31723    66.0
31724    66.0
Name: decktypeid, Length: 31725, dtype: float64
0        4.0
1        4.0
2        4.0
3        4.0
4        4.0
        ...
31720    4.0
31721    4.0
31722    4.0
31723    4.0
31724    4.0
Name: buildingclasstypeid, Length: 31725, dtype: float64
```

In [12]:
```
# explain reason
#The varibles at the top have no Correlation Value because all values
#in the catagories decktypeid and buildingclasstypeid are the same
#therefore there is no varience to calculate the Correlation Coef
```

d) non-linear regression model

In [13]:
```
# drop categorical features
# ("hashottuborspa", "propertycountylandusecode", "propertyzoningdesc", "fireplac
# drop "id" and "transactiondate"
RandForestData = CompleteData.drop(["hashottuborspa", "propertycountylandusecode
```

In [14]:
```
# split and train
from sklearn.model_selection import *
from sklearn.ensemble import *
x = RandForestData.drop(["logerror"],axis=1)
y = RandForestData["logerror"]
X_train, X_test, y_train, y_test = train_test_split(x, y, train_size=0.7)
Model = RandomForestRegressor().fit(X_train, y_train)
```

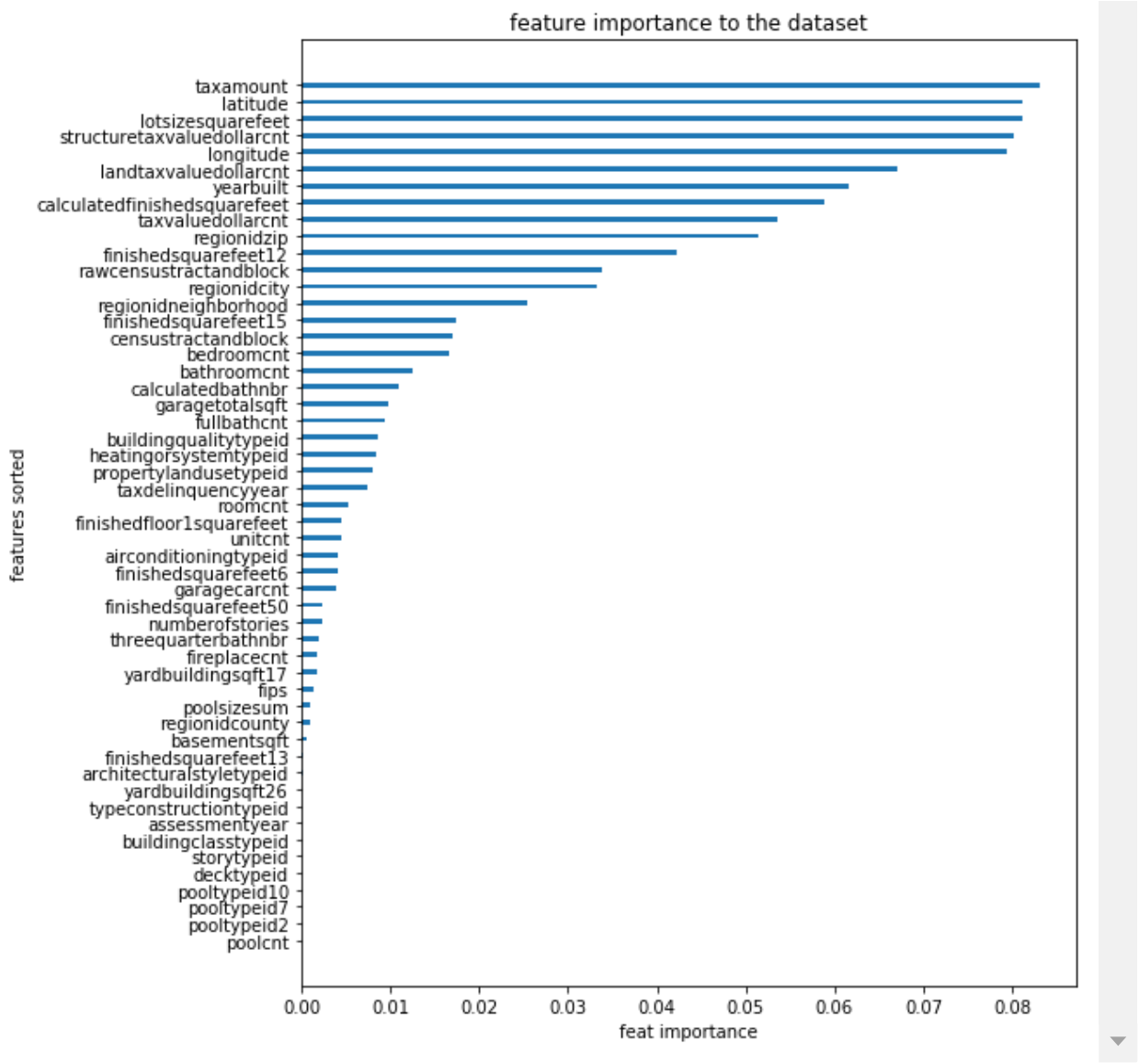
```
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:
30: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float
`. To silence this warning, use `float` by itself. Doing this will not modify a
ny behavior and is safe. If you specifically wanted the numpy scalar type, use
`np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  method='lar', copy_X=True, eps=np.finfo(np.float).eps,
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:
167: DeprecationWarning: `np.float` is a deprecated alias for the builtin `floa
t`. To silence this warning, use `float` by itself. Doing this will not modify
 any behavior and is safe. If you specifically wanted the numpy scalar type, us
e `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  method='lar', copy_X=True, eps=np.finfo(np.float).eps,
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:
284: DeprecationWarning: `np.float` is a deprecated alias for the builtin `floa
t`. To silence this warning, use `float` by itself. Doing this will not modify
 any behavior and is safe. If you specifically wanted the numpy scalar type, us
e `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  eps=np.finfo(np.float).eps, copy_Gram=True, verbose=0,
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:
862: DeprecationWarning: `np.float` is a deprecated alias for the builtin `floa
t`. To silence this warning, use `float` by itself. Doing this will not modify
 any behavior and is safe. If you specifically wanted the numpy scalar type, us
e `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  eps=np.finfo(np.float).eps, copy_X=True, fit_path=True,
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:
1101: DeprecationWarning: `np.float` is a deprecated alias for the builtin `flo
at`. To silence this warning, use `float` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, use
`np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  eps=np.finfo(np.float).eps, copy_X=True, fit_path=True,
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:
1127: DeprecationWarning: `np.float` is a deprecated alias for the builtin `flo
at`. To silence this warning, use `float` by itself. Doing this will not modify
```

```
any behavior and is safe. If you specifically wanted the numpy scalar type, use
`np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  eps=np.finfo(np.float).eps, positive=False):
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:
1362: DeprecationWarning: `np.float` is a deprecated alias for the builtin `flo
at`. To silence this warning, use `float` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, use
`np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  max_n_alphas=1000, n_jobs=None, eps=np.finfo(np.float).eps,
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:
1602: DeprecationWarning: `np.float` is a deprecated alias for the builtin `flo
at`. To silence this warning, use `float` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, use
`np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  max_n_alphas=1000, n_jobs=None, eps=np.finfo(np.float).eps,
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:
1738: DeprecationWarning: `np.float` is a deprecated alias for the builtin `flo
at`. To silence this warning, use `float` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, use
`np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  eps=np.finfo(np.float).eps, copy_X=True, positive=False):
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\decomposition\online_lda.py:
29: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float
`. To silence this warning, use `float` by itself. Doing this will not modify a
ny behavior and is safe. If you specifically wanted the numpy scalar type, use
 `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  EPS = np.finfo(np.float).eps
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\gradient_boosting.p
y:32: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modify an
y behavior and is safe. If you specifically wanted the numpy scalar type, use `
np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  from ._gradient_boosting import predict_stages
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\gradient_boosting.p
y:32: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool
`. To silence this warning, use `bool` by itself. Doing this will not modify an
y behavior and is safe. If you specifically wanted the numpy scalar type, use `
np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
```

```
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  from ._gradient_boosting import predict_stages
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

In [15]:
```
# report importances and mse
ImportData = Model.feature_importances_
y_predict = Model.predict(X_test)
ImportData = pd.DataFrame(data={'feat': x.columns,'import' : ImportData})
ImportDataSorted = ImportData.sort_values(by=['import'])
plt.figure(figsize=(8, 10))
plt.barh(ImportDataSorted['feat'],ImportDataSorted['import'],0.3)
plt.title('feature importance to the dataset')
plt.xlabel('feat importance')
plt.ylabel('features sorted')
plt.show()
print("MSE = " + str(mean_squared_error(y_test, y_predict)))
```

```
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
```




```
MSE = 0.02995938558317681
```

