A PROJECT REPORT

ON

CUSTOMER SEGMENTATION

(By K-means Algorithm)

FOR

EXPOSYS DATA LABS

SUBMITTED BY

- **Name**: Akshay Kumar Vadlamani
- **Email** : akshaykumarvadlamani@gmail.com
- **Contact No**:8522814944

ABSTRACT

In this project, we will perform one of the most essential applications of machine learning, Customer Segmentation by using K-Means Clustering Algorithm. In this project, we will implement customer segmentation in Python. Whenever you need to find your best customer, customer segmentation is the ideal methodology.

Then we will explore the data upon which we will be building our segmentation model. Also, in this data science project, we will see the descriptive analysis of our data and then implement several versions of the K-means algorithm.

Furthermore, through the data collected, we can gain a deeper understanding of customer preferences as well as the requirements for discovering valuable segments that would reap them maximum profit. This way, we can strategize the marketing techniques more efficiently and minimize the possibility of risk to the investment

2. Keywords

Clustering, Elbow Method, K-Means Algorithm, Customer Segmentation, Visualization.

TABLE OF CONTENTS

1.Introduction

1.1. Introduction to Prediction models

We all know that customer satisfaction is key to boosting a company's performance, but organizations still strive to utilize the increasing availability of data to satisfy customers. The report illustrates how machine learning and data science techniques can be employed to assess and evaluate customer satisfaction. It is necessary to present steps to develop customer-driven prediction models, starting from problem framing, to data exploratory analysis, data transformation, ML training, and recommendations. Predictive analytics involves certain manipulations of data from existing data sets to identify new trends and patterns. These trends and patterns are then used to predict future outcomes and trends. By performing predictive analysis, we can predict future trends and performance. It is also defined as the prognostic analysis; the word prognostic means prediction. Predictive analytics uses data, statistical algorithms and machine learning techniques to identify the probability of future outcomes based on historical data. In predictive analysis, we use historical data to predict future outcomes. Thus, predictive analysis plays a vital role in various fields. It improves decision-making, helps increase the profit rates of businesses, and reduces risk by identifying them early. Predictive analysis is used in various fields like Online Retail, and Improvised market campaigning.

1.2. Objective of the work

The objective is to analyse our expenditure on the start-ups and then know the profit put them. The r programming language will be quite helpful in such a situation where we need to find a profit based on how much we are spending in the market and for the market. In a nutshell, will help to find out the profit based on the amount we spend from the 50_ Start-up dataset

2.EXISTING METHOD

The existing method is storing customer data through paperwork and computer software (digital data) is increasing day by day. At end of the day they will analyse their data as how many things are sold or actual customer count etc. By analysing the collected data they got to know who is beneficial to their business and increase their sales. It requires more time and more paperwork. Also, it is not much effective solution to find the desired customers data.

3. THE PROPOSED METHOD WITH ARCHITECTURE

Gentle working of Prediction Models:

Step 1: Sample data

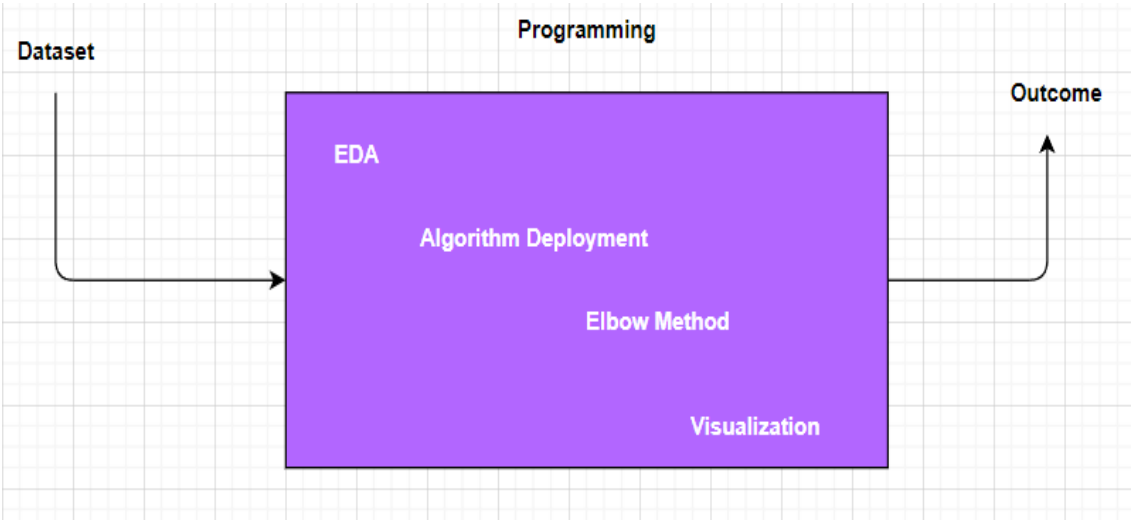
Data is information about the problem that you are working on. Imagine we want to identify the species of flower from the measurements of a flower. The data is comprised of four flower measurements in centimetres, these are the columns of the data. Each row of data is one example of a flower that has been measured and its known species. The problem we are solving is to create a model from the sample data that can tell us which species a flower belongs to from its measurements alone.

