Data-Mining Assignment

Dataset Info :

Dataset:

Malls or shopping complexes are often indulged in the race to increase their customers and hence making huge profits. To achieve this task machine learning is being applied by many stores already.

It is amazing to realize the fact that machine learning can aid in such ambitions. The shopping complexes make use of their customers' data and develop ML models to target the right ones. This not only increases sales but also makes the complexes efficient.

Aim : Customer Segmentation:

Customer segmentation is the practice of dividing a customer base into groups of individuals that are similar in specific ways. It allows for the effective allocation of marketing resources and the maximization of cross and up-selling opportunities.

Interpretations made:

- **Demographic information**, such as gender, age, familial and marital status, income, education, and occupation.
- **Geographical information**, which differs depending on the scope of the company. For localized businesses, this info might pertain to specific towns or counties. For larger companies, it might mean a customer's city, state, or even country of residence.
- **Psychographics**, such as social class, lifestyle, and personality traits.
- **Behavioral data**, such as spending and consumption habits, product/service usage, and desired benefits.

Using the above information following things can be done.

- Determine appropriate product pricing.
- Develop customized marketing campaigns.
- Design an optimal distribution strategy.
- Choose specific product features for deployment.
- Prioritize new product development efforts.

B. Description:

Source : Link to source
The data contains ~ 200 records with 5 features- Customer Id, Gender, Age, Annual Income (k$) and Spending Score (1–100). Here Spending Score refers to Score assigned by the mall based on customer behavior and spending nature. The features in the dataset are explained below:

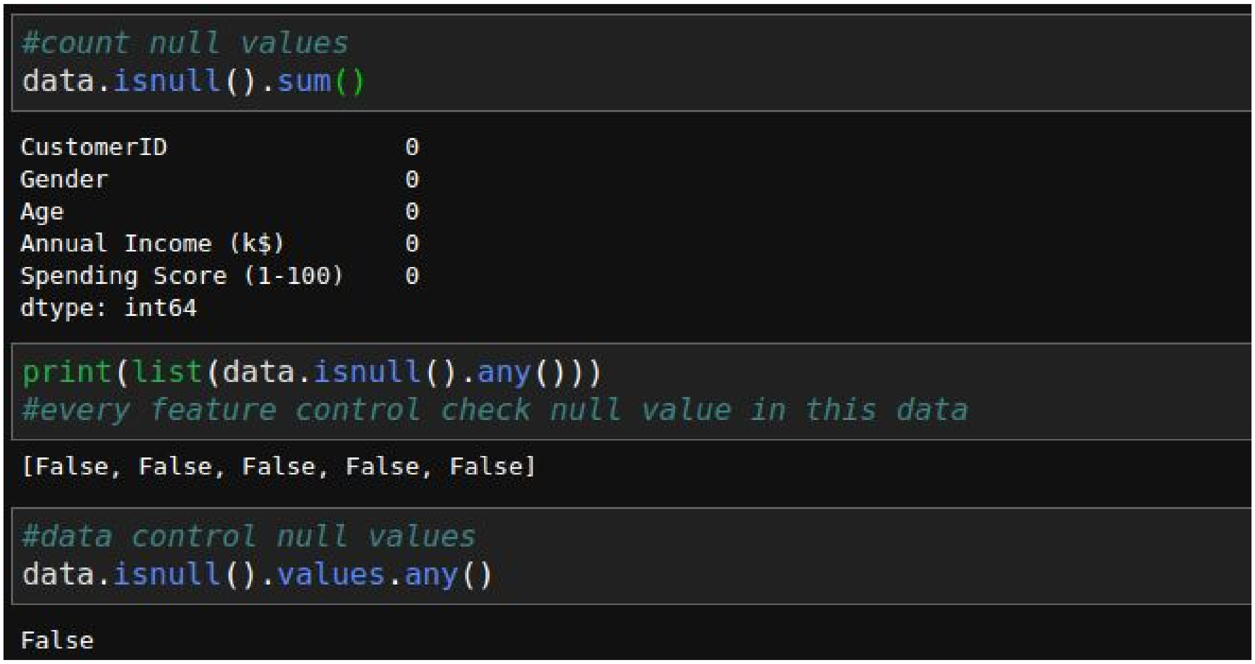
- Customer ID - It is the unique ID assigned to the customer
- Gender - Gender of the customer
- Age - Age of the customer(in years)
- Annual Income(k$) - Annual income of the customer in k$
- Spending Score - Score assigned to the customer by the mall/shopping complex based on the customer spending nature and behaviour

C. Packages

- Python
- Pandas
- Scikit-learn
- Numpy
- Matplotlib

D.