e) KFold

In [16]:
```
# KFold, k = 5
from sklearn.model_selection import KFold

xdata = x[0:500]
ydata = y[0:500]
KFolding = KFold(n_splits=5)

for Train, Test in KFolding.split(xdata):
    X_train, X_test, y_train, y_test = xdata.iloc[Train], xdata.iloc[Test],ydata
    ModelKFold = RandomForestRegressor().fit(X_train, y_train)
    y_predict = ModelKFold.predict(X_test)

print('KFold MSE = ' + str(mean_squared_error(y_test, y_predict)))
```

```
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:44
2: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int`. To
silence this warning, use `int` by itself. Doing this will not modify any behav
ior and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `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/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  fold_sizes = np.full(n_splits, n_samples // n_splits, dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:10
2: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool`.
 To silence this warning, use `bool` by itself. Doing this will not modify any
 behavior and is safe. If you specifically wanted the numpy scalar type, use `n
p.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:10
2: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool`.
 To silence this warning, use `bool` by itself. Doing this will not modify any
 behavior and is safe. If you specifically wanted the numpy scalar type, use `n
p.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
```

```
1.20.0-notes.html#deprecations)
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:10
2: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool`.
 To silence this warning, use `bool` by itself. Doing this will not modify any
 behavior and is safe. If you specifically wanted the numpy scalar type, use `n
p.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:10
2: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool`.
 To silence this warning, use `bool` by itself. Doing this will not modify any
 behavior and is safe. If you specifically wanted the numpy scalar type, use `n
p.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)

KFold MSE = 0.019193319533

C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
```

```
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
102: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `boo
l`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, u
se `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  test_mask = np.zeros(_num_samples(X), dtype=np.bool)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
```

In [17]:
```
# Run d2 for 100 times
xdata = x[0:500]
ydata = y[0:500]
num_times = 100

for RandState in range(num_times):
    X_train, X_test, y_train, y_test = train_test_split(xdata, ydata, train_size
    ModelD2 = RandomForestRegressor(random_state = RandState).fit(X_train, y_tra
    y_predict = ModelD2.predict(X_test)

print("MSE = "+ str(mean_squared_error(y_test, y_predict)))
```

```
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
```

```
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
```

```
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

```
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
```

```
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
```

```
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

```
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)

C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

```
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
```

```
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
```

```
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
```

```
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
```

```
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Deprec
ationWarning: `np.int` is a deprecated alias for the builtin `int`. 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.in
t32` to specify the precision. If you wish to review your current use, check th
e release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Futu
reWarning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

```
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
```

```
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)

C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
```

```
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
```

```
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)


C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
```

```
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
```

```
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
```

```
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)

C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
```

```
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
```

```
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
```

```
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)

C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
```

```
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: Fu
tureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\rdesa\Anaconda3\lib\site-packages\sklearn\ensemble\base.py:158: Depr
ecationWarning: `np.int` is a deprecated alias for the builtin `int`. To sile
nce this warning, use `int` by itself. Doing this will not modify any behavio
r and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` o
r `np.int32` to specify the precision. If you wish to review your current us
e, 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 (https://numpy.org/devdocs/relea
se/1.20.0-notes.html#deprecations)
  dtype=np.int)
```

```
MSE = 0.021743324953037036
```

In [ ]:

Question 3 Flower Classification using CNN

- Please **do not** change the default variable names in this problem, as we will use them in different parts.
- The default variables are initially set to "None".
- You only need to modify code in the "TODO" part. We added some "assertions" to check your code. **Do not** modify them.

In [1]:
```
import numpy as np # linear algebra
import matplotlib.pyplot as plt
import torch
import torch.nn as nn
import torchvision
from torchvision import datasets, transforms, models
from torch.utils.data import *
import random
from tqdm import tqdm
import warnings
```

You can upload your image folder on Google drive and access image folder from it. **Skip it if you run on local machine.** To mount google drive to your current colab page, use the following command

In [2]:
```
#from google.colab import drive
#drive.mount('/content/drive')
```

In [3]:
```
# check pytorch cuda and use cuda if possible
device = torch.cuda.is_available()
print('*' * 50)
if torch.cuda.is_available():
  print('CUDA is found! Tranining on %s.......'%torch.cuda.get_device_name(0))
else:
  warnings.warn('CUDA not found! Training may be slow......')
```

```
**************************************************

C:\Users\rdesa\Anaconda3\lib\site-packages\ipykernel_launcher.py:7: UserWarnin
g: CUDA not found! Training may be slow......
  import sys
```

P1. Data augmentation and plotting

TODO

- Design your image augmentation method for transform_image
- Load train and test data, and split them into train_loader and test_loader
- Visualize your augmented image