Step 2: Learn a Model

This problem described above is called supervised learning. The goal of a supervised learning algorithm is to take some data with a known relationship and create a model of those relationships. In this case, the output is a category and we call this type of problem a classification problem. If the output was a numerical value, we would call it a regression problem. The algorithm does the learning. The model contains the learned relationships.

Step 3: Make Predictions

We don't need to keep the training data as the model has summarized the relationships contained within it. The reason we keep the model learned from data is that we want to use it to make predictions. Our model will read the input perform a calculation of some kind with its internal numbers and make a prediction. The prediction may not be perfect, but if you have good sample data and a robust model learned from that data, it will be quite accurate



4. METHODOLOGY

1. First of all we will import all the necessary libraries or modules (pandas, numpy, seaborn,sklearn).

2. Then we will read dataset and anyalse whether it contains any null values, missing values and duplicate values. So we will fix them by dropping or fixing the value with their means, medians etc which is technically named as Data Preprocessing.

3. We will deploy our model algorithm K-Means Clustering, which divides the data into group of clusters based on similar characteristics.

4. To find no.of clusters we will use elbow method.

5. Finally, we will visualize our data using matplot, which concludes the customers divided into groups who are similar to each other on their group.

5.IMPLEMENTATION

5.1 Overview of a Dataset

This is a mall customer segmentation data which contains 5 columns and 200 rows.

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
...	...	...	...	...	...
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

200 rows × 5 columns

5.2 Exploratory Data Analysis

It deals with the data pre-processing , whether it contains any missing values or null values. There after we will see the information and description of the dataset.

5.2.1 Information of the dataset

#df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

As here it overview the information of the data. And it gives it doesn't contain any null values.

As we will remove the irrelevant data which is customer id.

df.drop(["CustomerID"], axis=1, inplace=True)

```
# so here customer data is not required to our analysis. We will drop it.

df.drop(["CustomerID"], axis=1, inplace=True)
```

```
# printing data frame again (Now, CustomerID column is removed)

df
```

	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	Male	19	15	39
1	Male	21	15	81
2	Female	20	16	6
3	Female	23	16	77
4	Female	31	17	40
5	Female	22	17	76
6	Female	35	18	6
7	Female	23	18	94

5.2.2 Description of the data

#df.describe()

	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000
mean	38.850000	60.560000	50.200000
std	13.969007	26.264721	25.823522
min	18.000000	15.000000	1.000000
25%	28.750000	41.500000	34.750000
50%	36.000000	61.500000	50.000000
75%	49.000000	78.000000	73.000000
max	70.000000	137.000000	99.000000

It describes about the count which counts the no of rows in it, mean of the columns, standard deviations, maximum and minimum and percentiles etc.

