Name:Basam Pavani

Pin No:21X05A6708

Branch: Data Science

College: Narasimha Reddy Engineering College

Project Title"

Analysis and prediction of "Mall_Customers.csv" of American mall markets called as phonix mall to find out how many customers are visited to a particular shop. On the basis of this prediction annual income verses spending scores

Disclamer:

In this paricular dataset we assume annual income as a centriod and spending score 1-100 called as datanodes of the cluster

Problem Statement:

The american finance market as per the GDP of 2011 "phonix_trillums mall" as in the first range out of 5.The owner wants to be exit which particular shop or product such in different kind of clusters in entire mall

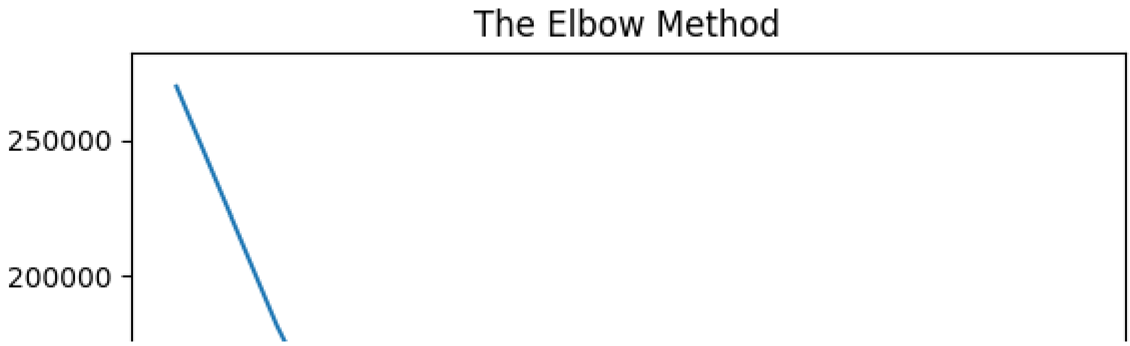
As a data science engineer predict the futuristic financial market for up coming GDP rate based on number of clusters .The client want atleasr top 5 clusters(shops)

```
#import the numpy, matlot, pandas libery's
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
#Read the dataset take variable name called "dataset" only.
dataset=pd.read_csv("Mall_Customers.csv")
# without printing this data add in separet variable as input variable Caqpital X only. lo
#and give the required colum index like[3,4].for this particular dataset.
X=dataset.iloc[:,[3,4]].values
```

```
## <THE ELBOW METHOD>
#from sklearn used "sklearn.cluster" attribute and import KMeans
#Take a distance from from centroid to cluster point with WrapsColumnExpression.
# Assume you have 10 cluster and iterate the for up to range 10 with iterater kmeans++.
# Fit the model if value comes too samlla in range.
#For clustering in wcss ,inertia is adding / appending is required.(kmeans.inertia_)#defal
#Plot the poarticular graph along with the wcss and your range which you taken as input va
#Add title "The Elbow Method".
#Lable x variable as "No of Customers".
#Lable y variable as "WCSS".
#Plot the graph using plt.show().
from sklearn.cluster import KMeans
wcss=[]
for i in range(1,11):
  kmeans=KMeans(n_clusters=i,init="k-means++",random_state=42)
  kmeans.fit(X)
  wcss.append(kmeans.inertia_)
plt.plot(range(1,11),wcss)
plt.title("The Elbow Method")
plt.xlabel("No of Custumers")
plt.ylabel("WCSS")
plt.show()
```

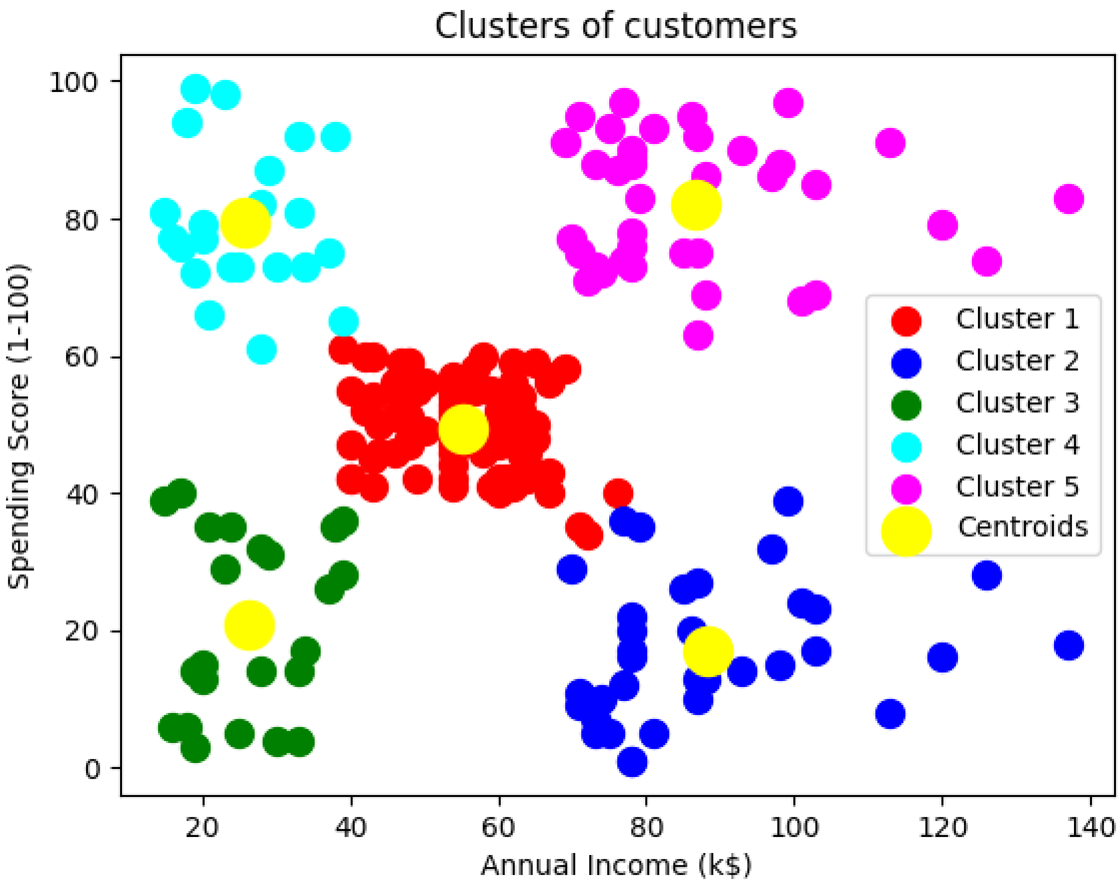
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
```



```
for i in range(1,11):
  kmeans=KMeans(n_clusters=5,init="k-means++",random_state=42)
  y_kmeans=kmeans.fit_predict(X)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning
  warnings.warn(
# Take any no of cluster and run you take 5.
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluste
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Clust
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluste
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Clu


plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'ye
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```



Conclusion:

According to the model basic prediction using Machine Learning Algorithm KMeans Clustering we found that custer-1 which consists Red colour is a highest cluster whixh attach more than 50 Data Nodes.

References:

The model building algorithm develop for all kinds of clusteration values. The yellow spots represents centriod which is max to max only 3

✓ 0s completed at 10:59 AM ● ×