In [4]:
```
# TODO: define your image augmentation method
# Make sure to crop the image in (3,224,224) using transforms.RandomResizedCrop(
transform_image = transforms.Compose([transforms.RandomResizedCrop(224), transfo


# TODO: Load data using ImageFolder. Specify your image folder path
path = "C:\\Users\\rdesa\\OneDrive\\Desktop\\Assignment5_Turn_In\\HW5\\q3_data\\
dataset = datasets.ImageFolder(path,transform=transform_image)

n = len(dataset)
n_test = int(0.1 * n)

# Split data into features(pixels) and labels(numbers from 0 to 4)
train_dataset, test_dataset = random_split(dataset, (n-n_test,n_test))
train_loader, test_loader = DataLoader(train_dataset, batch_size=16, shuffle=Tru
```

In [5]:
```
# Sample output
label_map = [['daisy'],['dandelion'],['rose'],['sunflower'],['tulip']]
random_image = random.randint(0,len(train_dataset))
image = train_dataset.__getitem__(random_image)
assert np.array_equal(image[0].detach().numpy().shape, [3,224,224])
plt.imshow(image[0].permute(1,2,0))
plt.title(f"Training example {label_map[image[1]]}")
plt.axis('off')
```

Out[5]:
```
(-0.5, 223.5, 223.5, -0.5)
```

P2. Build you own CNN model

TODO

- Design your own model class in **CNNModel(nn.Module)** and write forward pass in **forward(self, x)**
- Create loss function in **error**, optimizer in **optimizer**
- Define hyparparameters: **learning_rate**, **num_epochs**
- Plot your **loss vs num_epochs** and **accuracy vs num_epochs**

- Plot your first convolution layer kernels using **plot_filters_multi_channel()**

Hints

- Start with low number of epochs for debugging. (eg. num_epochs=1)
- You may want to use small learning rate for training. (eg. 1e-5)
- Be careful with the input dimension of fully connected layer.
- The dimension calculation of the output tensor from the input tensor is \ $D_{out} = \frac{D_{in}-K+2P}{S} + 1$ \ D_{out} : Dimension of output tensor \ D_{in} : Dimension of input tensor \ K : width/height of the kernel \ S : stride \ P : padding

Convolutional and Pooling Layers

A convolutional layer using pyTorch:

```
torch.nn.Conv2d(num_in_channels, num_out_channels, kernel_size, stride=
1, padding=0, dilation=1, groups=1, bias=True, padding_mode='zeros', dev
ice=None, dtype=None)
```

For example:

```
torch.nn.Conv2d(3, 32, 3)
```

It applies a 2D convolution over an input signal composed of several input planes. If we have input size with (N, C_{in}, H, W) and output size with $(N, C_{out}, H_{out}, W_{out})$, the 2D convolution can described as

$$out(N_i, C_{out_j}) = bias(C_{out_j}) + \sum_{k=0}^{C_{in}-1} weight(C_{out_j}, k) \star input(N_i, k)$$

num_in_channels: is the number of channels of the input tensor. If the previous layer is the input layer, num_in_channels is the number of channels of the image (3 channels for RGB images), otherwise num_in_channels is equal to the number of feature maps of the previous layer.

num_out_channels: is the number of filters (feature extractor) that this layer will apply over the image or feature maps generated by the previous layer.

kernel_size: is the size of the convolving kernel So for instance, if we have an RGB image and we are going to apply 32 filters of 3x3:

stide: is the stride of the convolution. Default: 1

padding: is the padding added to all four sides of the input. Default: 0

dilation: is the spacing between kernel elements. Default: 1

group: is the number of blocked connections from input channels to output channels. Default: 1

bias: If True, adds a learnable bias to the output. Default: True

A Simple Convolutional Neural Network

In our convnet we'll use the next structure shown in the comment:

input -> convolution -> pooling -> fully connected -> output \

Convolution #1

16 kernels of 5x5; *Width/Height:* (224 - 5 + 2x0) / 1 + 1 = 220; *Output dimensions:* (16, 220, 220)

Max Pooling #1

filter size = 2, stride = 2; *Width/Height:* (220 - 2) / 2 + 1 = 110; *Output dimensions:* (16, 110, 110)

So at the end of the last convolutional layer we get a tensor of dimension (16, 110, 110). And since now we are going to feed it to fully connected classifier, we need to convert it into a 1-D vector, and for that we use the reshape method:

```
x = x.view(x.size(0), -1)
```

The way of calculating size of the output size from previous convolution layer can be formulized as below:

$$H_{output} = \frac{H_{in} + 2 \times padding - kernel\_Size}{stride} + 1$$

For more details, you can refer to this link: \
https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html (https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html)

