Exposys Data Labs

Yelahanka, Bengaluru, Karnataka 560064

A PROJECT REPORT ON

“Customer Segmentation”

A Project Work Report Submitted in partial fulfillment of the requirement for the position of

Data Science Intern

Submitted by

MOHAMMED

TOUSEEF

May 2021

ABSTRACT

The zeitgeist of the modern era is innovation, where everyone is embroiled into the competition to be better than others

Today's business runs based on such innovation by offering different discounts to a certain group of people. Manually grouping people may take months together. This is where data science and machine learning models come into the big picture applying various algorithms to find out the hidden patterns in the dataset. This elude concept of which segment to target is made unequivocal by applying segmentation. The process of segmenting the customers with similar behaviors into the same segment and with different patterns into different segments is called customer segmentation. I have used the k-Means algorithm an unsupervised learning algorithm is been implemented to segment the customers into 5 groups depending on there spending score and annual income. To find the optimal centroids in the k-Means algorithm I have used elbow method and silhouette methods, For in-depth analysis, I have even considered one more feature that is age with spending score and annual income resulting in 6 groups with varying parameters on 200 samples of data .

Table of Contents

Abstract

INTRODUCTION

According to data collected by the US Census, the average person in the US 15 or older spent about 10 hours shopping for consumer goods per month in 2018

As different suppliers or e-commerce websites show up every day of our lives. We, customers, have a wide variety of products and suppliers to choose in an instant. E-commerce websites want us to stick to the singular brand so they do customer segmentation which helps them to give unique discounts to a group of people with returns of more profit

Customer Segmentation

Sometimes referred to as market segmentation, customer segmentation is a method of analyzing a client base and grouping customers into categories or segments which share particular attributes. Key differentials in segmentation include age, gender, education, location, spending patterns, and socio-economic group. Relevant differentials are those which are expected to influence customer behavior concerning a business. The selected criteria are used to create customer segments with similar values, needs, and wants.

When planning a targeted marketing campaign, it is also necessary to differentiate customers within these groupings according to their preferred means of communication.

Customer segmentation is an essential tool in customer relationship management, enabling businesses to market effectively to their customers. Businesses are expected to understand their customers and demonstrate their customer insights by sending only relevant, targeted communications to their customers. Customers want to feel valued and be treated as individuals, yet for anything other than perhaps the smallest of businesses this level of customer knowledge is impossible to achieve.

Segmentation also allows businesses to channel their resources appropriately. High-value customers who purchase frequently and generate more revenue usually belong in a segment that is allocated a higher level of marketing spend.

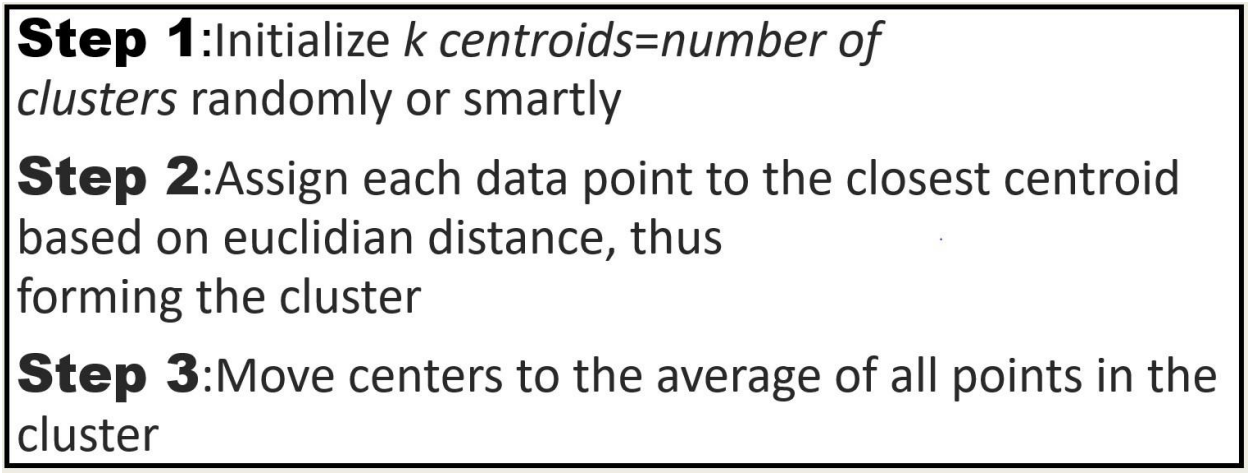
Analyzing customer demographics and psychographics gives layers of insights that help anticipate customers' needs and plan new products and services. This in turn enable

marketers to target more accurately those customers or prospects who would be most interested in them.

k-Means Clustering

K-means is an unsupervised clustering algorithm designed to partition unlabeled data into a certain number (that's the "*K*") of distinct groupings. In other words, k-means finds observations that share important characteristics and classifies them together into clusters. A good clustering solution is one that finds clusters such that the observations within each cluster are more similar than the clusters themselves.

k-Means Algorithm



The Elbow Method

In cluster analysis, the elbow method is a heuristic used in determining the number of clusters in a data set. The method consists of plotting the explained variation as a function of the number of clusters, and picking the elbow of the curve as the number of clusters to use

Using the "elbow" or "knee of a curve" as a cutoff point is a common heuristic in mathematical optimization to choose a point where diminishing returns are no longer worth the additional cos

To calculate the centroids the formula is

$$C = \frac{\sum_{i=1}^{m} x_i}{m}$$

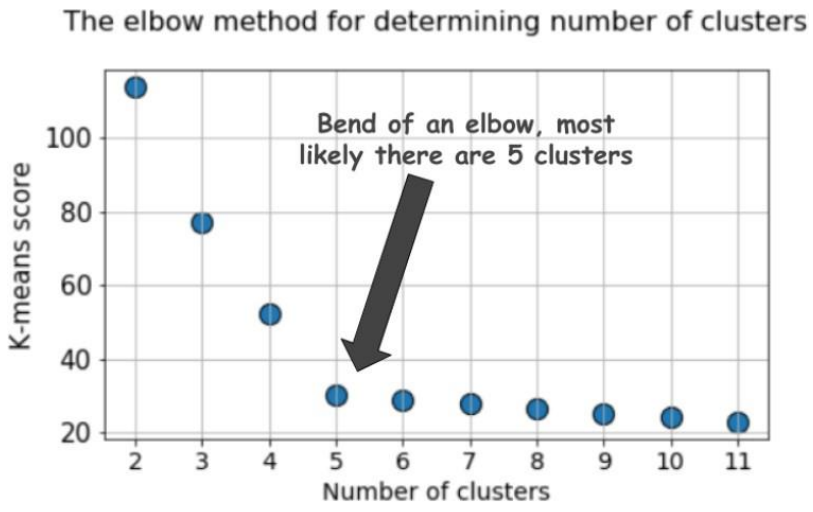
To calculate the sum of squared errors the formula is

$$J(x) = \sum_{i=1}^{m} ||x_i - C||^2$$

To minimize the sum of squared error across *all* centroids $c_j \in \{c_1, c_2, \dots, ck\}$ for *all* observations $x_i \in \{x_1, x_2, \dots, x_n\}$ the formula is

$$J(x) = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^{(j)} - c_j||^2$$

A typical elbow method plot looks like the following:



