Khoi Duong

Prof. Yang

CS483

8/18/2022

HW#3

1.

We know that the total accuracy is calculated by the formula:

$$\frac{\text{True Positives + True Negatives}}{\text{True Positives + True Negatives + False Positives + False Negatives}} = \frac{\text{N. of Correct Predictions}}{\text{N. of all Predictions}} = \frac{\text{N. of Correct Predictions}}{\text{Size of Dataset}}$$

Thus, Accuracy $= \frac{20 + 19 + 28}{22 + 22 + 30} = \frac{67}{74} = 0.9054 = 90.54\%$

We know that the precision is calculated by the formula:

$$\frac{\text{True Positives}}{\text{True Positives + False Positives}} = \frac{\text{N. of Correctly Predicted Positive Instances}}{\text{N. of Total Positive Predictions you Made}}$$

We have the precision for “Cat”: 20 / 23 = 0.8696 = 86.96%

We have the precision for “Dog”: 19 / 20 = 0.95 = 95%

We have the precision for “Bird”: 28 / 31 = 0.9032 = 90.32%

We know that the recall is calculated by the formula:

$$\frac{\text{True Positives}}{\text{True Positives + False Negatives}} = \frac{\text{N. of Correctly Predicted Positive Instances}}{\text{N. of Total Positive Instances in the Dataset}}$$

We have the recall for “Cat”: 20 / 22= 0.9091 = 90.91%

We have the recall for “Dog”: 19 / 22= 0.8636 = 86.36%

We have the recall for “Bird”: 28 / 30= 0.9333 = 93.33%

We know that the F1 score is calculated by the formula:

$$2 * \frac{\text{Precision * Recall}}{\text{Precision + Recall}}$$

We have the F1 score for “Cat”: 2 * (0.8696 * 0.9091) / (0.8696 + 0.9091) = 0.8889 = 88.89%

We have the F1 score for “Dog”: 2 * (0.95 * 0.8636) / (0.95 + 0.8636) = 0.9047 = 90.47%

We have the F1 score for “Bird”: 2 * (0.9032 * 0.9333) / (0.9032 + 0.9333) = 0.9180 = 91.80%

2.

Preprocess the dataset by switching the gender into 0 and 1

ID	Name	Age	Gender	Fan
0	Bill	32	0	Rolling Stones
1	Henry	40	0	Neither
2	Mary	16	1	Taylor Swift
3	Tiffany	14	1	Taylor Swift
4	Michael	55	0	Neither
5	Carlos	40	0	Taylor Swift
6	Ashely	20	1	Neither
7	Robert	15	0	Taylor Swift
8	Sally	55	1	Rolling Stones
9	John	15	0	Rolling Stones
10	Michelle	10	1	?

Source code:

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

from google.colab import drive
drive.mount('/content/drive')
data_path = "/content/drive/My Drive/Colab Notebooks/hw3_ex2.csv"
dataset = pd.read_csv(data_path)

X = dataset.iloc[:, 2:4].values           # Assign 1st/2nd/3rd/4th colums
values to X
y = dataset.iloc[:, 4].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)

X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

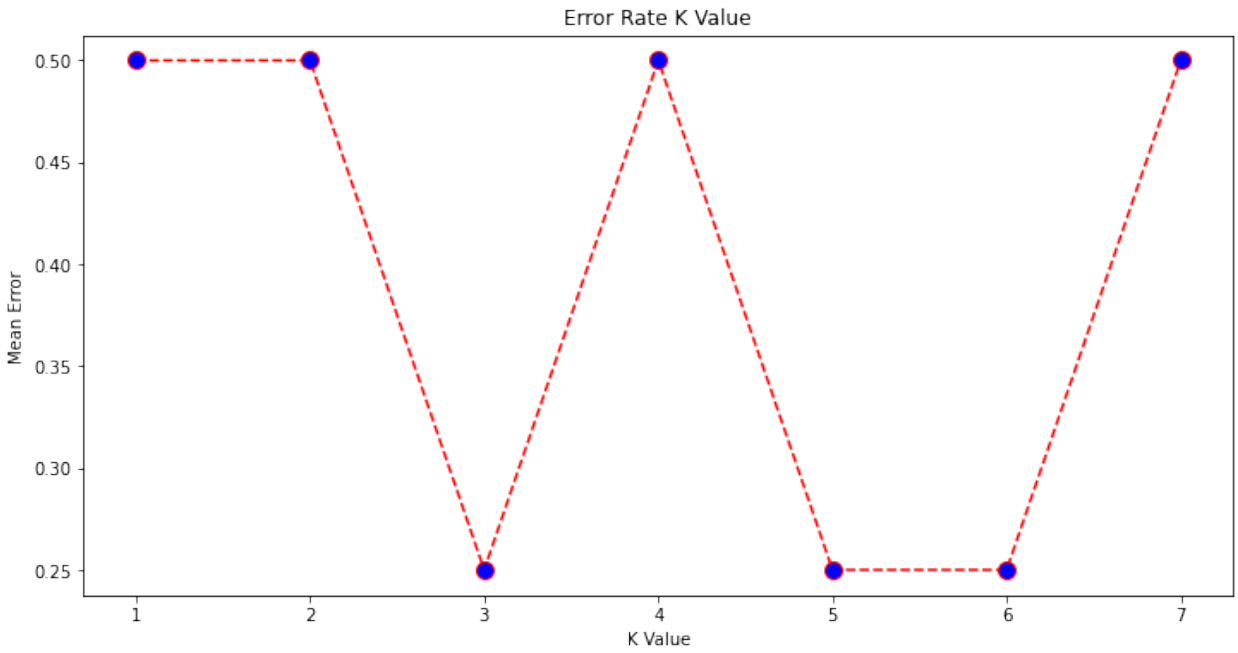
```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5)
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))

error = []

# Calculating error for K values between 1 and 7
for i in range(1, 8):
    knn = KNeighborsClassifier(n_neighbors=i)    # K = 1 to 7
    knn.fit(X_train, y_train)
    pred_i = knn.predict(X_test)
    error.append(np.mean(pred_i != y_test))
print("Error rate of validation set for K = 1 to 7", error)
plt.figure(figsize=(12, 6))
plt.plot(range(1, 8), error, color='red', linestyle='dashed', marker='o',
         markerfacecolor='blue', markersize=10)
plt.title('Error Rate K Value')
plt.xlabel('K Value')
plt.ylabel('Mean Error')
```