In [6]:

```
class CNNModel(nn.Module):
    def __init__(self):
        super(CNNModel, self).__init__()
        self.cnn1 = nn.Conv2d(in_channels=3, out_channels=20, kernel_size=5, str
        self.relu1 = nn.ReLU()
        self.maxpool1 = nn.MaxPool2d(kernel_size=5)
        self.cnn2 = nn.Conv2d(in_channels=20, out_channels=40, kernel_size=5, st
        self.relu2 = nn.ReLU()
        self.maxpool2 = nn.MaxPool2d(kernel_size=5)
        self.cnn3 = nn.Conv2d(in_channels=40, out_channels=80, kernel_size=5, st
        self.relu3 = nn.ReLU()
        self.maxpool3 = nn.MaxPool2d(kernel_size=5)
        self.cnn4 = nn.Conv2d(in_channels=80, out_channels=160, kernel_size=3, s
        self.relu4 = nn.ReLU()
        self.maxpool4 = nn.MaxPool2d(kernel_size=3)
        num_classes = 5
        self.fc1 = nn.Linear(160,1500)
        self.fc2 = nn.Linear(1500,1000)
        self.fc3 = nn.Linear(1000,500)
        self.fc4 = nn.Linear(500,num_classes)

    def forward(self,x):
        x = self.cnn1(x)
        #print(x.shape)
        x = self.relu1(x)
        #print(x.shape)
        x = self.maxpool1(x)
        #print("CNN1")
        #print(x.shape)
        x = self.cnn2(x)
        #print(x.shape)
        x = self.relu2(x)
        #print(x.shape)
        x = self.maxpool2(x)
        #print("CNN2")
        #print(x.shape)
        x = self.cnn3(x)
        #print(x.shape)
        x = self.relu3(x)
        #print(x.shape)
        x = self.maxpool3(x)
        #print(x.shape)
        #print("CNN3")
        x = self.cnn4(x)
        #print(x.shape)
        x = self.relu4(x)
        #print(x.shape)
        x = self.maxpool4(x)
        #print(x.shape)
        #print("CNN4")
        #x = self.cnn5(x)
        #print(x.shape)
        #x = self.relu5(x)
        #print(x.shape)
        #x = self.maxpool5(x)
        #print(x.shape)
```

```
        #print("CNN5")
        x = torch.flatten(x, 1)
        x = self.fc1(x)
        x = self.fc2(x)
        x = self.fc3(x)
        #x = self.fc4(x)
        out = self.fc4(x)

        return out
```

Starting Up Our Model

We'll send the model to our GPU if you have one so we need to create a CUDA device and instantiate our model. Then we will define the loss function and hyperparameters that we need to train the model: \

###TODO

- Define Cross Entropy Loss
- Create Adam Optimizer
- Define hyperparameters

In [7]:
```
# Create CNN
device = "cpu"
model = CNNModel()
model.to(device)

# TODO: define Cross Entropy Loss
error = nn.CrossEntropyLoss()

# TODO: create Adam Optimizer and define your hyperparameters
learning_rate = 50e-5
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)
num_epochs = 20
```

Training the Model

TODO

- Make predictions from your model
- Calculate Cross Entropy Loss from predictions and labels