The score is, in general, a measure of the input data on the k-means objective function i.e. some form of intra-cluster distance relative to inner-cluster distance

Sometime the elbow method plot might be pretty confusing and Scikit-learn's k-means estimator, a score method is readily available for this purpose.

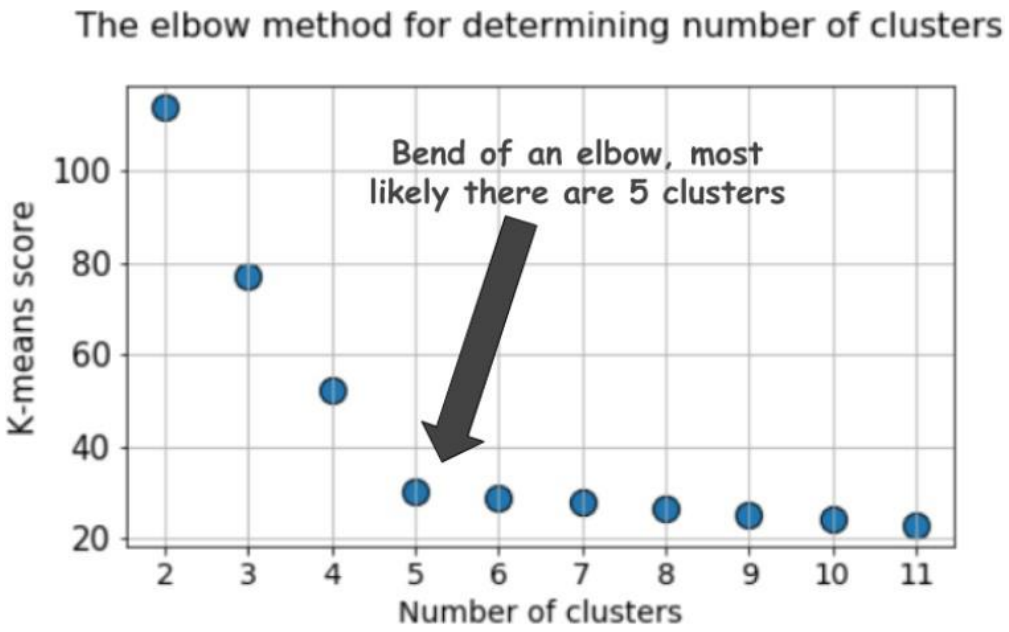
Silhouette coefficient Method

The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample.

The formula to find the silhouette coefficient is

$$s_i = \frac{b - a}{\max(a, b)}$$

A typical silhouette coefficient plot looks like the following:



"Silhouette coefficient exhibits a peak characteristic as compared to the gentle bend in the elbow method. This is easier to visualize and reason with".

Existing Method

Some of the basic and existing methods for customer segmentation where they don't use any data mining techniques or clustering algorithms to segments customers few of methods are that divide email lists into groups based on common features that tend to predict buying habits, such as demographics or interests, in order to better serve the customer. This is called a priori segmentation– a priori is Latin for from the former, and basically means that you've deducted these segments based on anecdotal knowledge or observed trends in your marketing efforts. Here are some a priori segments you can use:

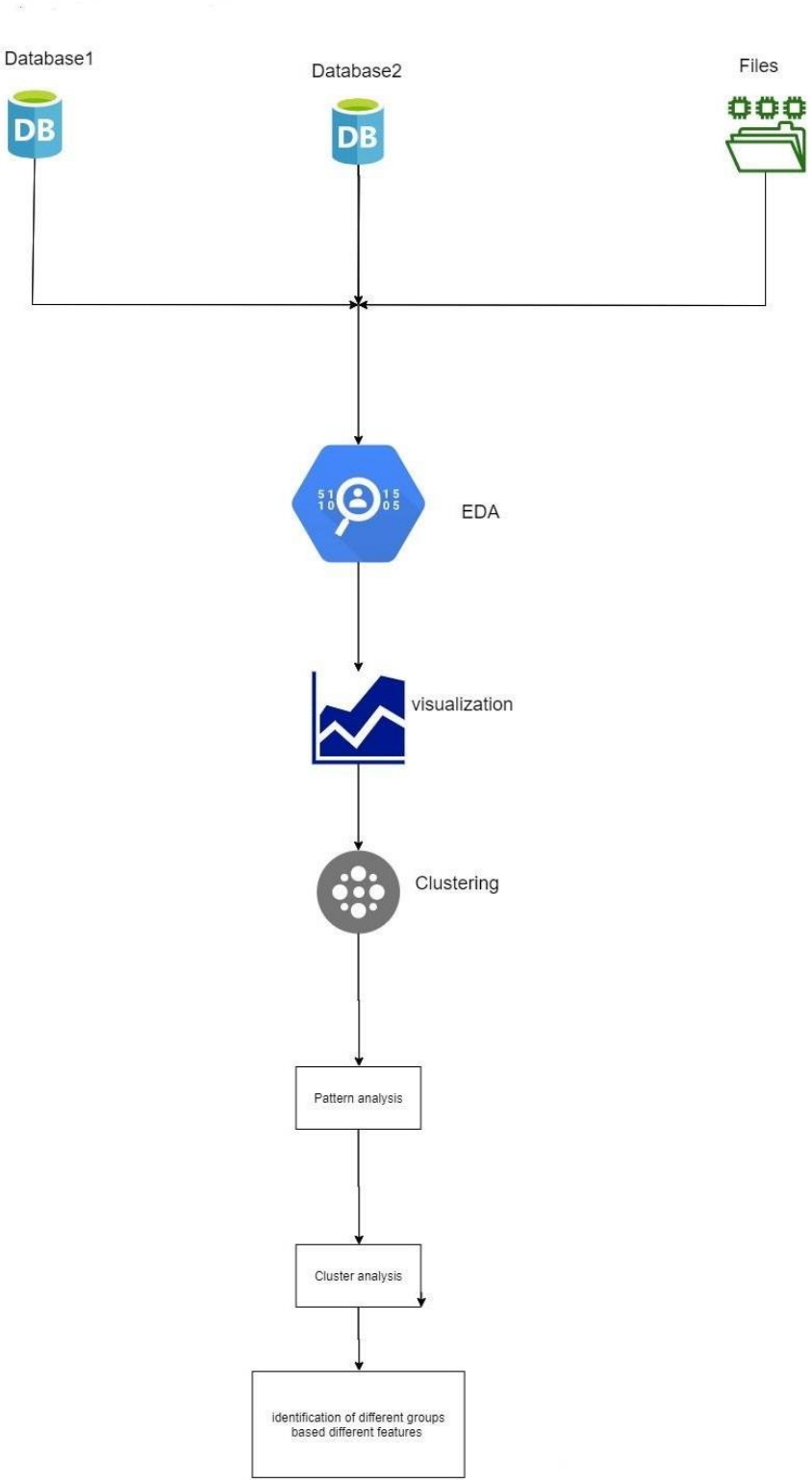
1:Demographics: much of this is publicly-available data that can be gained from your customers during the check-out phase for that first purchase or through other (non-invasive) means: location, age, gender, life stage, marital status.: Facebook and other social media platforms often have much of this information stored on users and can help you build custom audiences for ad outreach.