1. Data Preprocessing:



```
#count null values
data.isnull().sum()
```

```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

```
print(list(data.isnull().any()))
#every feature control check null value in this data
```

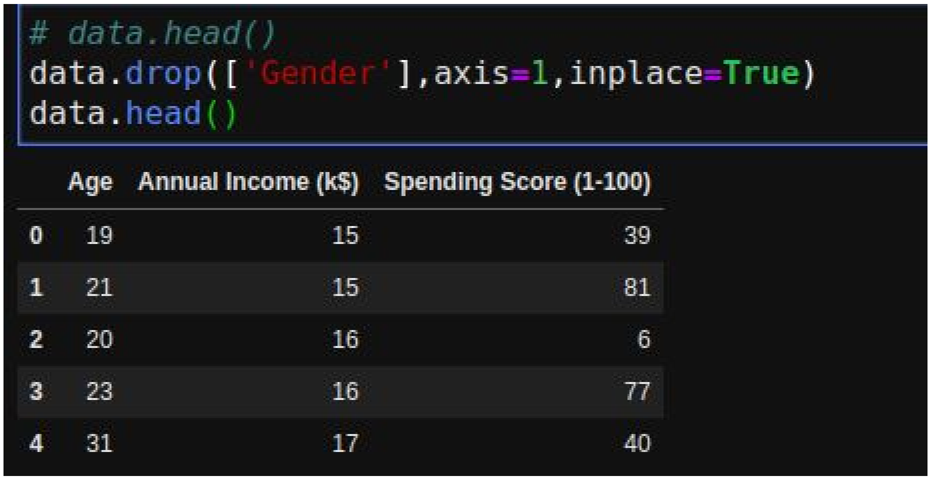
```
[False, False, False, False, False]
```

```
#data control null values
data.isnull().values.any()
```

```
False
```

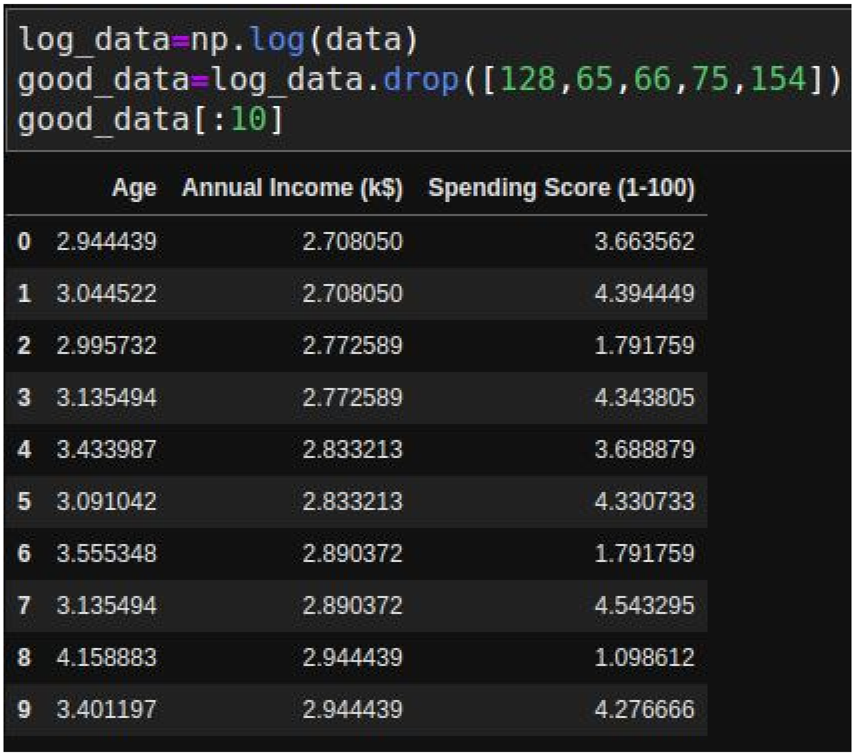
We have zero null values in any column.

To implement dimensionality reduction I have scaled the data. And I have also removed gender as to scale the data I have to perform logarithmic scaling. Since gender is categorical data I have removed it. Also removed CustomerId as it is not useful.



```
# data.head()
data.drop(['Gender'],axis=1,inplace=True)
data.head()
```

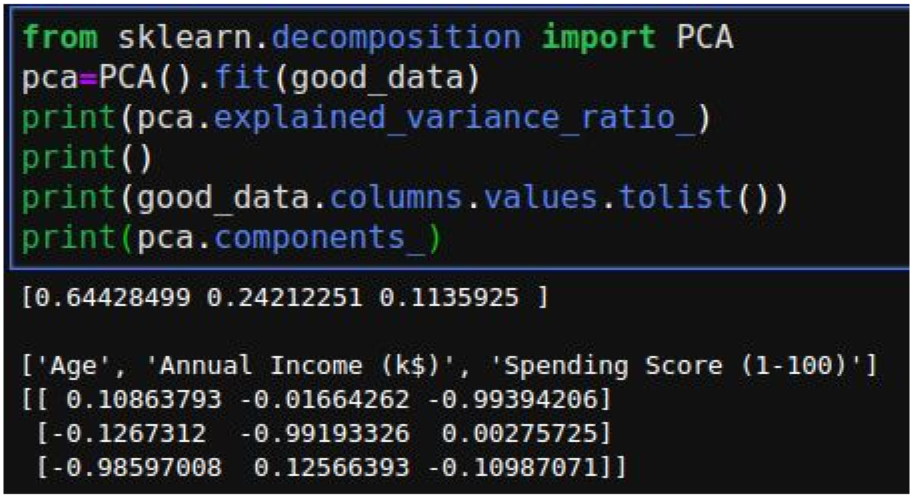
	Age	Annual Income (k$)	Spending Score (1-100)
0	19	15	39
1	21	15	81
2	20	16	6
3	23	16	77
4	31	17	40