In [8]:

```
count = 0
loss_list = []
iteration_list = []
accuracy_list = []
for epoch in tqdm(range(num_epochs)):
    model.train()
    print(epoch)
    for i, (images, labels) in enumerate(train_loader):
        images, labels = images.to(device), labels.to(device)

        # Clear gradients
        optimizer.zero_grad()

        # TODO: Forward propagation
        outputs = model(images)

        # TODO: Calculate softmax and ross entropy loss
        loss = error(outputs, labels)

        # Backprop agate your Loss
        loss.backward()

        # Update CNN model
        optimizer.step()

        count += 1

        if count % 50 == 0:
            model.eval()
            # Calculate Accuracy
            correct = 0
            total = 0
            # Iterate through test dataset
            for images, labels in test_loader:
                images, labels = images.to(device), labels.to(device)

                # Forward propagation
                outputs = model(images)

                # Get predictions from the maximum value
                predicted = torch.argmax(outputs,1)

                # Total number of labels
                total += len(labels)

                correct += (predicted == labels).sum()

            accuracy = 100 * correct / float(total)

            # store loss and iteration
            loss_list.append(loss.data.cpu())
            iteration_list.append(count)
            accuracy_list.append(accuracy.cpu())
            print(accuracy.cpu())
        if count % 500 == 0:
            # Print Loss
```

```
            print('Iteration: {}  Loss: {}  Accuracy: {} %'.format(count, loss.da
```

```
  0%|
| 0/20 [00:00<?, ?it/s]

0
tensor(28.0742)
tensor(32.2506)
tensor(30.6264)
tensor(40.1392)

  5%|
| 1/20 [01:24<26:42, 84.32s/it]

1
tensor(38.5151)
tensor(47.5638)
tensor(49.1879)
tensor(47.5638)
tensor(45.7077)

 10%|
| 2/20 [02:54<25:48, 86.05s/it]

2
tensor(50.1160)
Iteration: 500  Loss: 1.3326575756072998  Accuracy: 50.11600875854492 %
tensor(52.9002)
tensor(50.3480)
tensor(48.2599)
tensor(51.5081)

 15%|
| 3/20 [04:25<24:46, 87.47s/it]

3
tensor(50.8121)
tensor(51.9722)
tensor(52.9002)
tensor(55.6845)
tensor(57.5406)

 20%|
| 4/20 [05:56<23:36, 88.51s/it]

4
tensor(47.0998)
Iteration: 1000  Loss: 0.9134666323661804  Accuracy: 47.099769592285156 %
tensor(59.8608)
tensor(58.4687)
tensor(59.3968)
tensor(59.6288)

 25%|
| 5/20 [07:26<22:15, 89.00s/it]

5
tensor(58.2367)
tensor(57.5406)
tensor(58.9327)
```

```
tensor(65.1972)
tensor(62.6450)

 30%|██████████████████████▌
| 6/20 [08:56<20:52, 89.43s/it]

6
tensor(65.6613)
Iteration: 1500  Loss: 0.5248223543167114  Accuracy: 65.6612548828125 %
tensor(67.2854)
tensor(66.5893)
tensor(61.9490)
tensor(63.5731)

 35%|██████████████████████████
| 7/20 [10:26<19:23, 89.49s/it]

7
tensor(61.0209)
tensor(67.2854)
tensor(65.1972)
tensor(67.5174)

 40%|█████████████████████████████▏
| 8/20 [11:51<17:39, 88.28s/it]

8
tensor(61.4849)
tensor(64.7332)
Iteration: 2000  Loss: 0.9467145204544067  Accuracy: 64.7331771850586 %
tensor(67.0534)
tensor(69.1415)
tensor(67.5174)

 45%|█████████████████████████████████▊
| 9/20 [13:21<16:14, 88.61s/it]

9
tensor(67.2854)
tensor(67.7494)
tensor(68.2135)
tensor(71.2297)
tensor(68.4455)

 50%|████████████████████████████████████
| 10/20 [14:50<14:47, 88.77s/it]

10
tensor(66.8213)
tensor(67.7494)
Iteration: 2500  Loss: 0.9420069456100464  Accuracy: 67.74942016601562 %
tensor(68.2135)
tensor(65.4292)
tensor(69.8376)

 55%|████████████████████████████████████████▍
| 11/20 [16:20<13:22, 89.22s/it]

11
tensor(68.9095)
```

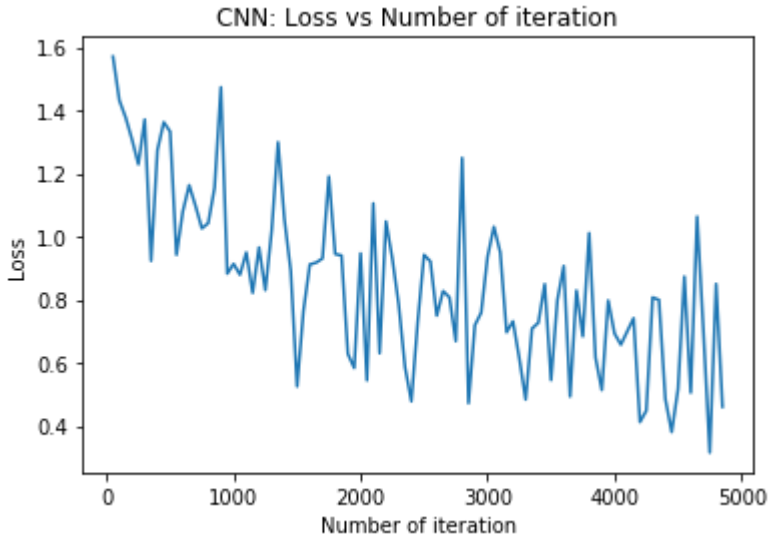
```
tensor(70.9977)
tensor(67.0534)
tensor(72.6218)
tensor(72.1578)

 60%|████████████████████████████████████████▍
| 12/20 [17:50<11:55, 89.46s/it]

12
tensor(67.2854)
tensor(67.7494)
Iteration: 3000  Loss: 0.9361701011657715  Accuracy: 67.74942016601562 %
tensor(70.9977)
tensor(66.8213)
tensor(70.5336)

 65%|████████████████████████████████████████████▏
| 13/20 [19:20<10:26, 89.45s/it]

13
tensor(70.7657)
tensor(70.3016)
tensor(69.8376)
tensor(72.1578)
tensor(64.9652)

 70%|████████████████████████████████████████████████
| 14/20 [20:52<09:01, 90.32s/it]

14
tensor(67.9814)
tensor(71.6937)
Iteration: 3500  Loss: 0.5448697209358215  Accuracy: 71.69373321533203 %
tensor(73.0858)
tensor(71.6937)

 75%|███████████████████████████████████████████████████▊
| 15/20 [22:18<07:24, 88.97s/it]

15
tensor(74.0139)
tensor(65.1972)
tensor(70.7657)
tensor(67.7494)
tensor(68.6775)

 80%|███████████████████████████████████████████████████████▌
| 16/20 [23:48<05:57, 89.43s/it]

16
tensor(75.4060)
tensor(70.3016)
tensor(72.6218)
Iteration: 4000  Loss: 0.6925155520439148  Accuracy: 72.62181091308594 %
tensor(69.1415)
tensor(70.3016)

 85%|███████████████████████████████████████████████████████████▎
| 17/20 [25:18<04:28, 89.62s/it]
```