2: **Decision-making status**: especially useful for business to business companies, knowing a customer's job title can make the difference between barking up the wrong tree and making a sale quickly. But decision-making status also works for business to customer companies. If you're trying to sell children's products, for example, you'll need to target those with the expendable income (the parents).

3: **Customer history with the company**: know how long each customer has used your product or followed your brand, which can tell you the most loyal customers, how they've seen you change, and maybe how they'll react to new changes.

4: **Personal interactions with customers**: Your support team can help you designate your power users, advocates, and even your difficult tickets. These groups are perfect test segments to start your experiments.

Proposed Method with Architecture



The proposed model is a three-phase model

1: **First Phase**: In this phase we will collect the data from then company and apply EDA methods to clean the data or reduce the noise of the data

2: **Second Phase**: In this phase, we will visualize the data and find out various data insights which will be helpful for us in the next phase

3: **Third Phase**: In this phase, we will take the insights from the previous step and apply k-Means clustering and create the clusters, find out the pattern between clusters & group people based on different cluster parameters

Methodology

My goal was to create clusters based on different features like gender ,age, annual income, and spending score from the mall customer dataset. As any of the data science project everything starts with data using the pandas library, I have applied some EDA techniques like checking null or duplicate values, removing unwanted features & checking the min, max boundary of the features

After reducing the noise of the data, visualization was the next most important step in this project visualizing the data gives us more in-depth insights of data or some unidentified patterns in the data. I have used different plots which give good infographics of the data like boxplot, distplot, lmplot, pair plots the most crucial plot which gave us more insights about clusters is scatter plot .All the plots in the project comes under matplotlib & seaborn libraries at the end of visualization we got to know that annual income & spending rate has more clustering nature this leads to next step or phase

I have used the k-Means algorithm for clustering the data points which come in pre-package of sklearn. clusters library to find the optimal no of clusters I have used different statistical methods like the elbow method which is the sum of squared errors (inertia)& silhouette coefficient method under sklearn.meterices

k-Means is unsupervised learning algorithm we don't have any target variable to check its accuracy so domain knowledge and experience plays a important role in it

After finding the optimal k values it is given as input for the k-Means algorithm which produces the minimized clusters ,using scatter plot and representing different clusters with the different colors for easy interpretation of the cluster.

For still more info about the data and more accurate customer clusters, I have chosen one more feature that is age with spending score and annual income. Again I have applied the elbow method and silhouette coefficient method to get the optimal k value and create k clusters with the k-Means algorithm. The clusters were formed and I used ploty to create 3D plots then interpreted the plot to understand the pattern and derive useful results about customer segmentation.

Implementation

Language: Python(Version:3.7.7)

IDE: Jupyter

Notebook(Version:6.1.1)

Dependencies

- pandas(V1.1.0): pandas is a software library written for the Python programming language for data manipulation and analysis
- numpy(V1.19.1): NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays
- matplotlib(V3.2.2): Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy
- seaborn(V0.10.0): Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics
- plotly(V4.8.1): Plotly's Python graphing library makes interactive, publication-quality graphs
- sklearn(V0.23.1): Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting,

k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

- kneed(V0.6.0):kneed in python library used to find the knee point of the function in kneedle algorithm
- warnings(V0.3.0) :warnings is a python library used to handle warnings in Python

import the dependencies

In [1]:
```
# import necessary modules
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly as py
import plotly.graph_objs as go
from kneed import KneeLocator
from sklearn.metrics import silhouette_score
from sklearn.cluster import KMeans
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
sns.set()
```

- %matplotlib inline sets the backend of matplotlib to the 'inline' backend
- sns.set() changes the plot style

EDA

Read the given dataset

In [2]:
```
# read the dataset
df=pd.read_csv(r'C:\Users\prajw\Downloads\customer-segmentation-dataset\Mall_Customers.csv')
df.head()
```

Out[2]:

	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

- pd.read_ csv() reads the csv file and converts that into pandas data frame
- df.head() to display the first 5 rows

EDA

Basic info about the dataframe

```
In [3]:
# basic dataframe info
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
```

- df.info() function is used to get a concise summary of the dataframe

This dataset is composed by the following five features:

- ***CustomerID***: Unique ID assigned to the customer
- ***Gender***: Gender of the customer
- ***Age***: Age of the customer
- ***Annual Income (k$)***: Annual Income of the customer
- ***Spending Score (1-100)***: Score assigned by the mall based on customer behavior and spending nature.
- In this particular dataset we have **200** samples to study.

Description of the data frame

```
In [4]:
# describe of the dataframe
df.describe()
```

Out[4]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

- df.describe() is used to view some basic statistical details like percentile, mean, std etc. of a data frame or a series of numeric values.

