*REPORT*

# Clustering Mall Customers — K-Means (Machine Learning)

ROSHAN D K

### ABSTRACT

Blob With the arrival of the big data era, customer data and data mining analysis have gradually dominated the process of Customer Relationship Management (CRM). This phenomenon indicates that customer data along with the use of information techniques (IT) have become the basis for building a successful CRM strategy. However, some companies can not discover valuable information through a large amount of customer data, which leads to the failure of making appropriate business strategy. Without suitable strategies, the companies may lose the competitive advantage or probably go bankrupt. The purpose of this study is to propose CRM strategies by segmenting customers into VIPs and Non-VIPs and identifying purchase patterns using the the VIPs' transaction data and data mining techniques (K-means clustering and association rules) of online shopping mall in Korea. The results of this paper indicate that 227 customers were segmented into VIPs among 1866 customers. And according to 51,080 transactions data of VIPs, home product and women wear are frequently associated with food, which means that the purchase of home product or women wears mainly affect the purchase of food. Therefore, marketing managers of shopping mall should consider these shopping patterns when they build CRM strategy.

**TABLE OF CONTENTS**

#### 1. INTRODUCTION

Introduction

Applications

Limitations

1. **SOFTWARE REQUIREMENTS**

1. **ALGORITHM**
2. **PROGRAM**

1. **RESULT**

1. **CONCLUSION AND FUTURE SCOPE**

1. **REFERENCES**

# INTRODUCTION

With the arrival of the big data era, the combi-nation of customer data analysis and data min-ing techniques has gradually occupied a leading position in the CRM domain. In fact, analyzing customer data is very helpful for managers to make correct decisions. However, according to the previous literature review, most of the studies focus on large companies while ignore the small and medium sized enterprises(SMEs) environ-ment. In addition, according to the development situation report of SMEs(KOSIS, Korean Statistical Information Service) and Global Bankrupt Report in 2016(Asia Paci󲺻c Partnerships), there are about 67,000 SMEs have been in a “halt production” or “bankrupt” situation compared with 2015 in Ko-rea. They also found out that the average lifetime of In particular, high customer value could bring much more bene󲺻t than the customer value who is low. Analyzing and understanding custom-er behaviors and characteristics is the basis for developing competitive CRM strategies and maxi-mizing the customer value. In addition, according to the marketing axiom of 80/20 rule, 80% of the enterprise’s bene󲺻t are mainly come from 20% of the loyal customers or the VIPs. Therefore, this study presents the following research questions from the perspective of SMEs: (1) ince appropriate data mining tools are one of the best supporting tools for developing di󲺳er-ent CRM decisions and generating suitable CRM strategies(Berson et al. 2000), and the application of data mining tools for CRM is worth pursuing in a customer-centric economy. And this study aims to facilitate SMEs to maintain the loyal cus-tomers or VIPs by providing effective rules and patterns. Un-satisfaction or customers’ churn will lead to unexpected loss, including both of the 󲺻-nancial loss and non󲺻nancial loss. Thus, it is vital for SMEs managers to make appropriate strategies and decision-makings to manage the relationship with their customers

## 

#### Applications of K-Means Clustering:

k-means can be applied to data that has a smaller number of dimensions, is numeric, and is continuous. such as document clustering, identifying crime-prone areas, customer segmentation, insurance fraud detection, public transport data analysis, clustering of IT alerts…etc.

.

## LIMITATIONS:

* The history of the target 󲺻rm is very short. Therefore, the dataset used for analysis is not sufficient to produce more meaningful results.
* Requires complex algorithm like background subtraction for similar background and foreground.
* In the process of this study, it could be found that the lack of data would affect the accuracy of the prediction model

**Software requirements:**

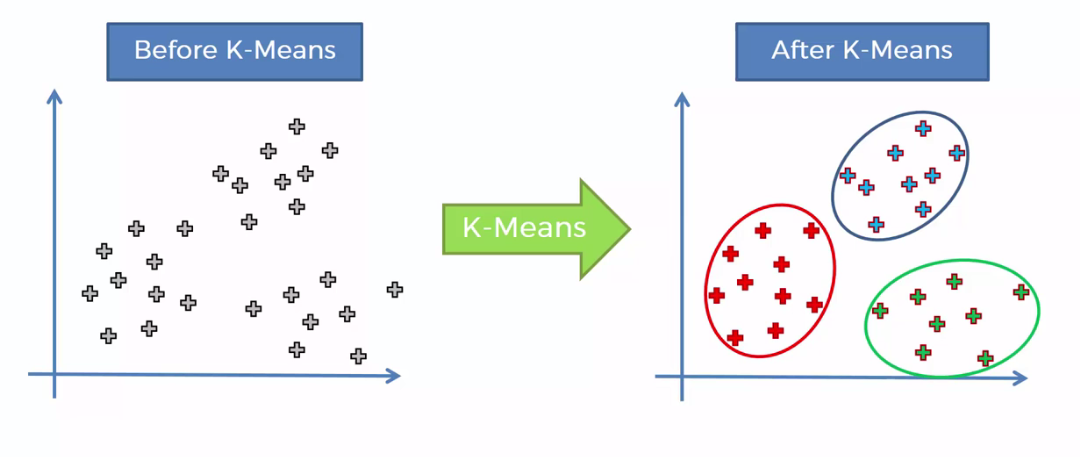
1. Python- programming software
2. Microsoft tools
3. PyCharm- IDE used

## 

## Algorithm:

## K — MEANS CLUSTERING

K-means clustering is a type of unsupervised learning which is used when you have unlabeled data. By using this algorithm you will try to find groups in the data. “k” value represent number of groups.



## PROGRAM:

1. Datasets for training the are taken from Mall\_Customers.csv which runs on Java platform.
2. import numpy as np  
   import matplotlib.pyplot as plt  
   import pandas as pd  
   import random as rd  
   from collections import defaultdict  
   import matplotlib.cm as cm  
     
   dataset=pd.read\_csv('Mall\_Customers.csv')  
   X = dataset.iloc[:, [3, 4]].values  
     
   K=5  
   m=200  
   Centroids=np.array([]).reshape(2,0)  
     
   for i in range(K):  
    rand=rd.randint(0,m-1)  
    Centroids=np.c\_[Centroids,X[rand]]  
      
     
   num\_iter=100  
   Output=defaultdict()  
   Output={}  
   for n in range(num\_iter):  
      
    EuclidianDistance=np.array([]).reshape(m,0)  
    for k in range(K):  
    tempDist=np.sum((X-Centroids[:,k])\*\*2,axis=1)  
    EuclidianDistance=np.c\_[EuclidianDistance,tempDist]  
      
    C=np.argmin(EuclidianDistance,axis=1)+1  
      
    Y={}  
    for k in range(K):  
    Y[k+1]=np.array([]).reshape(2,0)  
    for i in range(m):  
    Y[C[i]]=np.c\_[Y[C[i]],X[i]]  
      
    for k in range(K):  
    Y[k+1]=Y[k+1].T  
      
      
    for k in range(K):  
    Centroids[:,k]=np.mean(Y[k+1],axis=0)  
      
    Output=Y  
     
   plt.scatter(X[:,0],X[:,1],c='black',label='unclustered data')  
   plt.xlabel('Income')  
   plt.ylabel('Number of transactions')  
   plt.legend()  
   plt.title('Plot of data points')  
   plt.show()  
     
   color=['red','blue','green','cyan','magenta']  
   labels=['cluster1','cluster2','cluster3','cluster4','cluster5']  
   for k in range(K):  
    plt.scatter(Output[k+1][:,0],Output[k+1][:,1],c=color[k],label=labels[k])  
   plt.scatter(Centroids[0,:],Centroids[1,:],s=300,c='yellow',label='Centroids')  
   plt.xlabel('Income')  
   plt.ylabel('Number of transactions')  
   plt.legend()  
   plt.show()

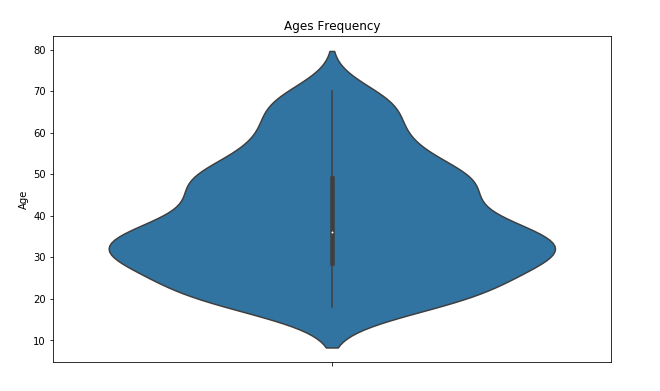
# Output:

1. The above code generates an output of several frames and are saved in a designated folder as shown below. Hence, the dataset is autogenerated.

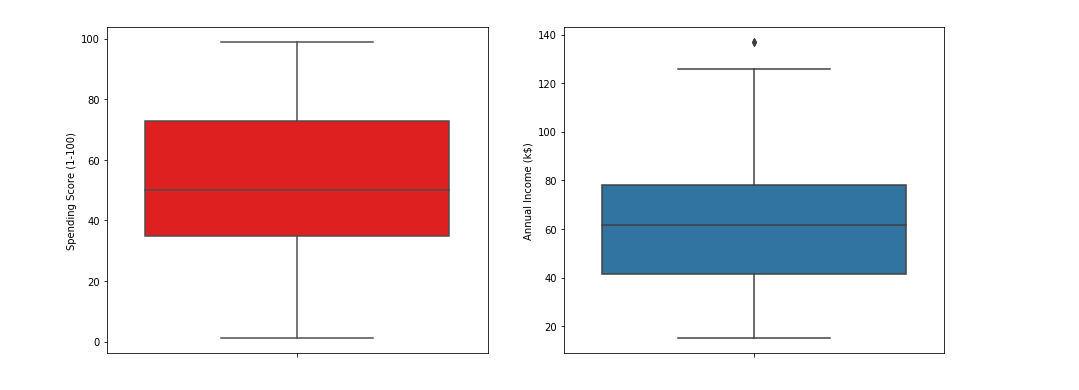


**Training the model:**

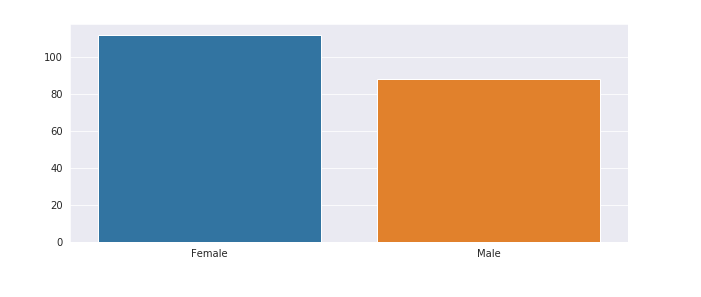
df.drop(["CustomerID"], axis = 1, inplace=True)  
plt.figure(figsize=(10,6))  
plt.title("Ages Frequency")  
sns.axes\_style("dark")  
sns.violinplot(y=df["Age"])  
plt.show()



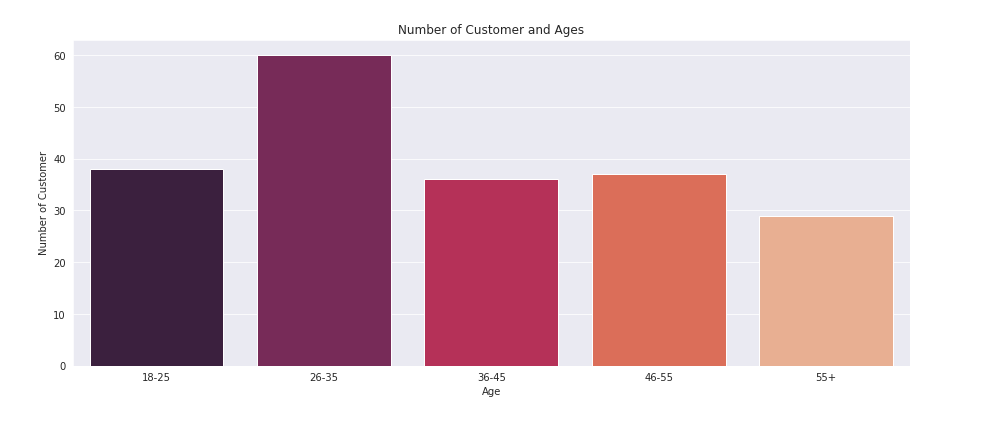
plt.figure(figsize=(15,6))  
plt.subplot(1,2,1)  
sns.boxplot(y=df["Spending Score (1-100)"], color="red")  
plt.subplot(1,2,2)  
sns.boxplot(y=df["Annual Income (k$)"])  
plt.show()



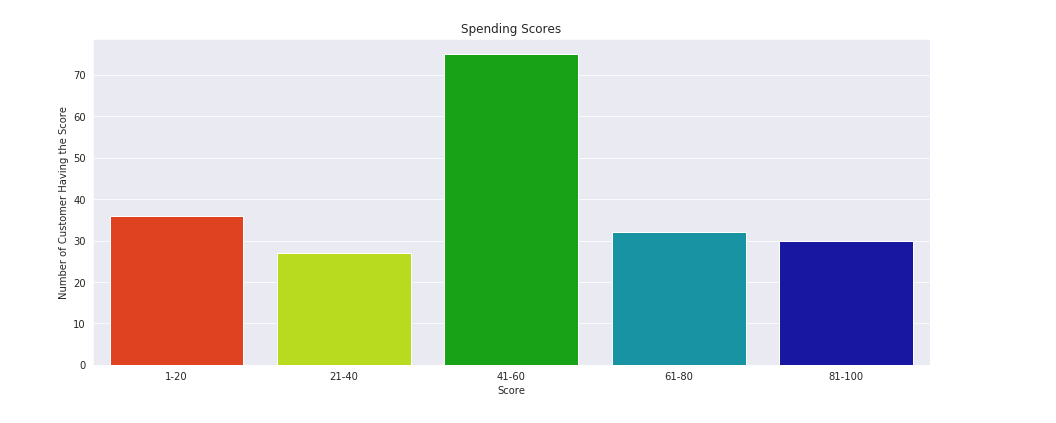
genders = df.Gender.value\_counts()  
sns.set\_style("darkgrid")  
plt.figure(figsize=(10,4))  
sns.barplot(x=genders.index, y=genders.values)  
plt.show()



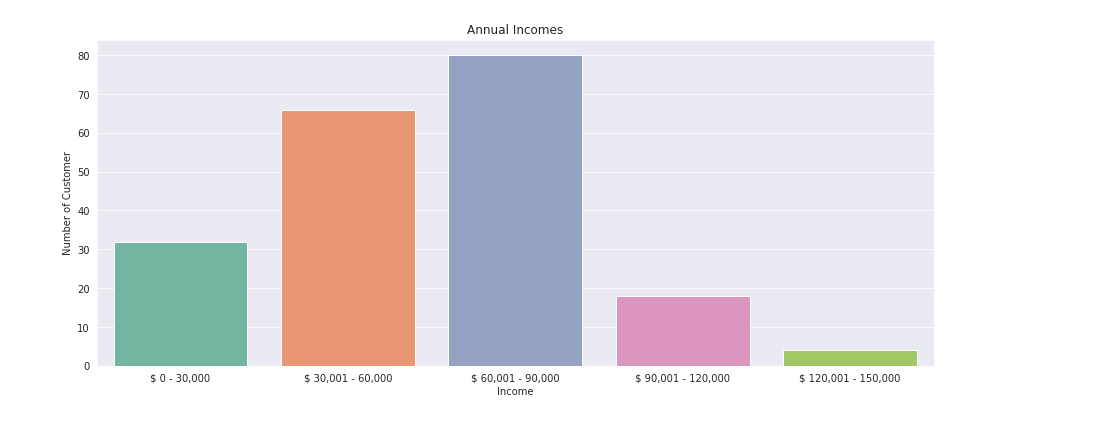
age18\_25 = df.Age[(df.Age <= 25) & (df.Age >= 18)]  
age26\_35 = df.Age[(df.Age <= 35) & (df.Age >= 26)]  
age36\_45 = df.Age[(df.Age <= 45) & (df.Age >= 36)]  
age46\_55 = df.Age[(df.Age <= 55) & (df.Age >= 46)]  
age55above = df.Age[df.Age >= 56]  
x = ["18-25","26-35","36-45","46-55","55+"]  
y = [len(age18\_25.values),len(age26\_35.values),len(age36\_45.values),len(age46\_55.values),len(age55above.values)]  
plt.figure(figsize=(15,6))  
sns.barplot(x=x, y=y, palette="rocket")  
plt.title("Number of Customer and Ages")  
plt.xlabel("Age")  
plt.ylabel("Number of Customer")  
plt.show()



ss1\_20 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 1) & (df["Spending Score (1-100)"] <= 20)]  
ss21\_40 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 21) & (df["Spending Score (1-100)"] <= 40)]  
ss41\_60 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 41) & (df["Spending Score (1-100)"] <= 60)]  
ss61\_80 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 61) & (df["Spending Score (1-100)"] <= 80)]  
ss81\_100 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 81) & (df["Spending Score (1-100)"] <= 100)]  
ssx = ["1-20", "21-40", "41-60", "61-80", "81-100"]  
ssy = [len(ss1\_20.values), len(ss21\_40.values), len(ss41\_60.values), len(ss61\_80.values), len(ss81\_100.values)]  
plt.figure(figsize=(15,6))  
sns.barplot(x=ssx, y=ssy, palette="nipy\_spectral\_r")  
plt.title("Spending Scores")  
plt.xlabel("Score")  
plt.ylabel("Number of Customer Having the Score")  
plt.show()



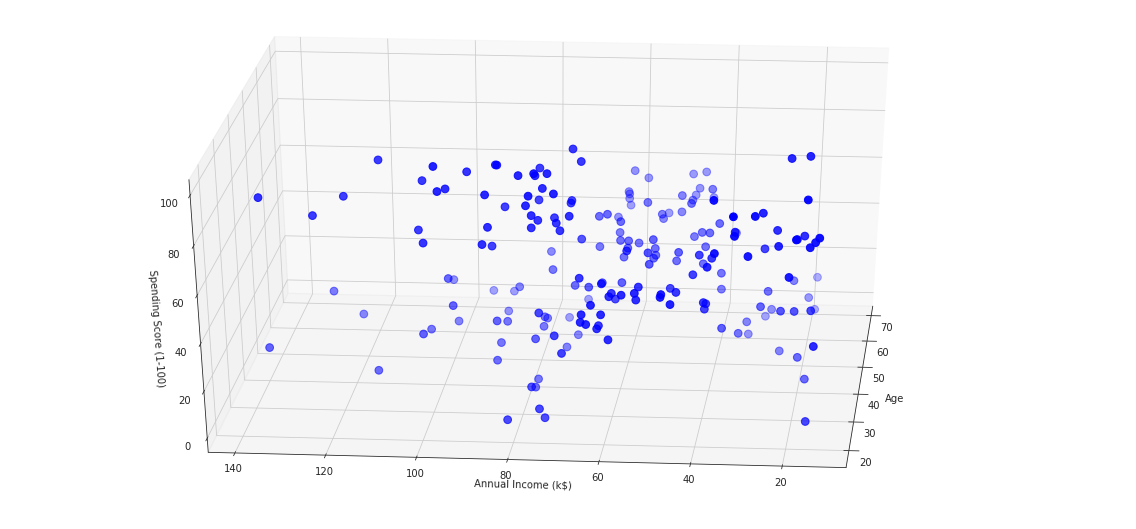
ai0\_30 = df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 0) & (df["Annual Income (k$)"] <= 30)]  
ai31\_60 = df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 31) & (df["Annual Income (k$)"] <= 60)]  
ai61\_90 = df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 61) & (df["Annual Income (k$)"] <= 90)]  
ai91\_120 = df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 91) & (df["Annual Income (k$)"] <= 120)]  
ai121\_150 = df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 121) & (df["Annual Income (k$)"] <= 150)]  
aix = ["$ 0 - 30,000", "$ 30,001 - 60,000", "$ 60,001 - 90,000", "$ 90,001 - 120,000", "$ 120,001 - 150,000"]  
aiy = [len(ai0\_30.values), len(ai31\_60.values), len(ai61\_90.values), len(ai91\_120.values), len(ai121\_150.values)]  
plt.figure(figsize=(15,6))  
sns.barplot(x=aix, y=aiy, palette="Set2")  
plt.title("Annual Incomes")  
plt.xlabel("Income")  
plt.ylabel("Number of Customer")  
plt.show()



**Testing the model:**

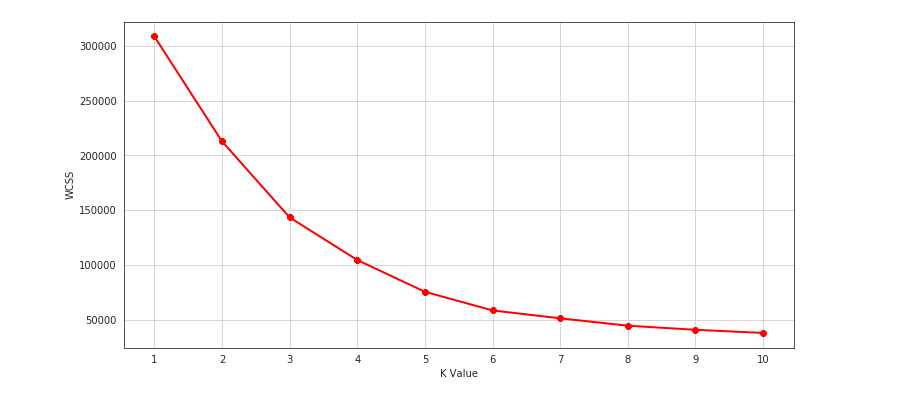
We will use Age, Annual Income and Spending Score for clustering customers. Let’s look how our plot is seen without clustering.

km = KMeans(n\_clusters=5)  
clusters = km.fit\_predict(df.iloc[:,1:])  
df["label"] = clusters  
from mpl\_toolkits.mplot3d import Axes3D  
import matplotlib.pyplot as plt  
import numpy as np  
import pandas as pd  
fig = plt.figure(figsize=(20,10))  
ax = fig.add\_subplot(111, projection='3d')  
ax.scatter(df.Age[df.label == 0], df["Annual Income (k$)"][df.label == 0], df["Spending Score (1-100)"][df.label == 0], c='blue', s=60)  
ax.scatter(df.Age[df.label == 1], df["Annual Income (k$)"][df.label == 1], df["Spending Score (1-100)"][df.label == 1], c='red', s=60)  
ax.scatter(df.Age[df.label == 2], df["Annual Income (k$)"][df.label == 2], df["Spending Score (1-100)"][df.label == 2], c='green', s=60)  
ax.scatter(df.Age[df.label == 3], df["Annual Income (k$)"][df.label == 3], df["Spending Score (1-100)"][df.label == 3], c='orange', s=60)  
ax.scatter(df.Age[df.label == 4], df["Annual Income (k$)"][df.label == 4], df["Spending Score (1-100)"][df.label == 4], c='purple', s=60)  
ax.view\_init(30, 185)  
plt.xlabel("Age")  
plt.ylabel("Annual Income (k$)")  
ax.set\_zlabel('Spending Score (1-100)')  
plt.show()



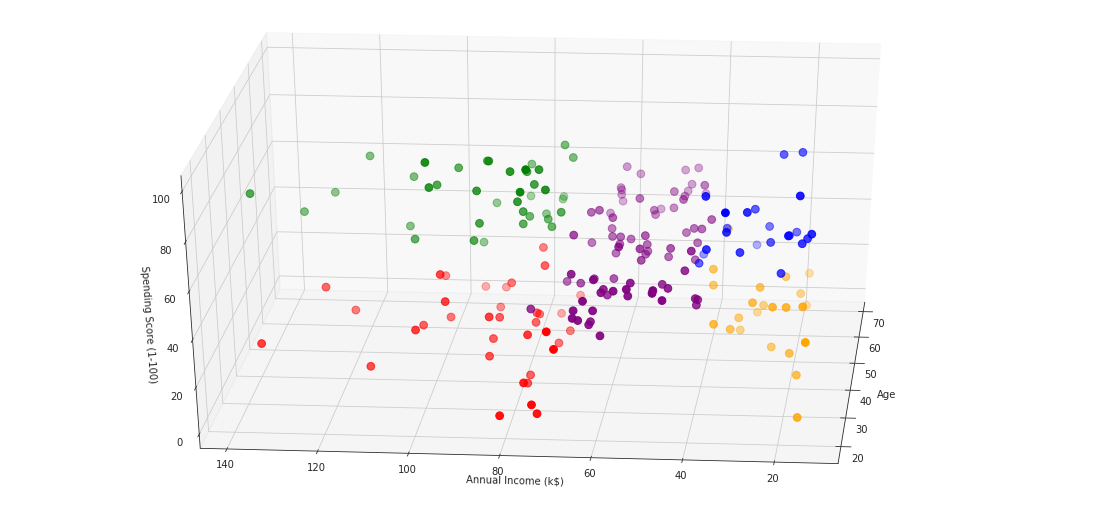
Now we will try to find what “k” value we should use. We will find it out with “elbow method”.

from sklearn.cluster import KMeans  
wcss = []  
for k in range(1,11):  
 kmeans = KMeans(n\_clusters=k, init="k-means++")  
 kmeans.fit(df.iloc[:,1:])  
 wcss.append(kmeans.inertia\_)  
plt.figure(figsize=(12,6))   
plt.grid()  
plt.plot(range(1,11),wcss, linewidth=2, color="red", marker ="8")  
plt.xlabel("K Value")  
plt.xticks(np.arange(1,11,1))  
plt.ylabel("WCSS")  
plt.show()



We used elbow method in the above and we may say 5 will be our number of cluster. Let’s use K-Means and see how our plot will look like.

km = KMeans(n\_clusters=5)  
clusters = km.fit\_predict(df.iloc[:,1:])  
df["label"] = clusters  
from mpl\_toolkits.mplot3d import Axes3D  
import matplotlib.pyplot as plt  
import numpy as np  
import pandas as pd  
fig = plt.figure(figsize=(20,10))  
ax = fig.add\_subplot(111, projection='3d')  
ax.scatter(df.Age[df.label == 0], df["Annual Income (k$)"][df.label == 0], df["Spending Score (1-100)"][df.label == 0], c='blue', s=60)  
ax.scatter(df.Age[df.label == 1], df["Annual Income (k$)"][df.label == 1], df["Spending Score (1-100)"][df.label == 1], c='red', s=60)  
ax.scatter(df.Age[df.label == 2], df["Annual Income (k$)"][df.label == 2], df["Spending Score (1-100)"][df.label == 2], c='green', s=60)  
ax.scatter(df.Age[df.label == 3], df["Annual Income (k$)"][df.label == 3], df["Spending Score (1-100)"][df.label == 3], c='orange', s=60)  
ax.scatter(df.Age[df.label == 4], df["Annual Income (k$)"][df.label == 4], df["Spending Score (1-100)"][df.label == 4], c='purple', s=60)  
ax.view\_init(30, 185)  
plt.xlabel("Age")  
plt.ylabel("Annual Income (k$)")  
ax.set\_zlabel('Spending Score (1-100)')  
plt.show()



## References:

1. Tsai, C. F., and Chen, M. Y. 2010. “Variables Selection by Association Rules for Customer Churn Prediction of Multimedia on Demand,” Expert Systems with Applications (37:3), pp. 2006-2015.19.
2. Turban, E., Sharda, R., and Delen, D. 2011. Decision Support and Business Intelligence Systems, Pearson Education India. 20. Wu, C. H., Kao, S. C., Su, Y. Y., and Wu, C. C. 2005.
3. “Targeting Customers Via Discovery Knowledge for the Insurance Industry,” Ex-pert Systems with Applications (29:2), pp. 291-299.