

2. Following is the code, make sure you update the path to the correct path where you placed the files and update the data frame name correctly:

```

1 # Importing the dataset
2 import pandas as pd
3
4 # Loading the dataset
5 path = "C:/Users/Aditya/Desktop/Iris/Iris.csv"
6 filename = os.path.basename(path)
7 data_iris = pd.read_csv(path, delimiter=',')
8 print(data_iris)
9 print(data_iris.columns)
10 print(data_iris.head())
11 print(data_iris.describe())
12 print(data_iris['Species'].unique())
13
14 # Splitting the predictor and target variables
15 columns = data_iris.columns.values.tolist()
16 predictors = columns[0:4]
17 target = columns[4]
18 print(predictors)
19 print(target)
20
21 # Splitting the dataset into train and test variables
22 from sklearn.model_selection import train_test_split
23 data_iris_train, data_iris_test, y_train, y_test = train_test_split(
24     data_iris[predictors], data_iris[target], test_size=0.3,
25     random_state=0)
26 print("Number of observations in the training data: ", len(y_train))
27 print("Number of observations in the test data: ", len(y_test))

```

3. Separate the predictors from the target then split the dataset using numpy random function. Following is the code, make sure you update the the data frame name correctly

```

1 # Importing the dataset
2 import pandas as pd
3
4 # Loading the dataset
5 path = "C:/Users/Aditya/Desktop/Iris/Iris.csv"
6 filename = os.path.basename(path)
7 data_iris = pd.read_csv(path, delimiter=',')
8 print(data_iris)
9 print(data_iris.columns)
10 print(data_iris.head())
11 print(data_iris.describe())
12 print(data_iris['Species'].unique())
13
14 # Splitting the predictor and target variables
15 columns = data_iris.columns.values.tolist()
16 predictors = columns[0:4]
17 target = columns[4]
18 print(predictors)
19 print(target)
20
21 # Splitting the dataset into train and test variables
22 from sklearn.model_selection import train_test_split
23 data_iris_train, data_iris_test, y_train, y_test = train_test_split(
24     data_iris[predictors], data_iris[target], test_size=0.3,
25     random_state=0)
26 print("Number of observations in the training data: ", len(y_train))
27 print("Number of observations in the test data: ", len(y_test))
28
29 # Splitting the dataset using numpy
30 from sklearn.model_selection import train_test_split
31 data_iris_train, data_iris_test, y_train, y_test = train_test_split(
32     data_iris[predictors], data_iris[target], test_size=0.3,
33     random_state=0)
34 print("Number of observations in the training data: ", len(y_train))
35 print("Number of observations in the test data: ", len(y_test))
36
37 # Building the decision tree
38 from sklearn.tree import DecisionTreeClassifier
39 dt_iris = DecisionTreeClassifier(criterion='entropy', min_samples_split=20,
40     random_state=0)
41 dt_iris.fit(data_iris_train, y_train)
42
43 # Predicting the test set results
44 y_pred = dt_iris.predict(data_iris_test[predictors])
45 pd.crosstab([data_iris_test['Species'], y_pred], columns=['Actual', 'Predicted'])
46
47 # Exporting the model
48 from sklearn.tree import export_graphviz
49 with open("C:/Users/Aditya/Desktop/Iris/iris_model.dot", "w") as outfile:
50     export_graphviz(dt_iris, out_file=outfile, feature_names=predictors)
51     outfile.close()
52
53 # Loading the model
54 from sklearn.tree import load_model
55 model = load_model("C:/Users/Aditya/Desktop/Iris/iris_model.pkl")
56
57 # Predicting the test set results
58 y_pred = model.predict(data_iris_test[predictors])
59 pd.crosstab([data_iris_test['Species'], y_pred], columns=['Actual', 'Predicted'])

```

4. Build the decision tree using the training dataset. Name the model dt\_firstname where firstname is your first name. Use entropy as a method for splitting, and split only when reaching 20 matches.

```

1 # Importing the dataset
2 import pandas as pd
3
4 # Loading the dataset
5 path = "C:/Users/Aditya/Desktop/Iris/Iris.csv"
6 filename = os.path.basename(path)
7 data_iris = pd.read_csv(path, delimiter=',')
8 print(data_iris)
9 print(data_iris.columns)
10 print(data_iris.head())
11 print(data_iris.describe())
12 print(data_iris['Species'].unique())
13
14 # Splitting the predictor and target variables
15 columns = data_iris.columns.values.tolist()
16 predictors = columns[0:4]
17 target = columns[4]
18 print(predictors)
19 print(target)
20
21 # Splitting the dataset into train and test variables
22 from sklearn.model_selection import train_test_split
23 data_iris_train, data_iris_test, y_train, y_test = train_test_split(
24     data_iris[predictors], data_iris[target], test_size=0.3,
25     random_state=0)
26 print("Number of observations in the training data: ", len(y_train))
27 print("Number of observations in the test data: ", len(y_test))
28
29 # Building the decision tree
30 from sklearn.tree import DecisionTreeClassifier
31 dt_firstname = DecisionTreeClassifier(criterion='entropy', min_samples_split=20,
32     random_state=0)
33 dt_firstname.fit(data_iris_train, y_train)
34
35 # Predicting the test set results
36 y_pred = dt_firstname.predict(data_iris_test[predictors])
37 pd.crosstab([data_iris_test['Species'], y_pred], columns=['Actual', 'Predicted'])
38
39 # Exporting the model
40 from sklearn.tree import export_graphviz
41 with open("C:/Users/Aditya/Desktop/Iris/iris_model_firstname.dot", "w") as outfile:
42     export_graphviz(dt_firstname, out_file=outfile, feature_names=predictors)
43     outfile.close()
44
45 # Loading the model
46 from sklearn.tree import load_model
47 model_firstname = load_model("C:/Users/Aditya/Desktop/Iris/iris_model_firstname.pkl")
48
49 # Predicting the test set results
50 y_pred = model_firstname.predict(data_iris_test[predictors])
51 pd.crosstab([data_iris_test['Species'], y_pred], columns=['Actual', 'Predicted'])

```

5. Test the model using the testing dataset and calculate a confusion matrix this time using pandas. Following is the code, make sure you update model name correctly:



