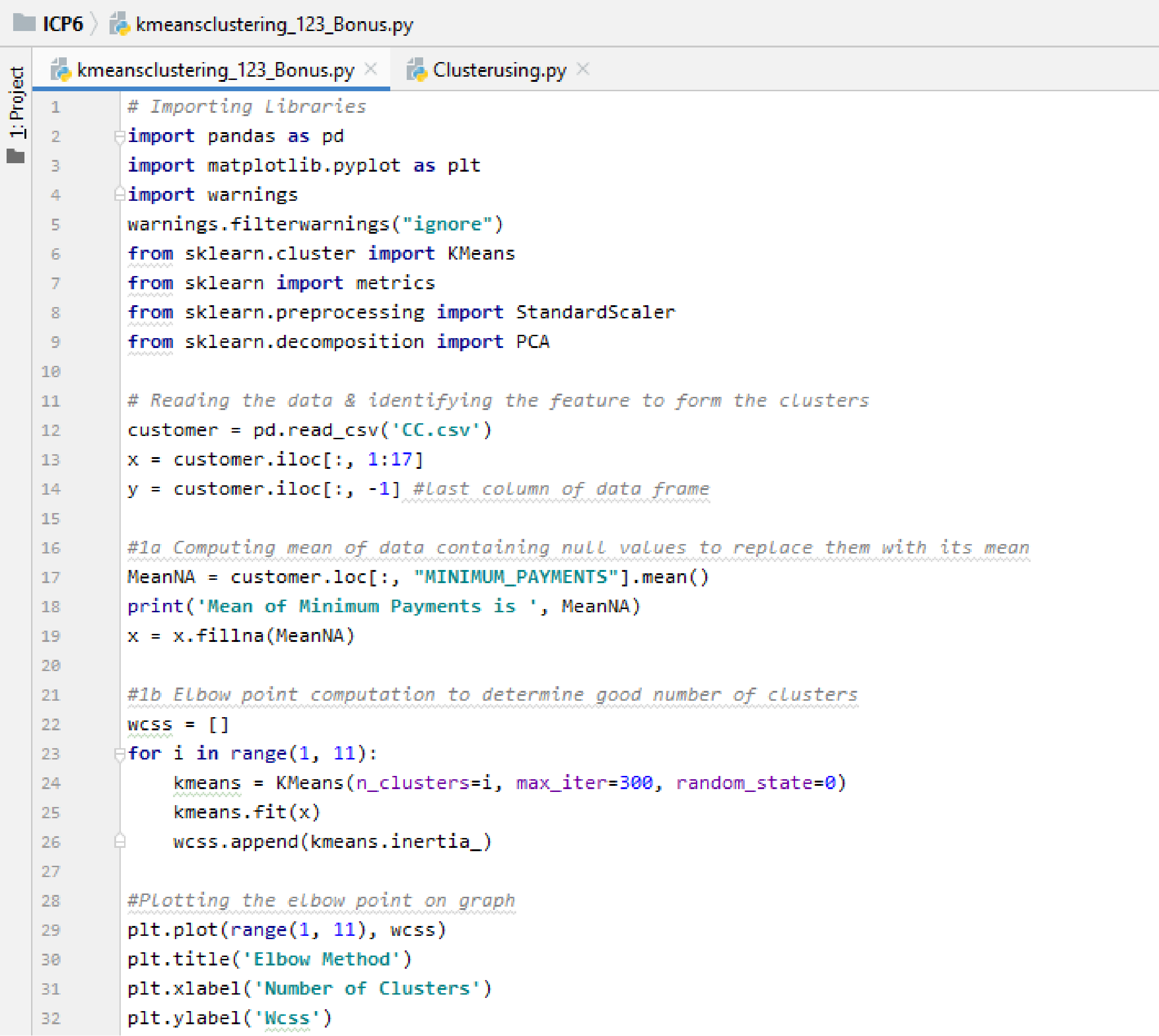
CS5590/490 - Python and Deep Learning Programming
In Class Programming Report - 6
Class ID 24 - Anurag Thantharate
Date Submitted: 03/01/2020

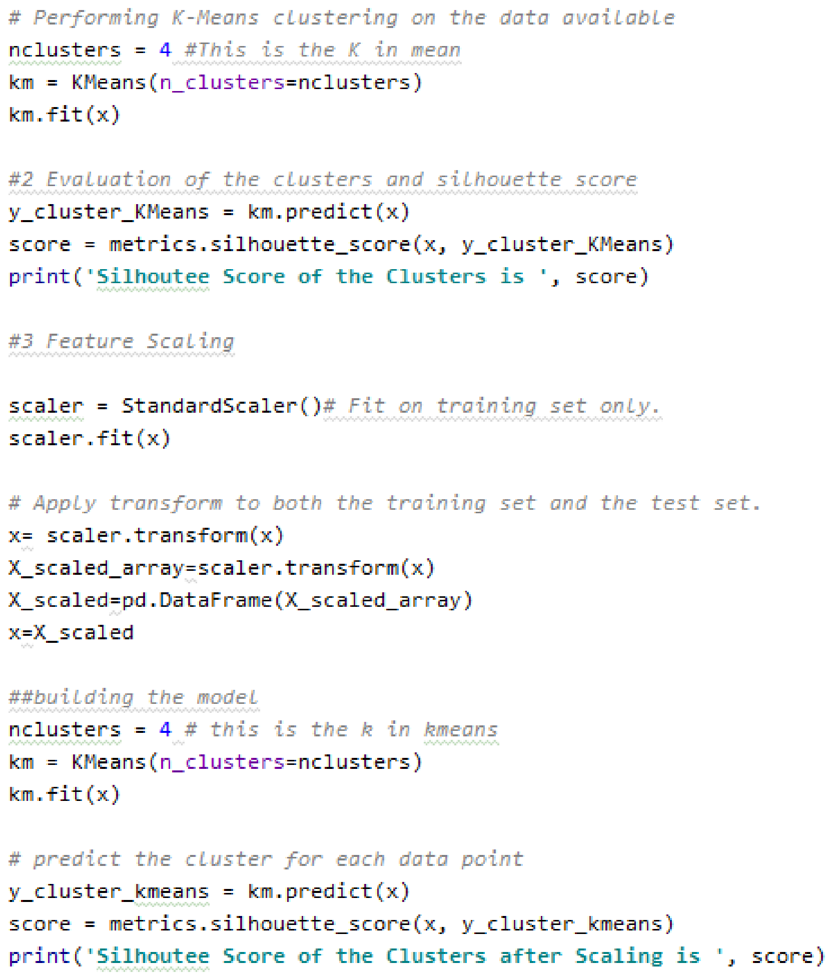
Video Link: https://www.loom.com/share/618271486570498289ab449e8f89dcba

---

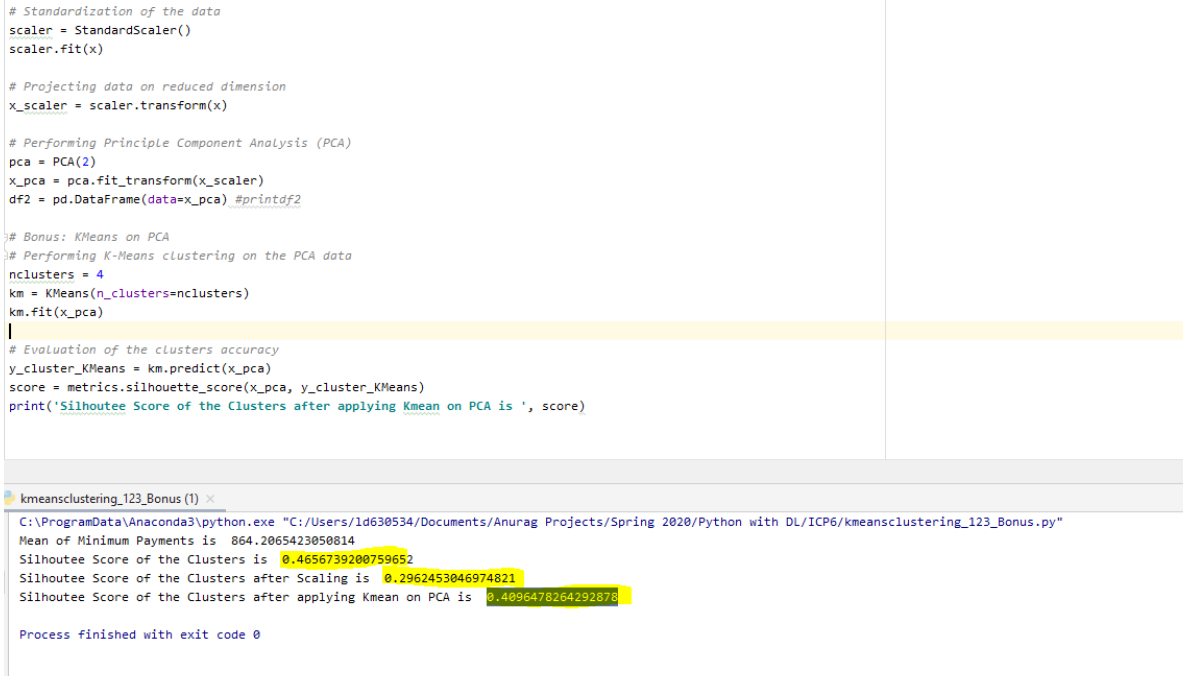
1. Apply K means clustering in this data set provided below: https://umkc.box.com/s/s15r7m0gnxu7b1s2kaobvc5w7da2nc1c
   a. Remove any null values by the mean.