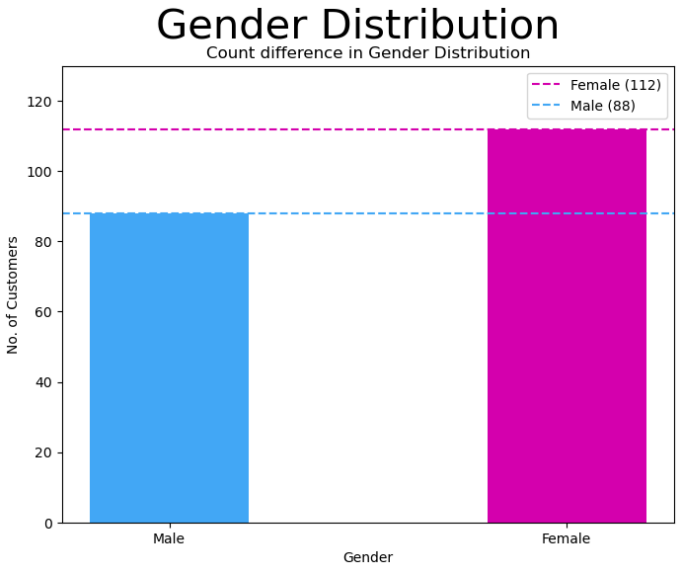
5.3 Gender plot Analysis

Here it overview the gender analysis

```
labels=data['Gender'].unique()
values=data['Gender'].value_counts(ascending=True)


fig, (ax0) = plt.subplots(ncols=1,figsize=(8,6))
bar = ax0.bar(x=labels, height=values, width=0.4, align='center',
color=['#42a7f5','#d400ad'])
ax0.set(title='Count difference in Gender Distribution',xlabel='Gender',
ylabel='No. of Customers')
ax0.set_ylim(0,130)
```

```
ax0.axhline(y=data['Gender'].value_counts()[0], color='#d400ad', linestyle='--',
label=f'Female ({data.Gender.value_counts()[0]})')
ax0.axhline(y=data['Gender'].value_counts()[1], color='#42a7f5', linestyle='--',
label=f'Male ({data.Gender.value_counts()[1]})')
ax0.legend()
fig.suptitle('Gender Distribution', fontsize=30);
```

So we label the x-axis as Gender and y-axis as Count and we plot it by using barplot.



From the plot we will conclude that there are more female customers than the male customers i.e female customers are more than 100 whereas male customers are nearly

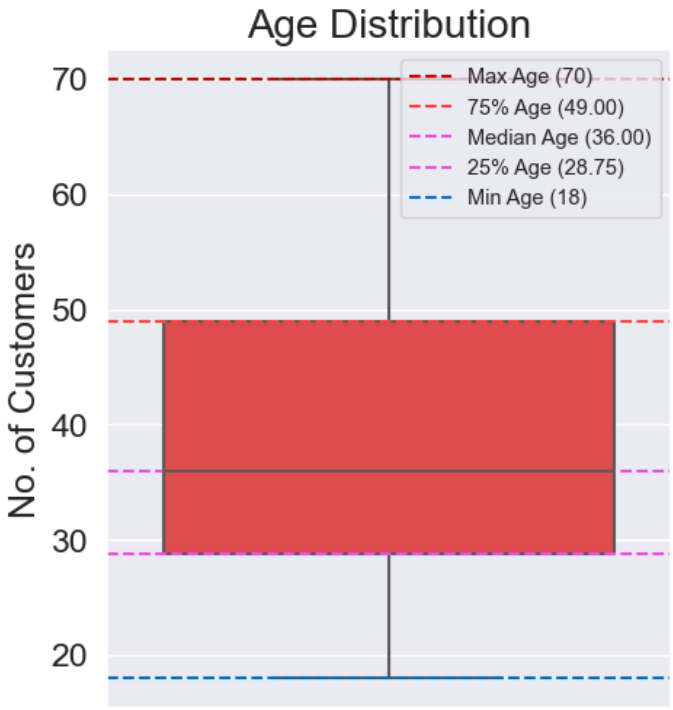
5.4 Age plot

We will use distplot for the distribution of age of the customers.

```
fig, ax = plt.subplots(figsize=(5,6))
```

```
sns.set(font_scale=1.5)
ax = sns.boxplot(y=data["Age"], color="#f73434")
ax.axhline(y=data['Age'].max(), linestyle='--',color='#c90404', label=f'Max Age
({data.Age.max()})')
ax.axhline(y=data['Age'].describe()[6], linestyle='--',color='#f74343',
label=f'75% Age ({data.Age.describe()[6]:.2f})')
ax.axhline(y=data['Age'].median(), linestyle='--',color='#eb50db',
label=f'Median Age ({data.Age.median():.2f})')
ax.axhline(y=data['Age'].describe()[4], linestyle='--',color='#eb50db',
label=f'25% Age ({data.Age.describe()[4]:.2f})')
ax.axhline(y=data['Age'].min(), linestyle='--',color='#046ebf', label=f'Min Age
({data.Age.min()})')
ax.legend(fontsize='xx-small', loc='upper right')
ax.set_ylabel('No. of Customers')

plt.title('Age Distribution', fontsize = 20)
plt.show()
```

So we label X-axis as range of age and y-axis as count.



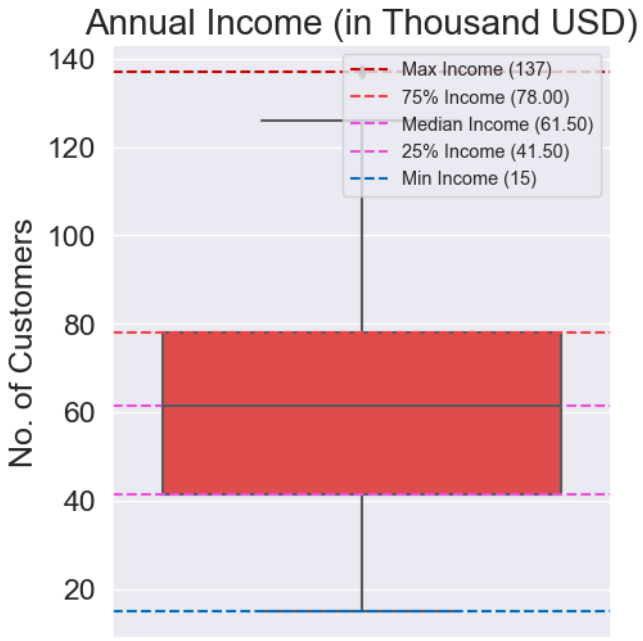
From the above boxplot, we can conclude that a large amount of ages are between 30 and 35. Min Age is 18, Max Age is 70. By comparing the age distribution of the customers, we can conclude that most of the customers were within the band between 30 to 50, where the mean is around 35 years old

5.5 Annual Income and Spending Score analysis

As we will use scatterplot and labelled x-axis as Annual Income(k$) and y-axis as Spending Score(1-100)

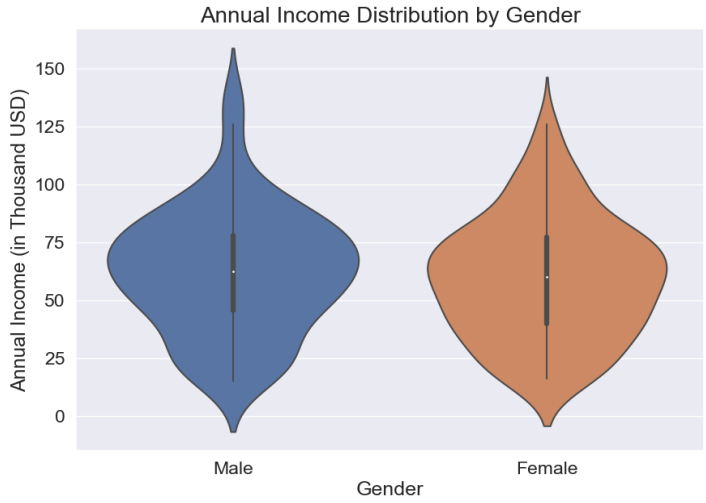
```
fig, ax = plt.subplots(figsize=(5,6))
sns.set(font_scale=1.5)
ax = sns.boxplot(y=data["Annual_Income"], color="#f73434")
ax.axhline(y=data["Annual_Income"].max(), linestyle='--',color='#c90404',
label=f'Max Income ({data.Annual_Income.max()})')
ax.axhline(y=data["Annual_Income"].describe()[6], linestyle='--
',color='#f74343', label=f'75% Income
({data.Annual_Income.describe()[6]:.2f})')
```

```
ax.axhline(y=data["Annual_Income"].median(), linestyle='--',color='#eb50db',
label=f'Median Income ({data.Annual_Income.median():.2f})')
ax.axhline(y=data["Annual_Income"].describe()[4], linestyle='--
',color='#eb50db', label=f'25% Income
({data.Annual_Income.describe()[4]:.2f})')
ax.axhline(y=data["Annual_Income"].min(), linestyle='--',color='#046ebf',
label=f'Min Income ({data.Annual_Income.min()})')
ax.legend(fontsize='xx-small', loc='upper right')
ax.set_ylabel('No. of Customers')

plt.title('Annual Income (in Thousand USD)', fontsize = 20)
plt.show()
```

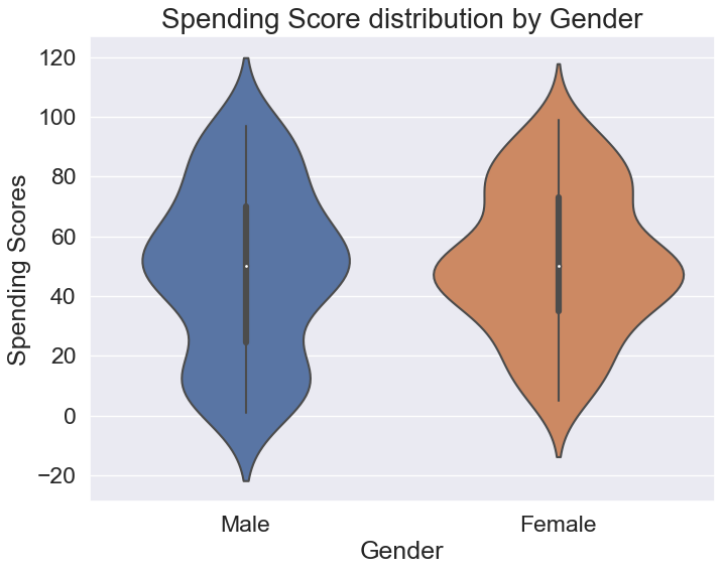


Characteristic relations:

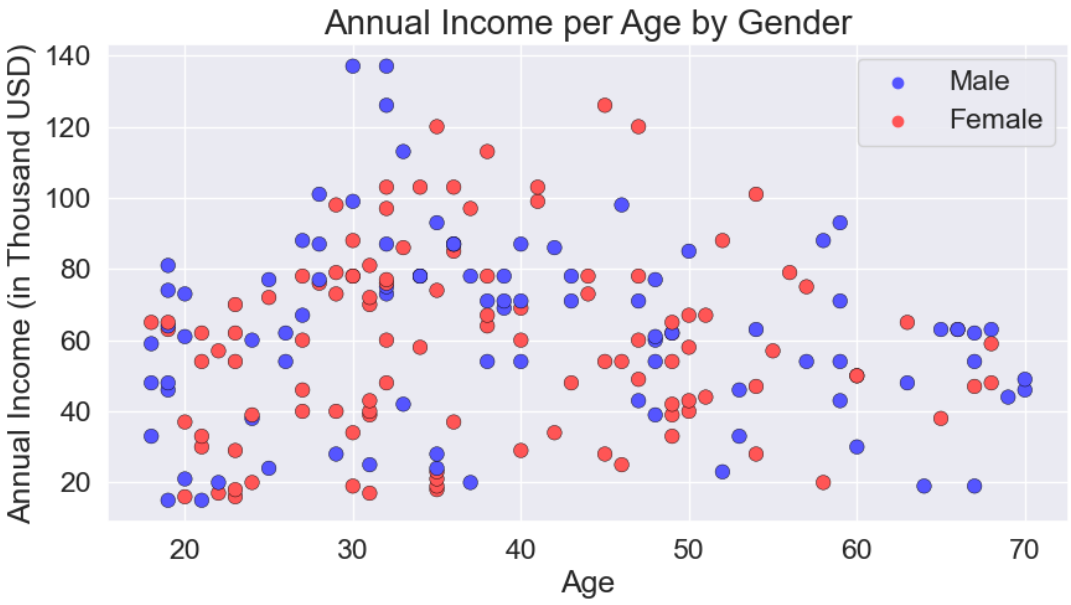
Annual income vs Age and Gender

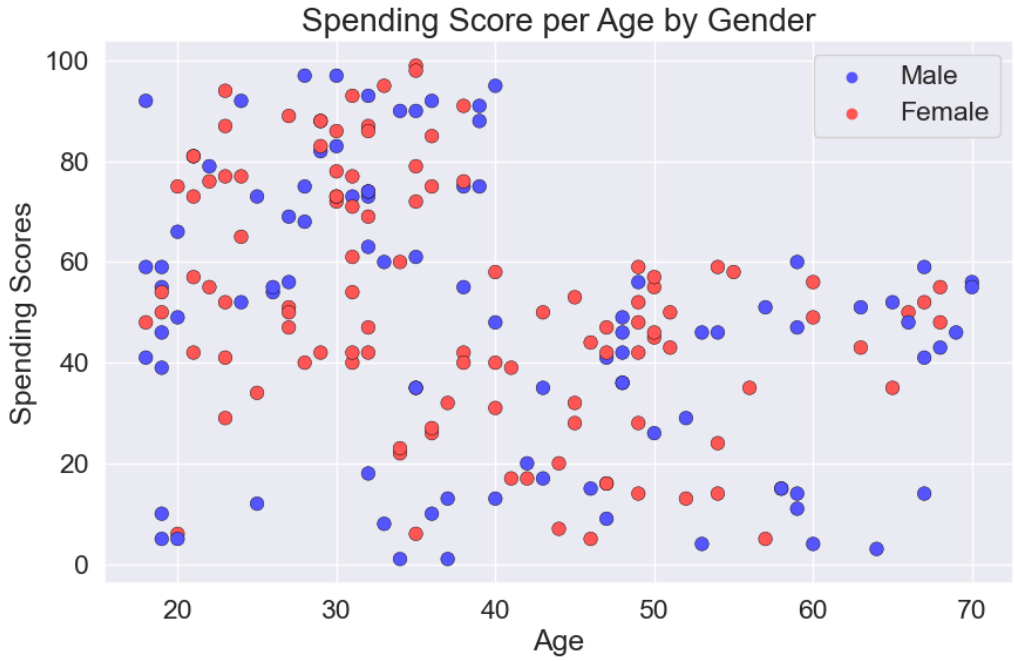


From the plot we can observe the difference in Annual Income between Male and Female .



From the plot we can observe the difference in Spending Score between Male and Female





5.6 Elbow Method

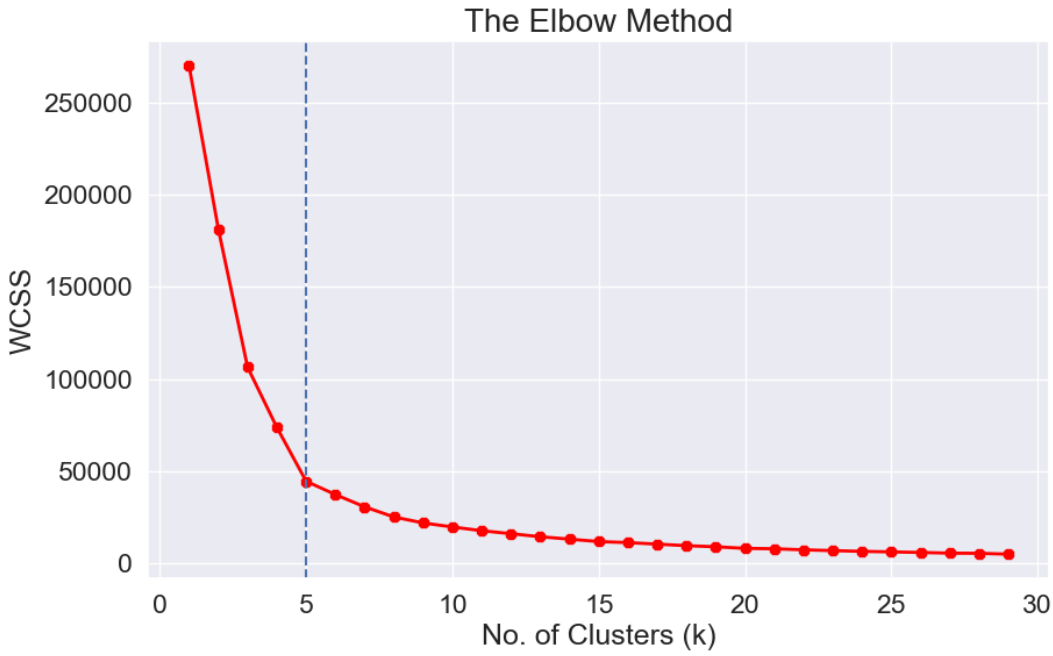
The elbow method is based on the observation that increasing the number of clusters can help to reduce the sum of within-cluster variance of each cluster. This is because having more clusters allows one to capture finer groups of data objects that are more similar to each other. To define the optimal clusters, Firstly, we use the clustering algorithm

for various values of k. This is done by ranging k from 1 to 10 clusters. Then we calculate the total intra-cluster sum of square. Then, we proceed to plot intra-cluster sum of square based on the number of clusters. The plot denotes the approximate number of clusters required in our model. The optimum clusters can be found from the graph where there is a bend in the graph.

$$WCSS = \sum_{i \in n} (X_i - Y_i)^2$$

where Yi is centroid for observation Xi. The main goal is to maximize number of clusters and in limiting case each data point becomes its own cluster centroid.

```
fig, ax = plt.subplots(figsize=(10,6))
ax = plt.plot(range(1,30),wcss, linewidth=2, color="red", marker
="8")
plt.axvline(x=5, ls='--')
plt.ylabel('WCSS')
plt.xlabel('No. of Clusters (k)')
plt.title('The Elbow Method', fontsize = 20)
plt.show()
```



It is clear, that the optimal number of clusters for our data are 5, as the slope of the curve is not steep enough after it. When we observe this curve, we see that last elbow comes at k = 5, it would be difficult to visualize the elbow if we choose the higher range

5.7 Fitting the Algorithm

```
from sklearn.cluster import KMeans

kms = KMeans(n_clusters=5, init='k-means++')
kms.fit(clustering_data)
```

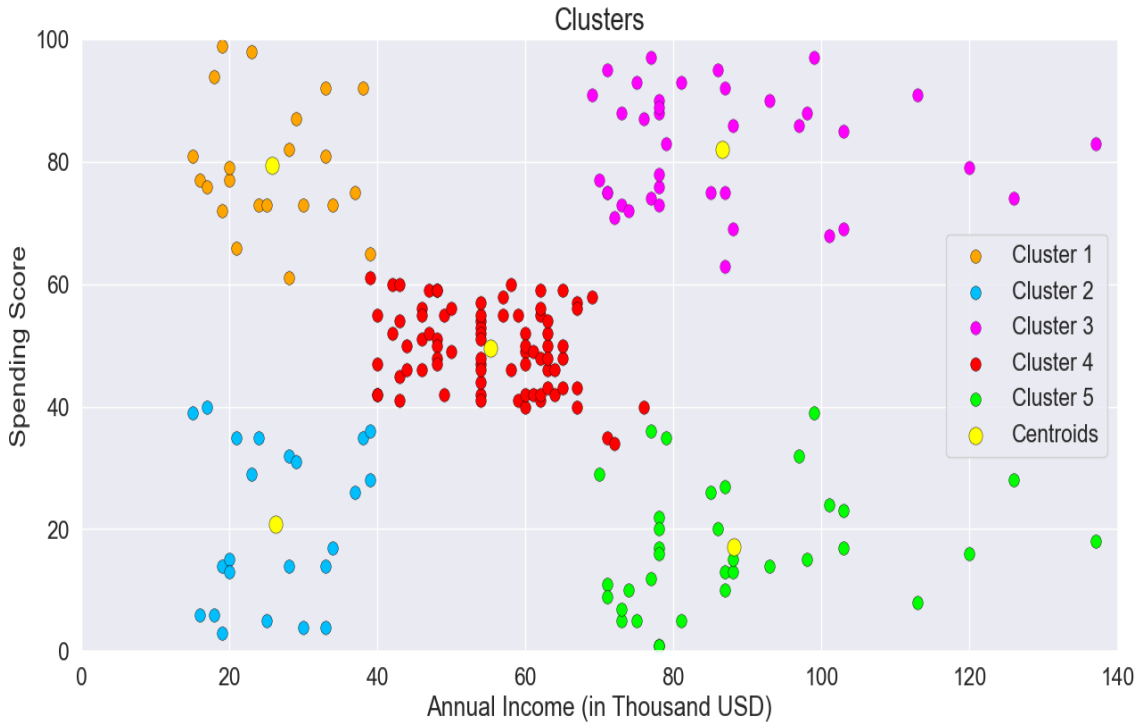
As here we initialized the k-means as km with 5 clusters and we will fit it. There after we will predict the data and store it in y. And then we will add new column named as Cluster and data as y.

5.8 Visualization the clusters

Visualizing the clusters based on Annual Income and Spending Score of the customers. As here we plot a graph named as Clusters of Customers to visualize the data in terms of groups or cluster.

```
fig, ax = plt.subplots(figsize=(15,7))

plt.scatter(x=clusters[clusters['Cluster_Prediction'] == 4]['Annual_Income'],
            y=clusters[clusters['Cluster_Prediction'] == 4]['Spending_Score'],
            s=70,edgecolor='black', linewidth=0.3, c='orange', label='Cluster 1')

plt.scatter(x=clusters[clusters['Cluster_Prediction'] == 0]['Annual_Income'],
            y=clusters[clusters['Cluster_Prediction'] == 0]['Spending_Score'],
            s=70,edgecolor='black', linewidth=0.3, c='deepskyblue', label='Cluster
2')

plt.scatter(x=clusters[clusters['Cluster_Prediction'] == 2]['Annual_Income'],
            y=clusters[clusters['Cluster_Prediction'] == 2]['Spending_Score'],
            s=70,edgecolor='black', linewidth=0.2, c='Magenta', label='Cluster 3')

plt.scatter(x=clusters[clusters['Cluster_Prediction'] == 1]['Annual_Income'],
            y=clusters[clusters['Cluster_Prediction'] == 1]['Spending_Score'],
            s=70,edgecolor='black', linewidth=0.3, c='red', label='Cluster 4')

plt.scatter(x=clusters[clusters['Cluster_Prediction'] == 3]['Annual_Income'],
            y=clusters[clusters['Cluster_Prediction'] == 3]['Spending_Score'],
            s=70,edgecolor='black', linewidth=0.3, c='lime', label='Cluster 5')
```

```
plt.scatter(x=kms.cluster_centers_[:, 0], y=kms.cluster_centers_[:, 1], s = 120, c
= 'yellow', label = 'Centroids',edgecolor='black', linewidth=0.3)
plt.legend(loc='right')
plt.xlim(0,140)
plt.ylim(0,100)
plt.xlabel('Annual Income (in Thousand USD)')
plt.ylabel('Spending Score')
plt.title('Clusters', fontsize = 20)
plt.show()
```



So from the above one we observed that the there are 5 clusters which are named as 1, 2, 3, 4,5

- Cluster 1 which is at centre, average annual income with average spending score.
- Cluster 2 which is at top right, highest annual income with highest spending score.
- Cluster 3 which is at top left, lowest annual income with highest spending score.
- Cluster 4 which is at bottom right, high annual income with low spending score.
- Cluster 5 which is at bottom left, lowest annual income with lowest spending score.

6. Conclusion

So we concluded that the ,

- The Highest income , high spending can be target these type of customers as they earn more money and spend as much as they want.
- Highest income, low spending can be target these type of customers by asking feedback and advertising the product in a better way.
- Average income, Average spending may or may not be beneficial to the mall owners of this type of customers.
- Low income, High spending can be target these type of customers by providing them with low-cost EMI’s etc.
- Low income, Low spending don’t target these type of customers because they earn a bit and spend some amount of money.