```
log_data=np.log(data)
good_data=log_data.drop([128,65,66,75,154])
good_data[:10]
```

	Age	Annual Income (k$)	Spending Score (1-100)
0	2.944439	2.708050	3.663562
1	3.044522	2.708050	4.394449
2	2.995732	2.772589	1.791759
3	3.135494	2.772589	4.343805
4	3.433987	2.833213	3.688879
5	3.091042	2.833213	4.330733
6	3.555348	2.890372	1.791759
7	3.135494	2.890372	4.543295
8	4.158883	2.944439	1.098612
9	3.401197	2.944439	4.276666

Perform PCA:

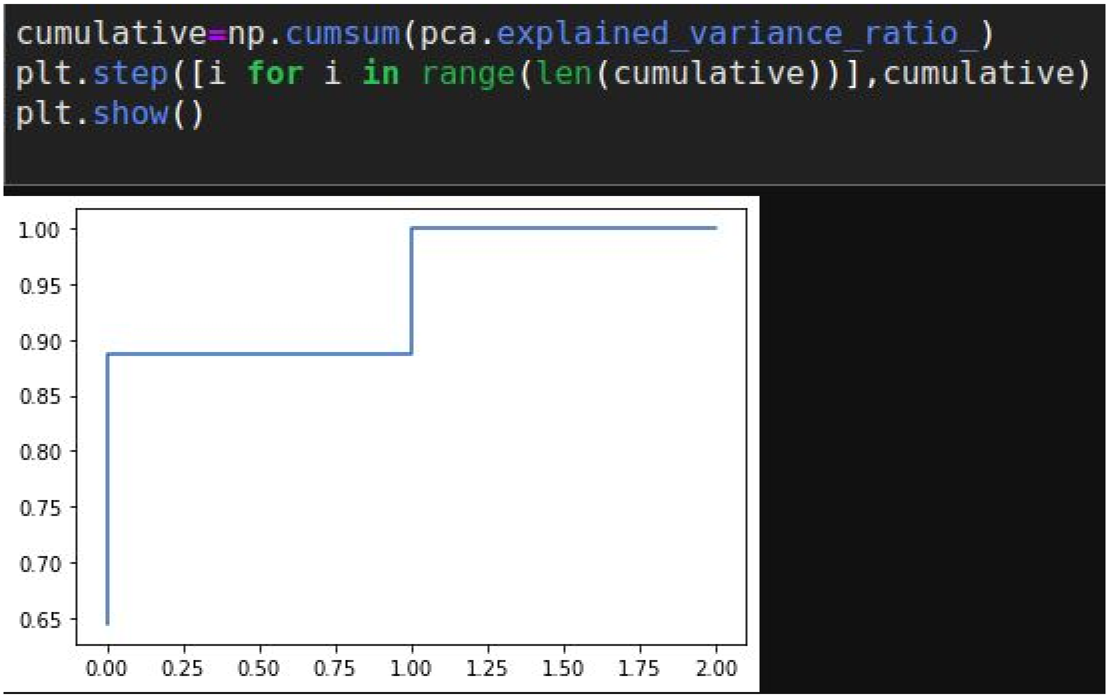


```
from sklearn.decomposition import PCA
pca=PCA().fit(good_data)
print(pca.explained_variance_ratio_)
print()
print(good_data.columns.values.tolist())
print(pca.components_)
```

```
[0.64428499 0.24212251 0.1135925 ]

['Age', 'Annual Income (k$)', 'Spending Score (1-100)']
[[ 0.10863793 -0.01664262 -0.99394206]
 [-0.1267312  -0.99193326  0.00275725]
 [-0.98597008  0.12566393 -0.10987071]]
```

First 2 components cover 88% of the data.

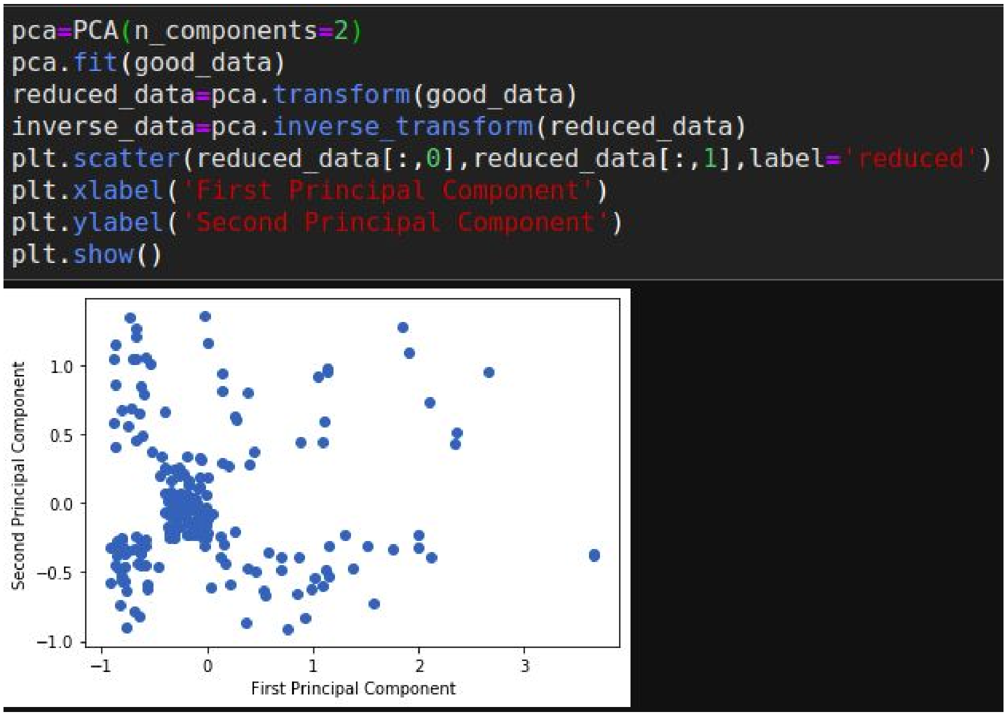
To see the variance in the cumulative manner i will plot a step graph like below:



```
cumulative=np.cumsum(pca.explained_variance_ratio_)
plt.step([i for i in range(len(cumulative))],cumulative)
plt.show()
```

This plot also shows that the first two components consist of ~ **70%** of the data. Hence I have taken no of PCA Components = 2.

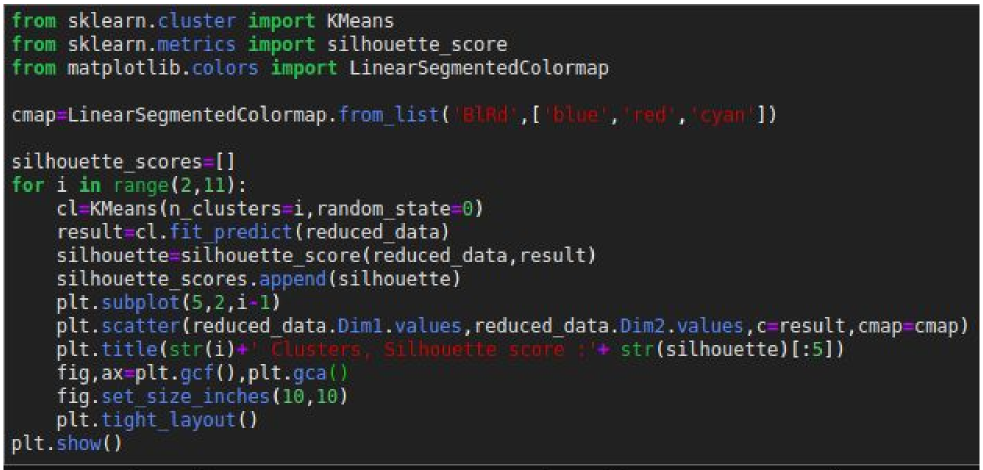
The reduced data can be seen in the graph below:



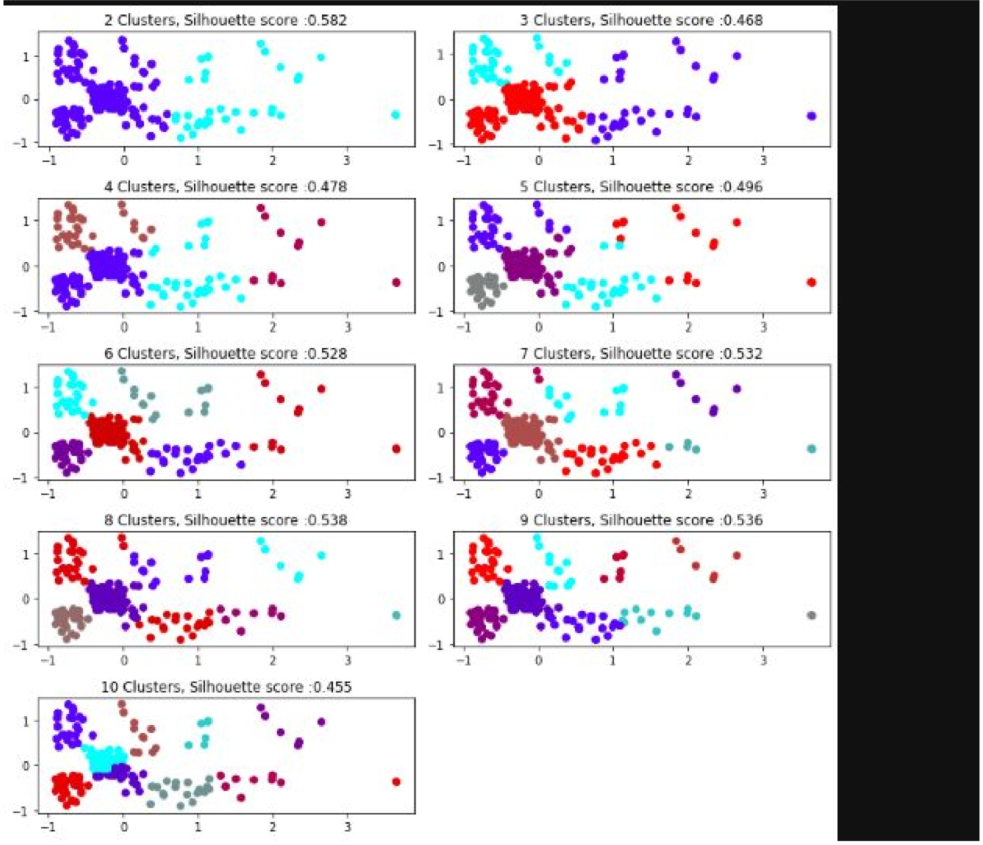
```
pca=PCA(n_components=2)
pca.fit(good_data)
reduced_data=pca.transform(good_data)
inverse_data=pca.inverse_transform(reduced_data)