   b. Use the elbow method to find a good number of clusters with the KMeans algorithm
2. Calculate the silhouette score for the above clustering
3. Try feature scaling to see if it will improve the Silhouette score
4. Apply PCA on the same dataset.
5. *** Bonuspoints Apply kmeans algorithm on the PCA result and report your observation if the score improved or not?


ICP6 〉 kmeansclustering_123_Bonus.py
kmeansclustering_123_Bonus.py ×
Clusterusing.py ×
1: Project

```python
# Importing Libraries
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
from sklearn.cluster import KMeans
from sklearn import metrics
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

# Reading the data & identifying the feature to form the clusters
customer = pd.read_csv('CC.csv')
x = customer.iloc[:, 1:17]
y = customer.iloc[:, -1] #last column of data frame

#1a Computing mean of data containing null values to replace them with its mean
MeanNA = customer.loc[:, "MINIMUM_PAYMENTS"].mean()
print('Mean of Minimum Payments is ', MeanNA)
x = x.fillna(MeanNA)

#1b Elbow point computation to determine good number of clusters
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, max_iter=300, random_state=0)
    kmeans.fit(x)
    wcss.append(kmeans.inertia_)

#Plotting the elbow point on graph
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('Wcss')
```



```
# Performing K-Means clustering on the data available
nclusters = 4 #This is the K in mean
km = KMeans(n_clusters=nclusters)
km.fit(x)

#2 Evaluation of the clusters and silhouette score
y_cluster_KMeans = km.predict(x)
score = metrics.silhouette_score(x, y_cluster_KMeans)
print('Silhoutee Score of the Clusters is ', score)

#3 Feature Scaling

scaler = StandardScaler()# Fit on training set only.
scaler.fit(x)

# Apply transform to both the training set and the test set.
x= scaler.transform(x)
X_scaled_array=scaler.transform(x)
X_scaled=pd.DataFrame(X_scaled_array)
x=X_scaled

##building the model
nclusters = 4 # this is the k in kmeans
km = KMeans(n_clusters=nclusters)
km.fit(x)

# predict the cluster for each data point
y_cluster_kmeans = km.predict(x)
score = metrics.silhouette_score(x, y_cluster_kmeans)
print('Silhoutee Score of the Clusters after Scaling is ', score)
```



```
# Standardization of the data
scaler = StandardScaler()
scaler.fit(x)

# Projecting data on reduced dimension
x_scaler = scaler.transform(x)

# Performing Principle Component Analysis (PCA)
pca = PCA(2)
x_pca = pca.fit_transform(x_scaler)
df2 = pd.DataFrame(data=x_pca) #printdf2

# Bonus: KMeans on PCA
# Performing K-Means clustering on the PCA data
nclusters = 4
km = KMeans(n_clusters=nclusters)
km.fit(x_pca)

# Evaluation of the clusters accuracy
y_cluster_KMeans = km.predict(x_pca)
score = metrics.silhouette_score(x_pca, y_cluster_KMeans)
print('Silhoutee Score of the Clusters after applying Kmean on PCA is ', score)
```

```
C:\ProgramData\Anaconda3\python.exe "C:/Users/ld630534/Documents/Anurag Projects/Spring 2020/Python with DL/ICP6/kmeansclustering_123_Bonus.py"
Mean of Minimum Payments is  864.2065423050814
Silhoutee Score of the Clusters is  0.4656739200759652
Silhoutee Score of the Clusters after Scaling is  0.2962453046974821
Silhoutee Score of the Clusters after applying Kmean on PCA is  0.4096478264292878

Process finished with exit code 0
```