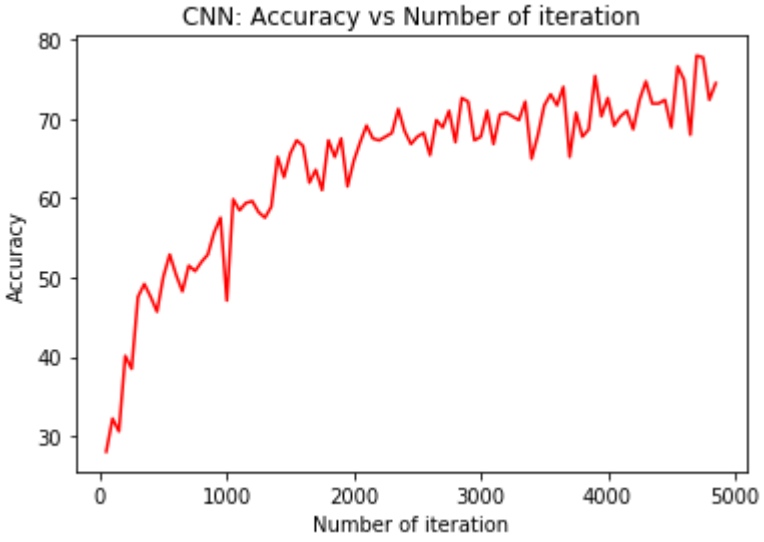
```
17
tensor(70.9977)
tensor(68.6775)
tensor(72.3898)
tensor(74.7100)
tensor(71.9258)

 90%|████████████████████████████████████████████████████████████████
█         | 18/20 [26:49<03:00, 90.09s/it]

18
tensor(71.9258)
tensor(72.3898)
tensor(68.9095)
Iteration: 4500  Loss: 0.5167238712310791  Accuracy: 68.90951538085938 %
tensor(76.5661)
tensor(74.9420)

 95%|████████████████████████████████████████████████████████████████
████     | 19/20 [28:20<01:30, 90.29s/it]

19
tensor(67.9814)
tensor(77.9582)
tensor(77.7262)
tensor(72.3898)
tensor(74.4780)

100%|████████████████████████████████████████████████████████████████
████████| 20/20 [29:51<00:00, 89.56s/it]
```

In [9]:
```
print(accuracy_list)
```

```
[tensor(28.0742), tensor(32.2506), tensor(30.6264), tensor(40.1392), tensor(38.
5151), tensor(47.5638), tensor(49.1879), tensor(47.5638), tensor(45.7077), tens
or(50.1160), tensor(52.9002), tensor(50.3480), tensor(48.2599), tensor(51.508
1), tensor(50.8121), tensor(51.9722), tensor(52.9002), tensor(55.6845), tensor
(57.5406), tensor(47.0998), tensor(59.8608), tensor(58.4687), tensor(59.3968),
tensor(59.6288), tensor(58.2367), tensor(57.5406), tensor(58.9327), tensor(65.1
972), tensor(62.6450), tensor(65.6613), tensor(67.2854), tensor(66.5893), tenso
r(61.9490), tensor(63.5731), tensor(61.0209), tensor(67.2854), tensor(65.1972),
tensor(67.5174), tensor(61.4849), tensor(64.7332), tensor(67.0534), tensor(69.1
415), tensor(67.5174), tensor(67.2854), tensor(67.7494), tensor(68.2135), tenso
r(71.2297), tensor(68.4455), tensor(66.8213), tensor(67.7494), tensor(68.2135),
tensor(65.4292), tensor(69.8376), tensor(68.9095), tensor(70.9977), tensor(67.0
534), tensor(72.6218), tensor(72.1578), tensor(67.2854), tensor(67.7494), tenso
r(70.9977), tensor(66.8213), tensor(70.5336), tensor(70.7657), tensor(70.3016),
tensor(69.8376), tensor(72.1578), tensor(64.9652), tensor(67.9814), tensor(71.6
937), tensor(73.0858), tensor(71.6937), tensor(74.0139), tensor(65.1972), tenso
r(70.7657), tensor(67.7494), tensor(68.6775), tensor(75.4060), tensor(70.3016),
tensor(72.6218), tensor(69.1415), tensor(70.3016), tensor(70.9977), tensor(68.6
775), tensor(72.3898), tensor(74.7100), tensor(71.9258), tensor(71.9258), tenso
r(72.3898), tensor(68.9095), tensor(76.5661), tensor(74.9420), tensor(67.9814),
tensor(77.9582), tensor(77.7262), tensor(72.3898), tensor(74.4780)]
```