plt.scatter(reduced_data[:,0],reduced_data[:,1],label='reduced')
plt.xlabel('First Principal Component')
plt.ylabel('Second Principal Component')
plt.show()
```

4. Clustering

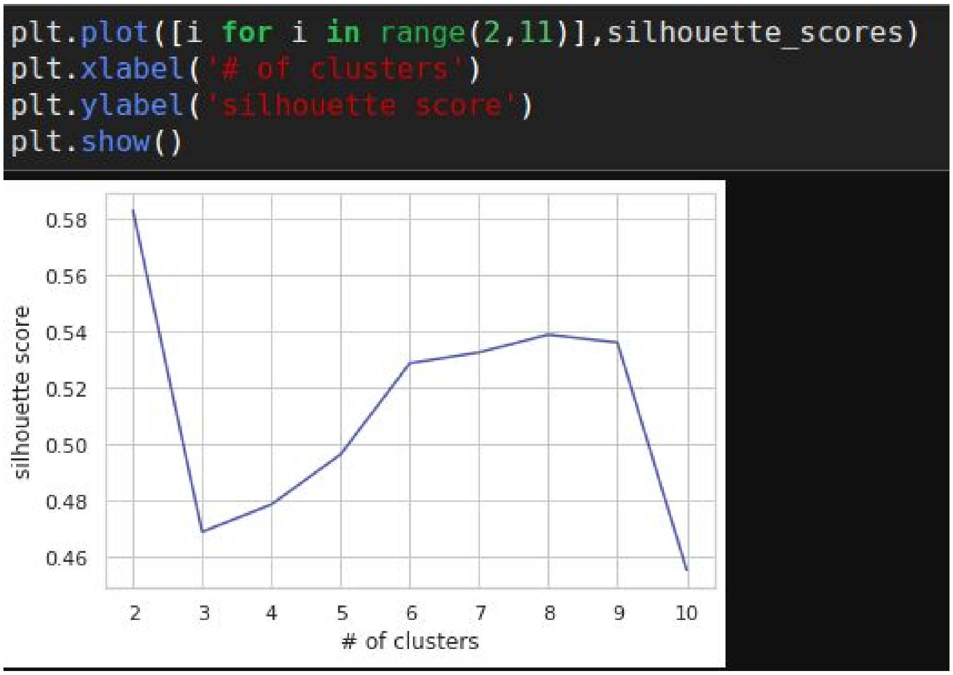
Libraries used: Sklearn, matplotlib
Now, we can perform clustering of the data so that we can extract information related to customer annual spending behaviours.
K-Means:
I will run K-Means from k =2 to k= 10.
I will collect the silhouette scores for each of the results so that I can determine the best number of clusters.



```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from matplotlib.colors import LinearSegmentedColormap

cmap=LinearSegmentedColormap.from_list('BlRd',['blue','red','cyan'])

silhouette_scores=[]
for i in range(2,11):
    cl=KMeans(n_clusters=i,random_state=0)
    result=cl.fit_predict(reduced_data)
    silhouette=silhouette_score(reduced_data,result)
    silhouette_scores.append(silhouette)
    plt.subplot(5,2,i-1)
    plt.scatter(reduced_data.Dim1.values,reduced_data.Dim2.values,c=result,cmap=cmap)
    plt.title(str(i)+' Clusters, Silhouette score :'+ str(silhouette)[:5])
    fig,ax=plt.gcf(),plt.gca()
    fig.set_size_inches(10,10)
    plt.tight_layout()
plt.show()
```

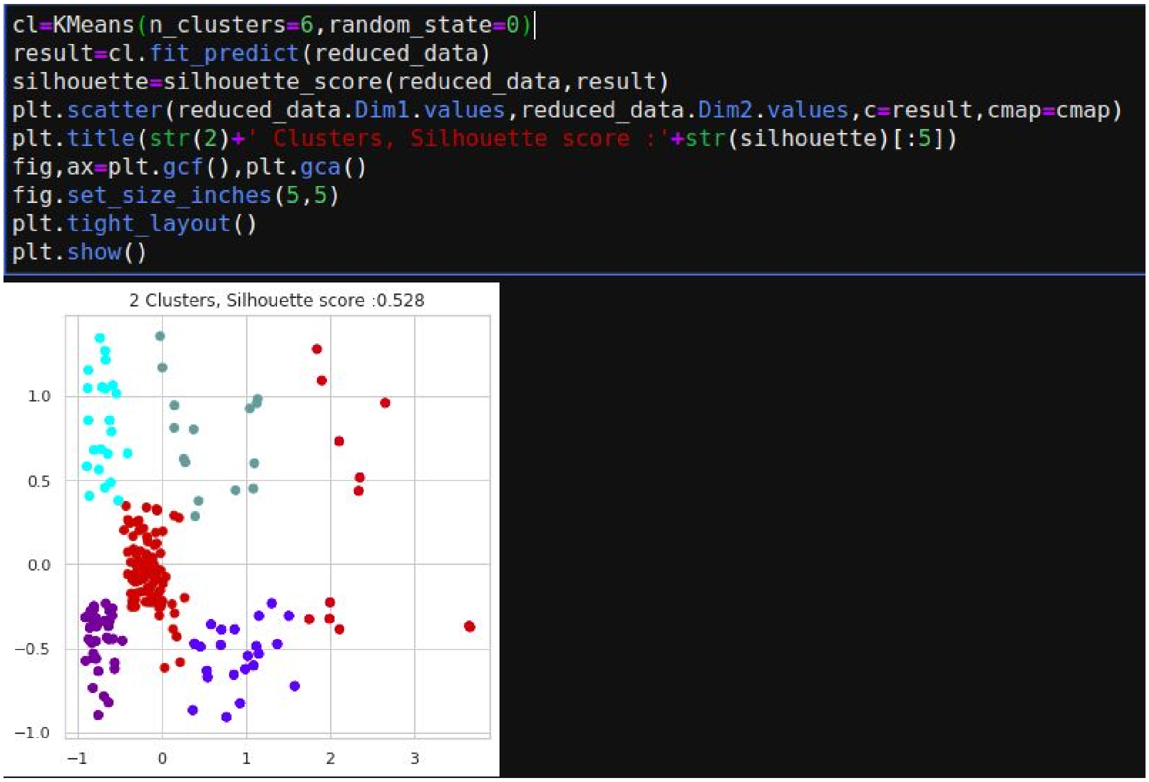


Now the plot for silhouette scores:

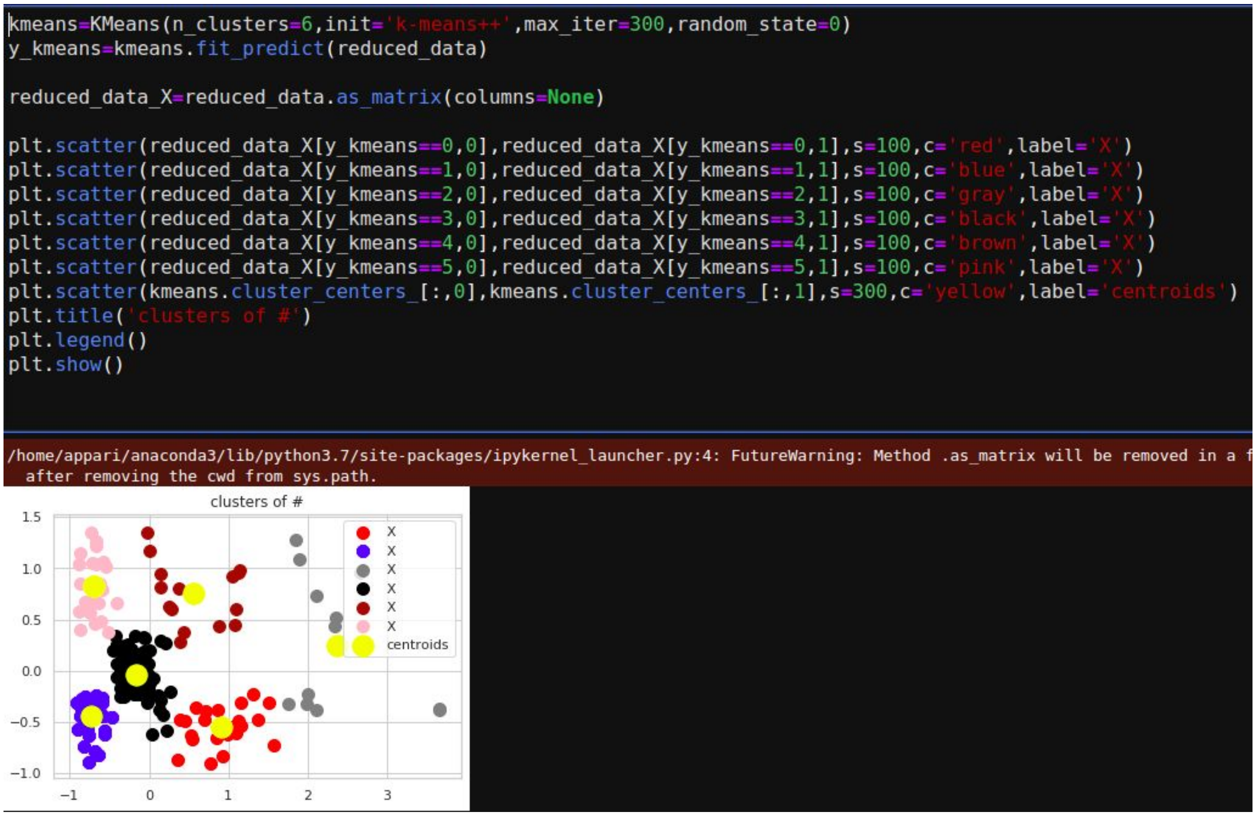


```
plt.plot([i for i in range(2,11)],silhouette_scores)
plt.xlabel('# of clusters')
plt.ylabel('silhouette score')
plt.show()
```

The best number of clusters seem to be 6(or 9) or 2 in this case.



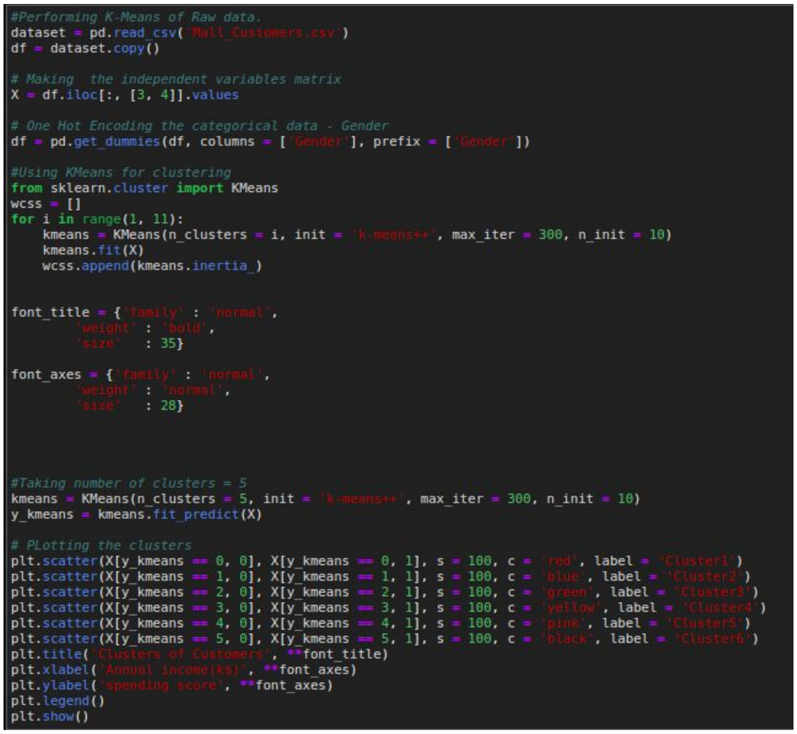
```
cl=KMeans(n_clusters=6,random_state=0)
result=cl.fit_predict(reduced_data)
silhouette=silhouette_score(reduced_data,result)
plt.scatter(reduced_data.Dim1.values,reduced_data.Dim2.values,c=result,cmap=cmap)
plt.title(str(2)+' Clusters, Silhouette score :'+str(silhouette)[:5])
fig,ax=plt.gcf(),plt.gca()
fig.set_size_inches(5,5)
plt.tight_layout()
plt.show()
```



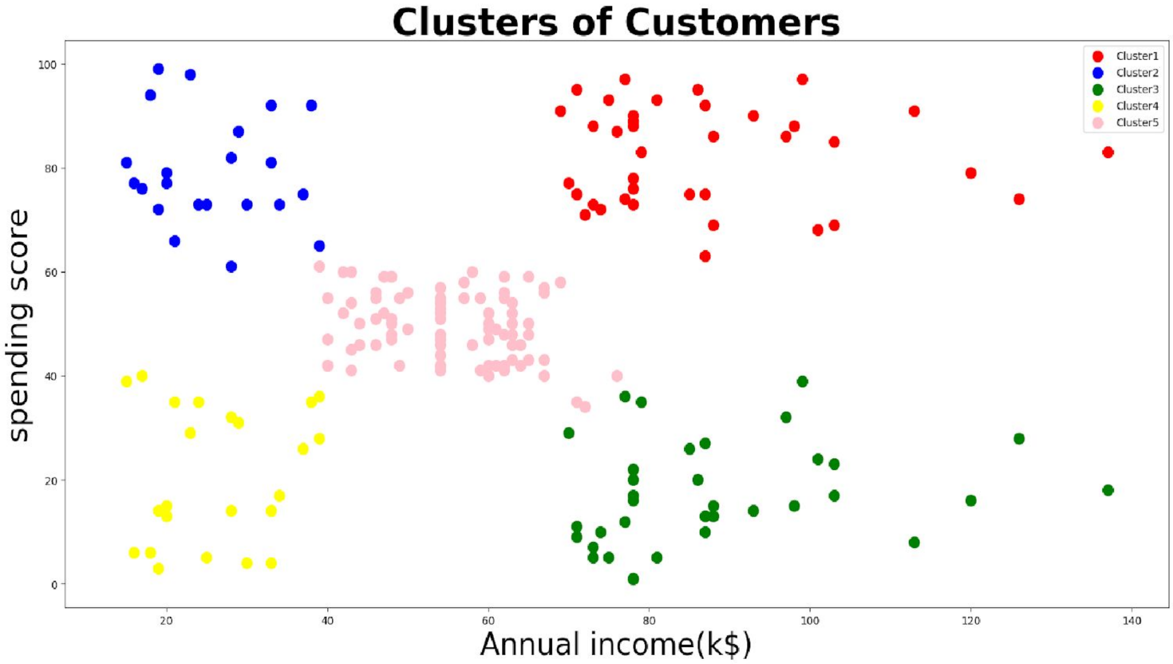
```
kmeans=KMeans(n_clusters=6,init='k-means++',max_iter=300,random_state=0)
y_kmeans=kmeans.fit_predict(reduced_data)

reduced_data_X=reduced_data.as_matrix(columns=None)

plt.scatter(reduced_data_X[y_kmeans==0,0],reduced_data_X[y_kmeans==0,1],s=100,c='red',label='X')
plt.scatter(reduced_data_X[y_kmeans==1,0],reduced_data_X[y_kmeans==1,1],s=100,c='blue',label='X')
plt.scatter(reduced_data_X[y_kmeans==2,0],reduced_data_X[y_kmeans==2,1],s=100,c='gray',label='X')
plt.scatter(reduced_data_X[y_kmeans==3,0],reduced_data_X[y_kmeans==3,1],s=100,c='black',label='X')
plt.scatter(reduced_data_X[y_kmeans==4,0],reduced_data_X[y_kmeans==4,1],s=100,c='brown',label='X')
plt.scatter(reduced_data_X[y_kmeans==5,0],reduced_data_X[y_kmeans==5,1],s=100,c='pink',label='X')
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],s=300,c='yellow',label='centroids')
plt.title('clusters of #')
plt.legend()
plt.show()
```

```
/home/appari/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:4: FutureWarning: Method .as_matrix will be removed in a f
  after removing the cwd from sys.path.
```

Performing K- Means on Raw data:



```
#Performing K-Means of Raw data.
dataset = pd.read_csv('Mall_Customers.csv')
df = dataset.copy()

# Making  the independent variables matrix
X = df.iloc[:, [3, 4]].values

# One Hot Encoding the categorical data - Gender
df = pd.get_dummies(df, columns = ['Gender'], prefix = ['Gender'])

#Using KMeans for clustering
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)


font_title = {'family' : 'normal',
        'weight' : 'bold',
        'size'   : 35}

font_axes = {'family' : 'normal',
        'weight' : 'normal',
        'size'   : 28}




#Taking number of clusters = 5
kmeans = KMeans(n_clusters = 5, init = 'k-means++', max_iter = 300, n_init = 10)
y_kmeans = kmeans.fit_predict(X)

# PLotting the clusters
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'yellow', label = 'Cluster4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'pink', label = 'Cluster5')
plt.scatter(X[y_kmeans == 5, 0], X[y_kmeans == 5, 1], s = 100, c = 'black', label = 'Cluster6')
plt.title('Clusters of Customers', **font_title)
plt.xlabel('Annual income(k$)', **font_axes)
plt.ylabel('spending score', **font_axes)
plt.legend()
plt.show()
```



Analyzing the Results

We can see that the mall customers can be broadly grouped into 5 groups based on their purchases made in the mall.

In cluster 4(yellow colored) we can see people have low annual income and low spending scores, this is quite reasonable as people having low salaries prefer to buy less, in fact, these are the wise people who know how to spend and save money. The shops/mall will be least interested in people belonging to this cluster.

In cluster 2(blue colored) we can see that people have low income but higher spending scores, these are those people who for some reason love to buy products more often even though they have a low income. Maybe it’s because these people are more than satisfied with the mall services. The shops/malls might not target these people that effectively but still will not lose them.

In cluster 5(pink colored) we see that people have average income and an average spending score, these people again will not be the prime targets of the shops or mall, but again they will be considered and other data analysis techniques may be used to increase their spending score.

In cluster 1(red-colored) we see that people have high income and high spending scores, this is the ideal case for the mall or shops as these people are the prime sources of profit. These people might be the regular customers of the mall and are convinced by the mall’s facilities.

In cluster 3(green colored) we see that people have high income but low spending scores, this is interesting. Maybe these are the people who are unsatisfied or unhappy by the mall’s services. These can be the prime targets of the mall, as they have the potential to spend money. So, the mall authorities will try to add new facilities so that they can attract these people and can meet their needs.

Finally, based on our machine learning technique we may deduce that to increase the profits of the mall, the mall authorities should target people belonging to cluster 3 and cluster 5 and should also maintain its standards to keep the people belonging to cluster 1 and cluster 2 happy and satisfied.

References:

1. Jiawei Han, Micheline Kamber, Jian Pei , Data Mining: Concepts and Techniques, 3/e, Morgan,Kaufmann publishers, 2011
    a. 3.2 - Data Cleaning
    b. 3.4.3 - PCA
    c. 10.2.1 - k-Means.
    d. 6.2 - Frequent Itemset Mining Methods
2. https://www.analyticsvidhya.com/blog/2016/07/practical-guide-data-preprocessing-python-scikit-learn/
3. https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-learn-data-science-python-scratch-2/
4. https://www.springboard.com/blog/data-mining-python-tutorial/
5. https://stackoverflow.com/questions/51138686/how-to-use-silhouette-score-in-k-means-clustering-from-sklearn-library
6. https://towardsdatascience.com/decision-trees-introduction-id3-8447fd5213e9
7. https://medium.com/@jyotiyadav99111/selecting-optimal-number-of-clusters-in-kmeans-algorithm-silhouette-score-c0d9ebb11308