Run program & result:

```
[[1 0 0]
 [0 0 1]
 [0 0 2]]
                precision    recall  f1-score   support

       Neither       1.00      1.00      1.00         1
Rolling Stones       0.00      0.00      0.00         1
  Taylor Swift       0.67      1.00      0.80         2
```

```
      accuracy                           0.75         4
     macro avg       0.56      0.67      0.60         4
  weighted avg       0.58      0.75      0.65         4
```

```
Error rate of validation set for K = 1 to 7 [0.5, 0.5, 0.25, 0.5, 0.25,
0.25, 0.5]
Text(0, 0.5, 'Mean Error')
```



Based on the rule of thumb of K selection: K = sqrt(10) = 3.16 = 3

We choose K = 3

And after prediction, we have

```
10    Michelle    10    1     Taylor Swift
```

Thus, the new data in the red color belongs to class “Taylor Swift”

3.

We have N = 8, thus the best K-value will be: $K = \sqrt{8} = 2.828 \approx 3$

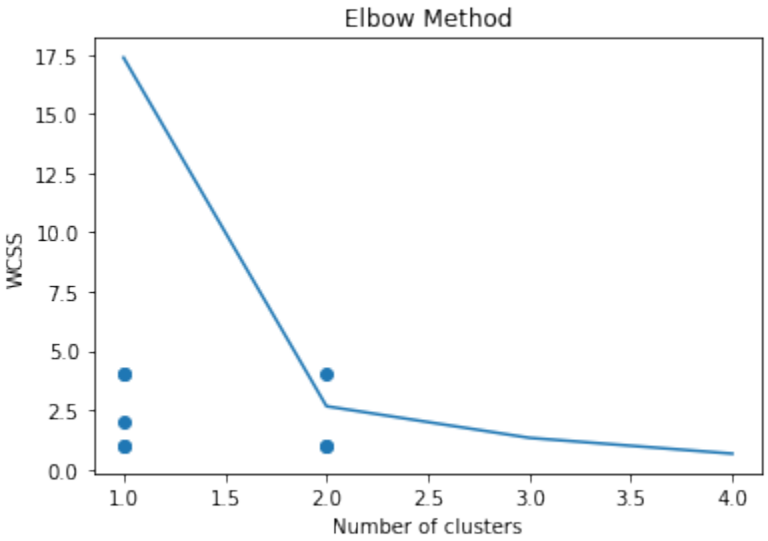
We have the source code below:

```
import numpy as np
from matplotlib import pyplot as plt
from sklearn.cluster import KMeans

X = np.array([[1,4],
              [1,2],
              [1,4],
              [2,1],
              [1,1],
              [2,4],
              [1,1],
              [2,1]])
y = np.array([1,2,2,2,1,2,2,1])

plt.scatter(X[:,0], X[:,1])
print(y.shape, X.shape)


wcss = []
for i in range(1, 5):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300,
n_init=10, random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 5), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
print(wcss)


kmeans = KMeans(n_clusters=3, init='k-means++', max_iter=300, n_init=10,
random_state=0)
pred_y = kmeans.fit_predict(X)
plt.scatter(X[:,0], X[:,1])
```

```
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
s=300, c='red')
plt.show()
```

Run program & result:

```
(8,) (8, 2)
```




```
[17.375, 2.6666666666666665, 1.3333333333333335, 0.6666666666666666]
```