In [10]:
```
# visualization loss
plt.plot(iteration_list,loss_list)
plt.xlabel("Number of iteration")
plt.ylabel("Loss")
plt.title("CNN: Loss vs Number of iteration")
plt.show()

# visualization accuracy
plt.plot(iteration_list,accuracy_list,color = "red")
plt.xlabel("Number of iteration")
plt.ylabel("Accuracy")
plt.title("CNN: Accuracy vs Number of iteration")
plt.show()
```





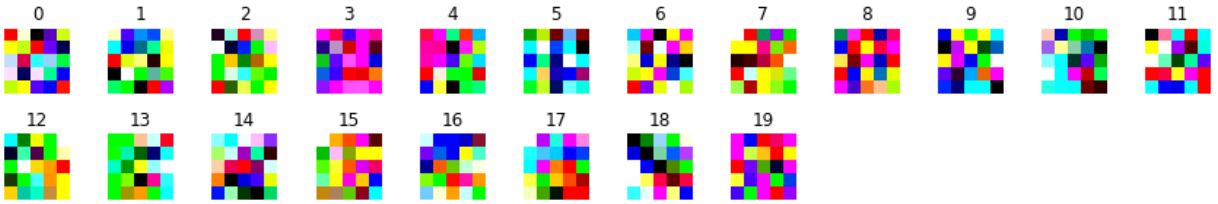
Evaluating the Model

In [11]:
```
# Evaluate your model
random_image = random.randint(0,len(train_dataset))
image = train_dataset.__getitem__(random_image)
model.eval()
images, labels = next(iter(train_loader))
images, labels = images.to(device), labels.to(device)
predictions = torch.argmax(model(images),1)
num_cols=1
num_rows = len(labels)
fig = plt.figure(figsize=(num_cols,num_rows))
for idx in range(num_rows):
  ax1 = fig.add_subplot(num_rows,num_cols,idx+1)
  img = images.cpu().detach()[idx].numpy()
  img = (img - np.mean(img)) / np.std(img)
  img = np.minimum(1, np.maximum(0, (img + 0.5)))
  ax1.imshow(img.transpose((1,2,0)))

  ax1.set_title(f"Label {label_map[labels[idx]]}, Prediction {label_map[predictio
  ax1.axis('off')
plt.savefig('Prediction.png', dpi=100)
plt.show()
```

Visualizing your first layer filter

In [12]:

```
# plot your first layer kernels

def plot_filters_multi_channel(t):
    #make sure the input channel is 3
    assert(t.shape[1]==3)

    #get the number of kernals
    num_kernels = t.shape[0]

    #define number of columns for subplots
    num_cols = 12

    #rows = num of kernels
    num_rows = num_kernels

    #set the figure size
    fig = plt.figure(figsize=(num_cols,num_rows))

    #looping through all the kernels
    for i in range(t.shape[0]):
        ax1 = fig.add_subplot(num_rows,num_cols,i+1)

        #for each kernel, we convert the tensor to numpy
        npimg = np.array(t[i].cpu().detach().numpy(), np.float32)

        #standardize the numpy image
        npimg = (npimg - np.mean(npimg)) / np.std(npimg)
        npimg = np.minimum(1, np.maximum(0, (npimg + 0.5)))
        npimg = npimg.transpose((1, 2, 0))
        ax1.imshow(npimg)
        ax1.axis('off')
        ax1.set_title(str(i))
        ax1.set_xticklabels([])
        ax1.set_yticklabels([])

    plt.savefig('Filter.png', dpi=100)
    plt.tight_layout()
    plt.show()

plot_filters_multi_channel(list(model.parameters())[0])
```



In [ ]: