

Sample Code (1/4)

Mounting the drive to make dataset available

```
✓ [1] #import io
      from google.colab import drive
      drive.mount("/content/drive/")
```

Fitting dataset in a dataframe and checking its shape

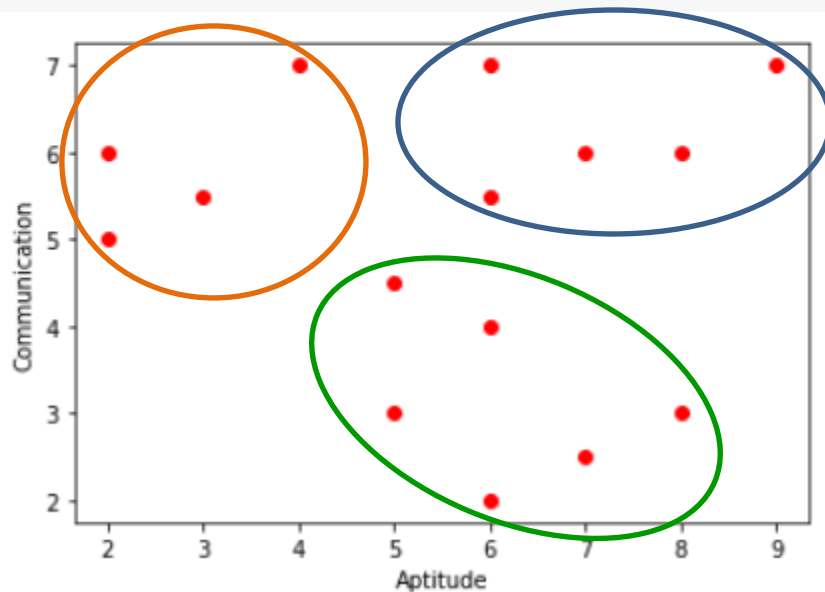
```
[ ] import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt

     df = pd.read_csv('/content/drive/My Drive/Machine Learning/KNN Classification/KNN_Data_Set_v1.0.csv')
     print("Shape of dataset: ", df.shape)
     print("Columns: ", df.columns)
     df.describe()
```

Sample Code (2/4)

Visualize the data

```
▶ plt.xlabel('Aptitude')  
plt.ylabel('Communication')  
plt.scatter(df.Aptitude, df.Communication, color = 'red')  
plt.show()
```



Sample Code (3/4)

Split the dataset into the training set and test set

✓



```
from sklearn.model_selection import train_test_split
predictors = df.iloc[:,1:3]
target = df.iloc[:,3]
#print(predictors)
#print(target)
training_data, testing_data = train_test_split(df, test_size=0.3, random_state=0)
#print(training_data.head)
#print(testing_data.head)
train_X, test_X, train_y, test_y = train_test_split(predictors, target, test_size=0.1, random_state=0)
print(train_X)
print(test_X)
print(train_y)
print(test_y)
```

Create the KNN Classification Model

✓



```
from sklearn.neighbors import KNeighborsClassifier
KNN_Classifier = KNeighborsClassifier(n_neighbors=3)
KNN_Classifier.fit(train_X,train_y)
print("Test set predictions: {}".format(KNN_Classifier.predict(test_X)))
```

Sample Code (4/4)

Evaluate the model:



```
from sklearn.metrics import accuracy_score
accuracy=KNN_Classifier.score(test_X, test_y)
print(accuracy)
from sklearn.metrics import confusion_matrix
predictions = KNN_Classifier.predict(test_X)
cnf_matrix = confusion_matrix(test_y, predictions)
print(cnf_matrix)
```