Checking for null & duplicate values

```
In [5]: 1 print("Are there any null values in the dataset:{}".format(bool(df.isnull().any().sum())))
        2 print("Are there any duplicates in the dataset :{}".format(bool(df.duplicated().sum())))

Are there any null values in the dataset:False
Are there any duplicates in the dataset :False
```

- df.isnull() is used to check is there are any Null values in the dataframe
- df.duplicated() is used to check whether there are any duplicate values in the data frame

Dropping unwanted features for our data frame

```
In [6]: 1 # dropping unwanted column for our analysis
        2 df = df.drop('CustomerID', 1)
        3 df
```

Out[6]:

	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
...	...	...	...	...
195	Female	35	120	79
196	Female	45	126	28
197	Male	32	126	74
198	Male	32	137	18
199	Male	30	137	83

200 rows × 4 columns

- df.drop() function is used to drop the rows or columns from the data frame
- axis= 1 will drop the columns & axis=0 will drop rows
- we dropped customer id column because that's not useful for clustering

Min and Max values of the features

In [7]:
```
# to check min and max of the spending score
print("The minimum value in age is {}".format(df["Age"].min()))
print("The maximum value in age is {}".format(df["Age"].max()))
print("."*50)
print("The minimum value in annual income is {}".format(df["Annual Income (k$)"].min()))
print("The maximum value in annual income is {}".format(df["Annual Income (k$)"].max()))
print("."*50)
print("The minimum value in spending score is {}".format(df["Spending Score (1-100)"].min()))
print("The maximum value in spending score is {}".format(df["Spending Score (1-100)"].max()))
```

```
The minimum value in age is 18
The maximum value in age is 70
..................................................
The minimum value in annual income is 15
The maximum value in annual income is 137
..................................................
The minimum value in spending score is 1
The maximum value in spending score is 99
```

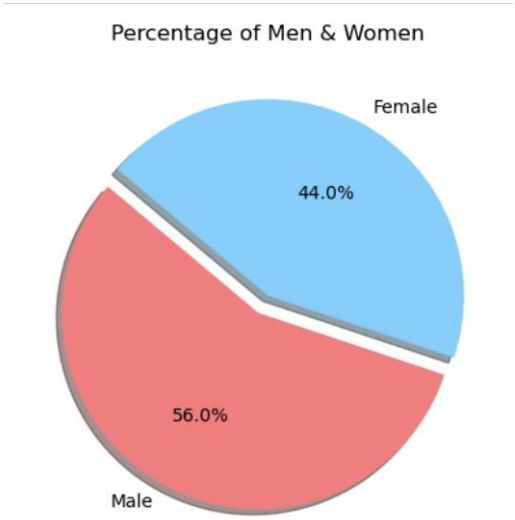
- df.min() function returns the minimum of the values in the given object. If the input is a series, the method will return a scalar which will be the minimum of the values in the series.
- df.max() function returns the maximum of the values in the given object. If the input is a series, the method will return a scalar which will be the maximum of the values in the series.

Visualization

What's the percentage of Male & Female in total customers?

In [8]:
```
sizes=list(df['Gender'].value_counts())
labels=['Female','Male']
fig1,axis1=plt.subplots()
axis1.pie(sizes,labels=labels, autopct='%1.f%%', shadow=True ,colors=['red','orange'],explode=(0,0.1))
axis1.axis('equal')
plt.savefig('malewomenperct.png', dpi=300, bbox_inches='tight')
plt.show()
```

- A pie chart is a circular statistical graphic, which is divided into slices to illustrate numerical proportion. In a pie chart, the arc length of each slice, is proportional to the quantity it represents.
- axis1,pie() function in matplotlib is used to create the pie chart
- plt.savefig() is used to save the plots to the device for different formats



- In the below plot we can see that we have 56% of Female & 44% of Male shopper who shop from the mall

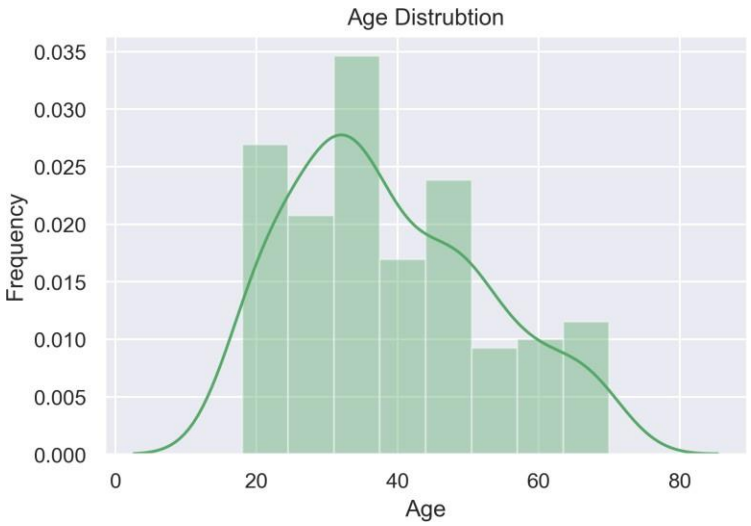
FREQUENCY DISTRIBUTION

Age Distribution

```
In [9]:
ax=sns.distplot(df['Age'],kde = True,label='Count',color="g")
plt.title('Age Distrubtion')
plt.ylabel('Frequency')
plt.savefig('agedist.png', dpi=300, bbox_inches='tight')
plt.show()
```

- A distplot plots a univariate distribution of observations. The distplot() function combines the matplotlib hist function with the seaborn kdeplot() and rugplot() functions**.**

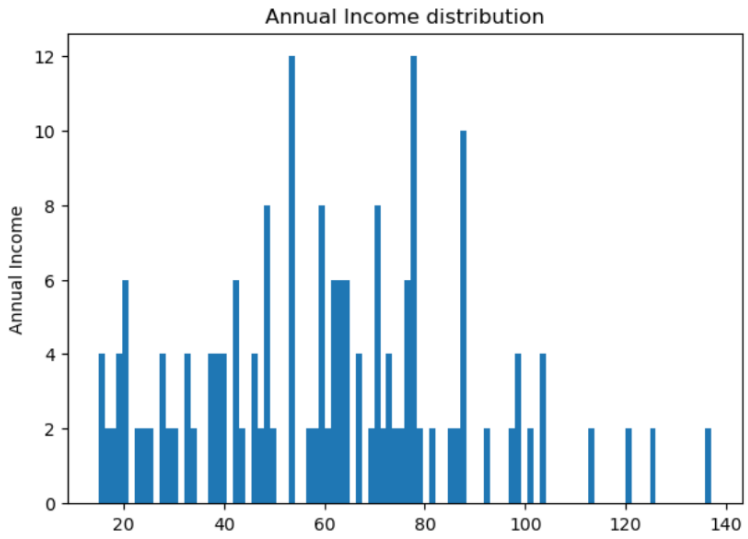
- A distplot plots a univariate distribution of observations. The distplot() function combines the matplotlib hist function with the seaborn kdeplot() and rugplot() functions



- In the above plot we can see that people with age from 30-40 shops a lot in the mall
- The least is age between 50-60

Annual Income Distribution

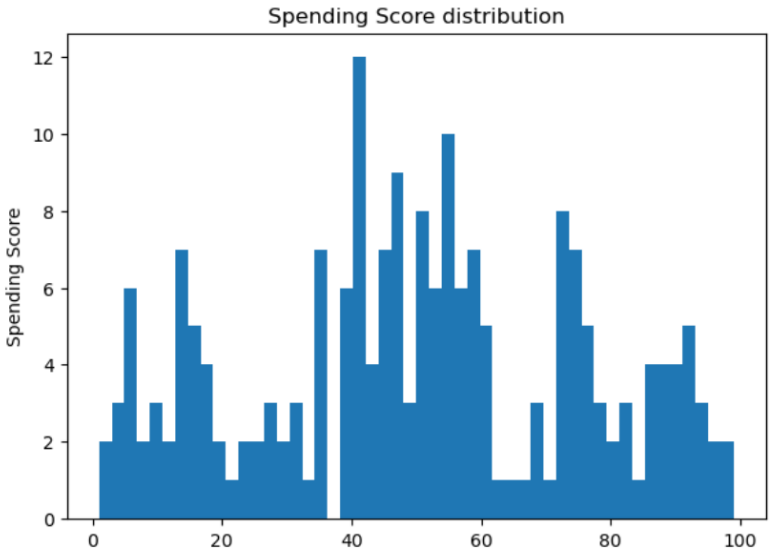
```
In [10]:
1 ax=sns.distplot(df['Annual Income (k$)'],kde = True,color="y")
2 plt.title('Annual Income Distrubtion')
3 plt.ylabel('Frequency')
4 plt.savefig('incomedist.png', dpi=300, bbox_inches='tight')
5 plt.show()
```



- From the annual income distplot we can see that people with annual income 50k-60k spend a lot in the mall
- After 85-95k annual income people tend to come to malls

Spending Score Distribution

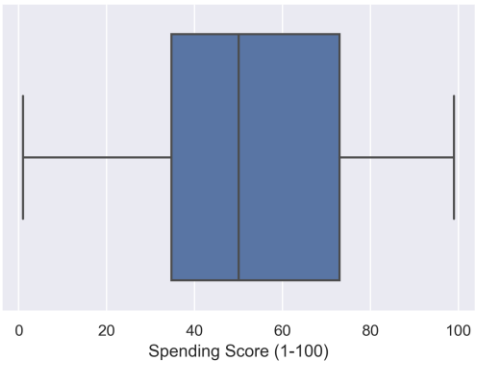
```
In [11]:
ax=sns.distplot(df['Spending Score (1-100)'],kde=True,color='m')
plt.title("Spending Score Frequency Distrubtion")
plt.savefig('scoredist.png', dpi=300, bbox_inches='tight')
plt.show()
```



- People tend to have more spending score in between 40-60

Are there any Outlier in the Spending score?

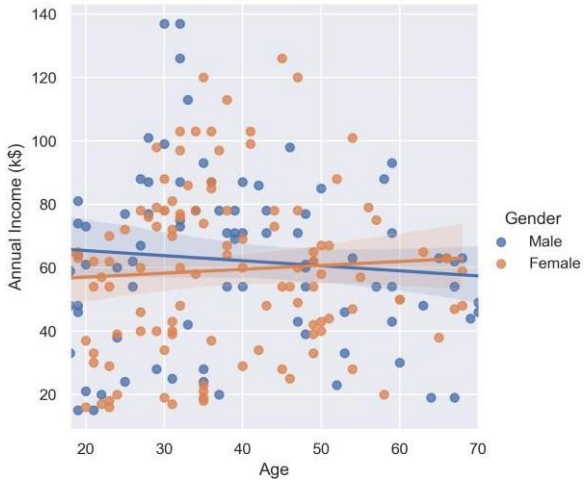
```
In [13]:
sns.boxplot(x=df['Spending Score (1-100)'])
plt.savefig('boxplot.png', dpi=300, bbox_inches='tight')
plt.show()
```



- A box plot is a type of chart often used in explanatory data analysis to visually show the distribution of numerical data and skewness through displaying the data quartiles (or percentiles) and averages.
- Even box plots is used for checking out outliers
- In spending score there is no outliers

Which age group earns more income?

```
In [14]:
1 sns.lmplot(x='Age',y='Annual Income (k$)',data=df,hue="Gender")
2 plt.savefig('reglm.png', dpi=300, bbox_inches='tight')
3 plt.show()
```



- lmplot() is used to plot the data and do regression models fits across a FaceGrid
- We can see that people with age group in 30-50 earn more than the rest of the people

Relationship between age ,annual income and spending score pairwise

```
In [16]:
1 sns.pairplot(df,vars=['Age','Annual Income (k$)','Spending Score (1-100)'], palette="husl", hue = "Gender")
2 plt.savefig('pairplot.png', dpi=300, bbox_inches='tight')
3 plt.show()
```

- To plot multiple pairwise bivariate distributions in a dataset, you can use the pairplot() function. This shows the relationship for (n, 2) combination of variable in a DataFrame as a matrix of plots and the diagonal plots are the univariate plots.

- As we can see we have pairwise pair plot which gives good insight by trying different pairwise combinations with all the numerical features

Will spending score decrease after certain age?

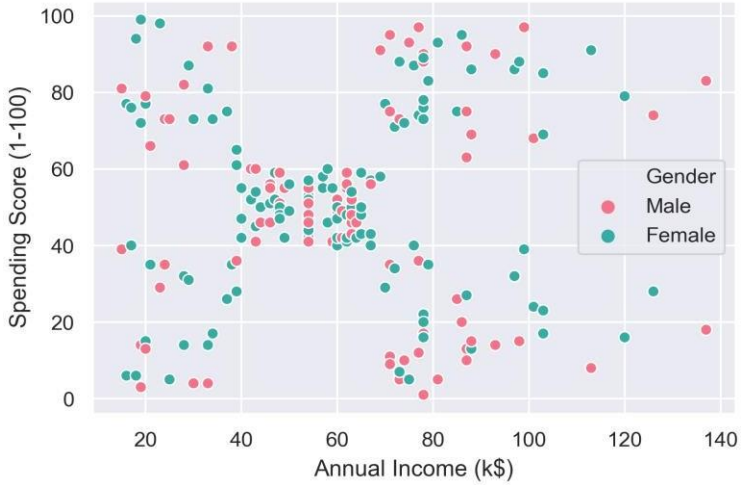
In [17]:
```
sns.scatterplot(x="Age",y="Spending Score (1-100)",data=df, palette="husl", hue = "Gender")
plt.title('Age V/S Spending Score')
plt.savefig('agescplot.png', dpi=300, bbox_inches='tight')
plt.show()
```

- A scatter plot is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data. If the points are coded, one additional variable can be displayed.

- As per the scatter plot we can see that people whose age greater thng score an 40's have spending score less than 60

Are there any clusters in annual income and spending score?

```
In [19]:
1 sns.scatterplot(x='Annual Income (k$)',y='Spending Score (1-100)',data=df, palette="husl", hue = "Gender")
2 plt.savefig('annualpairplot.png', dpi=300, bbox_inches='tight')
3 plt.show()
```



We can checkout manually that there are chances that there will be 5 clusters between annual income & spending score

Clustering

Clustering using 2 features

```
In [17]:
# selecting annual income and spending score
f=df.loc[:,['Annual Income (k$)','Spending Score (1-100)']].values
f
```

```
Out[17]: array([[ 15,  39],
                [ 15,  81],
                [ 16,   6],
                [ 16,  77],
                [ 17,  40],
                [ 17,  76],
                [ 18,   6],
                [ 18,  94],
                [ 19,   3],
                [ 19,  72],
                [ 19,  14],
                [ 19,  99],
                [ 20,  15],
                [ 20,  77],
                [ 20,  13],
                [ 20,  79],
                [ 21,  35],
                [ 21,  66],
                [ 23,  29],
                [ 23,  98],
```

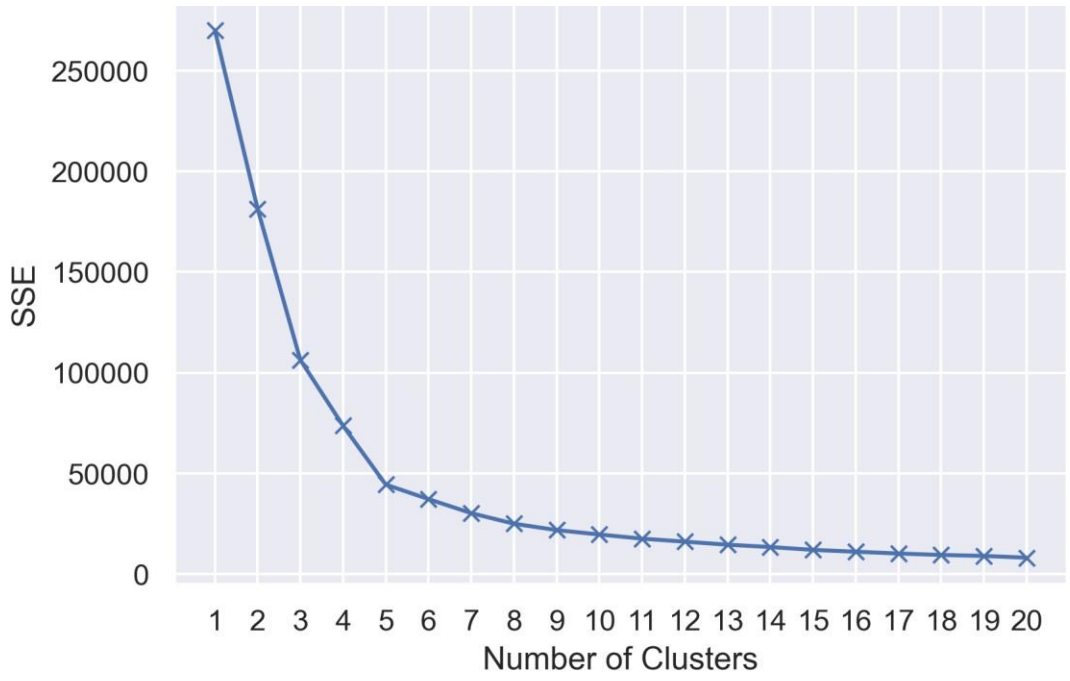
- We gonna select the columns which are important for clustering that is annual income & spending score

Elbow Method

```
In [18]:
sse=[]
l=range(1,21)
for k in l:
    kmeans_model_twof=KMeans(n_clusters=k,random_state=0)
    kmeans_model_twof.fit(f)
    sse.append(kmeans_model_twof.inertia_)
```

- We calculate the sum of squared errors then plot the graph

```
In [19]:
plt.plot(l, sse,'bx-')
plt.xticks(l)
plt.xlabel("Number of Clusters")
plt.ylabel("SSE")
plt.savefig('elbowtwof.png', dpi=300, bbox_inches='tight')
plt.show()
```



- Through the plot we can get to know that 5 is the optimal value for k

```
In [20]: 1 kl = KneeLocator(l, sse,curve="convex", direction="decreasing")
         2 print('The optimal k value for two features is {}'.format(kl.knee))

The optimal k value for two features is 5
```

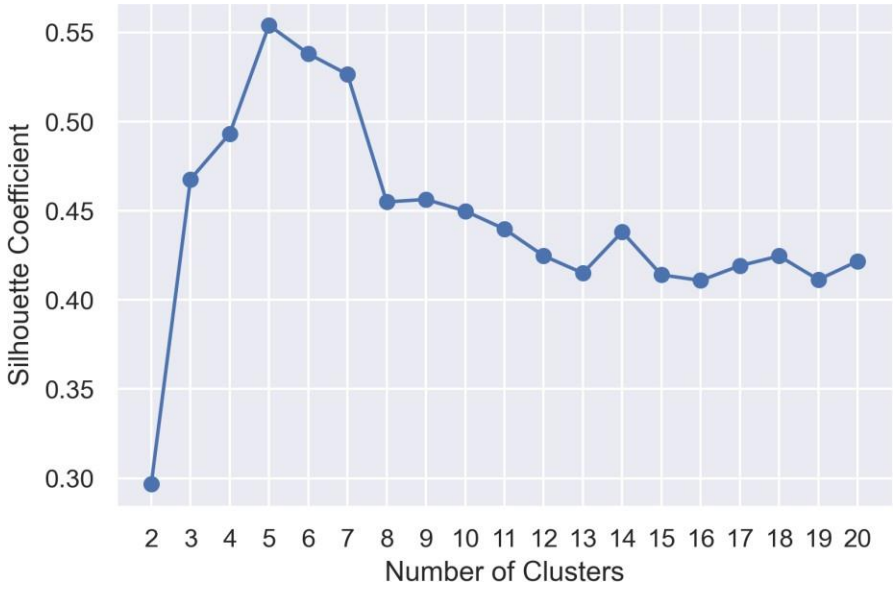
- Sometimes the plot might not give proper information so we will kneed library to check the elbow or knee

Silhouette Coefficient Method

```
In [21]: 1 SC=[]
         2 r= range(2,21)
         3 for k in r:
         4     s_kmeans_model_twof=KMeans(n_clusters=k)
         5     s_kmeans_model_twof.fit(f)
         6     SC.append(silhouette_score(f,s_kmeans_model_twof.labels_))
```

- We have pre-defined function to calculate silhouette score under sklearn.metrics by name silhouette_score

In [22]:
```
plt.plot(r,SC,'bo-')
plt.xticks(r)
plt.xlabel("Number of Clusters")
plt.ylabel("Silhouette Coefficient")
plt.savefig('sichplot.png', dpi=300, bbox_inches='tight')
plt.show()
```



- From the above plot we can observe that 5 has the max rating so we will choose 5 has the optimal no of clusters

K-Means Clustering

In [23]:
```
update_kmeans_model_twof=(KMeans(n_clusters=5,init='k-means++',
                      max_iter=600,
                      random_state=23,algorithm='elkan'))
update_kmeans_model_twof.fit(f)
labels_twof=update_kmeans_model_twof.labels_
centroids_twof=update_kmeans_model_twof.cluster_centers_
y_kmeans=update_kmeans_model_twof.fit_predict(f)
print("Within the clusters the interia(sum of the squares) of the model is:",update_kmeans_model_twof.inertia_)
```

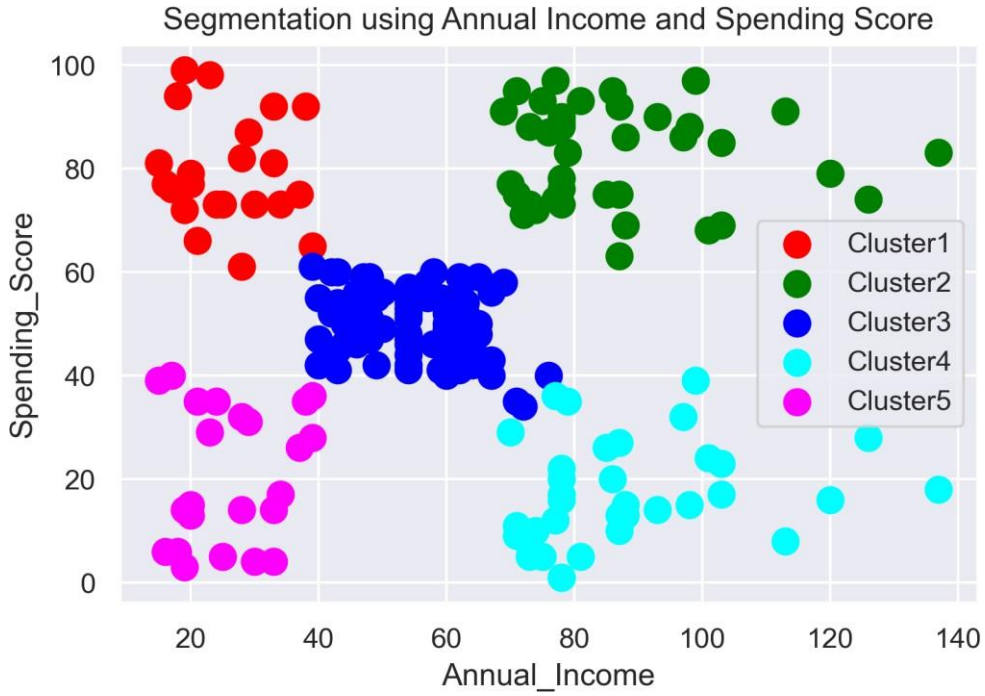
```
Within the clusters the interia(sum of the squares) of the model is: 44448.45544793369
```

- After we choose the optimal k value we will passed that as a parameter to KMeans() which is under sklearn.cluster
- Then we fit our data to the k-Means model
- Store the labels & centroids for plotting purpose

After training the model we will plot it so it would be easy for us to interpret

K-Means scatterplot

```
In [25]:
plt.scatter(f[y_kmeans ==0,0], f[y_kmeans == 0,1], s=100, c='red',label = 'Cluster1')
plt.scatter(f[y_kmeans ==1,0], f[y_kmeans == 1,1], s=100, c='green',label ='Cluster2')
plt.scatter(f[y_kmeans ==2,0], f[y_kmeans == 2,1], s=100, c='blue',label = 'Cluster3')
plt.scatter(f[y_kmeans ==3,0], f[y_kmeans == 3,1], s=100, c='cyan',label = 'Cluster4')
plt.scatter(f[y_kmeans ==4,0], f[y_kmeans == 4,1], s=100, c='magenta',label = 'Cluster5')
plt.xlabel('Annual_Income')
plt.ylabel('Spending_Score')
plt.title('Segmentation using Annual Income and Spending Score')
plt.legend()
plt.savefig('kmeansplot.png', dpi=300, bbox_inches='tight')
plt.show()
```



Cluster 1:The Red cluster groups people with low annual income & high spending score
Cluster 2:The Green cluster groups people with high annual income & high spending score
Cluster 3:The Blue Cluster groups people with medium annual income & spending score
Cluster 4:The Cyan cluster groups people with high annual income & low spending score
Cluster 5:The Magenta Cluster groups people with low annual income & spending score

Clustering for 3 Features

Select the important features

Clustering for 3 features

In [26]:
```
# selecting  necessary features for clustring which are useful and insightful
x=df.loc[:,['Age','Annual Income (k$)','Spending Score (1-100)']]
x
```

Out[26]:

	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
...	...	...	...
195	35	120	79
196	45	126	28
197	32	126	74
198	32	137	18
199	30	137	83

200 rows × 3 columns

- We have selected age,annual income and spending score as our clustering features

Elbow Method

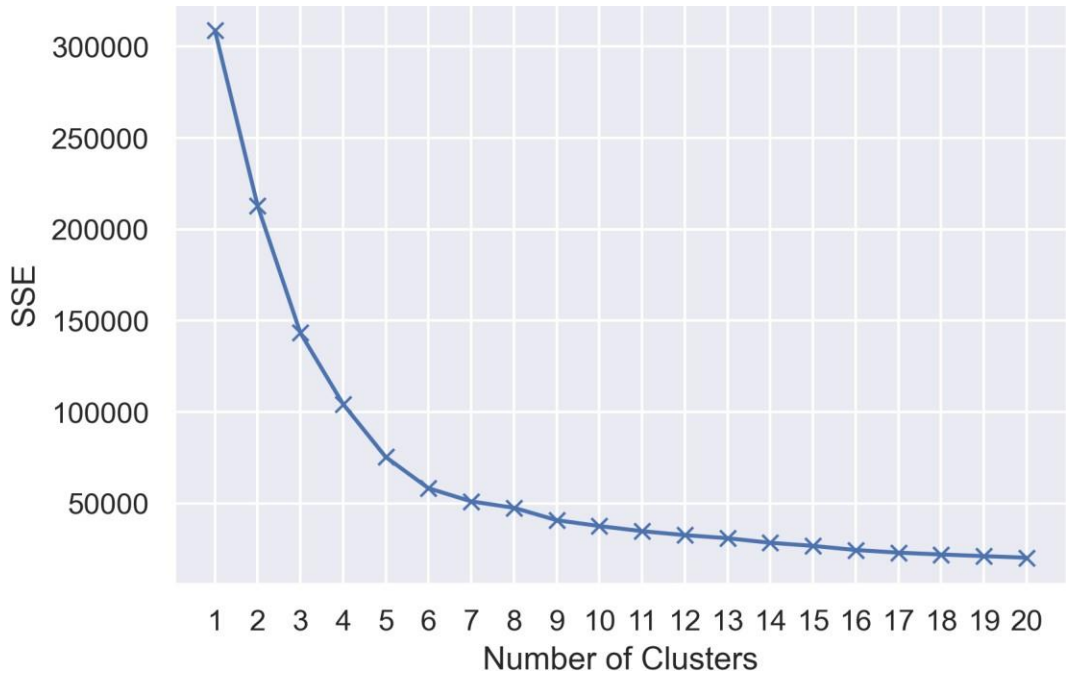
In [27]:
```
SSE=[]
for k in l:
    kmeans_model=KMeans(n_clusters=k, random_state=0)
    kmeans_model.fit(x)
    SSE.append(kmeans_model.inertia_)
```

In [28]:
```
plt.plot(l, SSE,'bx-')
plt.xticks(l)
plt.xlabel("Number of Clusters")
plt.ylabel("SSE")
plt.savefig('elbowtwof.png', dpi=300, bbox_inches='tight')
plt.show()
```

- We have followed the same steps as the clustering for 2 features

In [29]:
```
kl = KneeLocator(l, SSE,curve="convex", direction="decreasing")
print('The optimal k value for two features is {}'.format(kl.knee))
```

```
The optimal k value for two features is 6
```

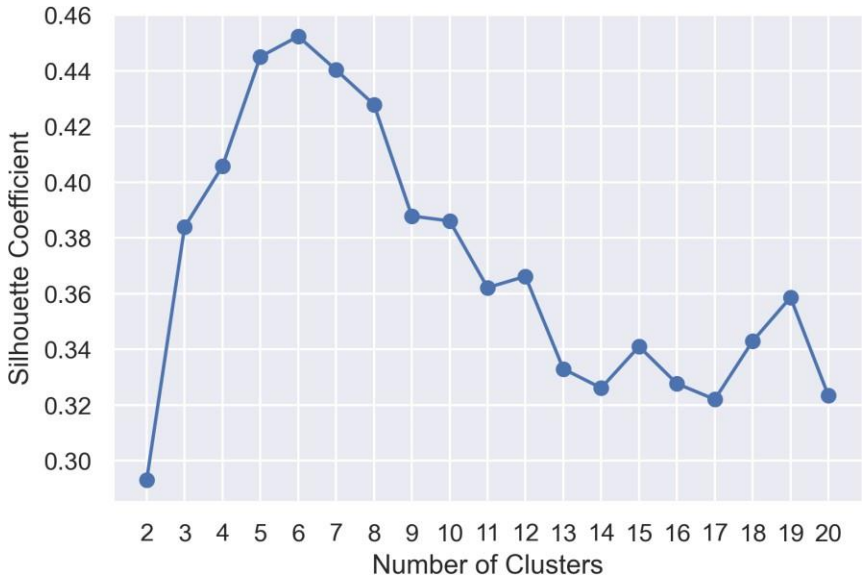


- According to the plot we can see that 6 cluster shows the elbow effect so we will choose k value as 6

Silhouette Coefficient Method

In [30]:
```
sc=[]
r=range(2,21)
for k in r:
    s_kmeans_model=KMeans(n_clusters=k)
    s_kmeans_model.fit(x)
    sc.append(silhouette_score(x,s_kmeans_model.labels_))
```

In [31]:
```
plt.plot(r,sc,'bo-')
plt.xticks(r)
plt.xlabel("Number of Clusters")
plt.ylabel("Silhouette Coefficient")
plt.savefig('sichplotthreef.png', dpi=300, bbox_inches='tight')
plt.show()
```



- After elbow method & silhouette method we will choose 6 has optimal k value

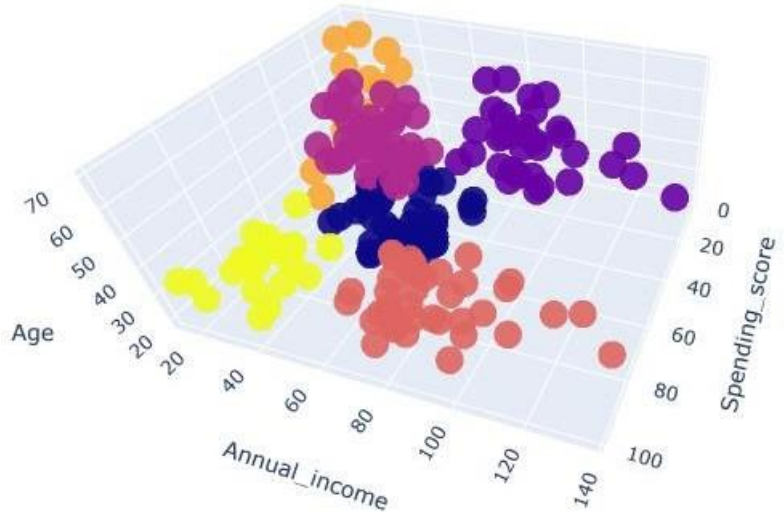
K-Means Clustering

In [32]:
```
update_kmeans_model = KMeans(n_clusters=6,init='k-means++',
                      max_iter=600,
                      random_state=23)
update_kmeans_model.fit(x)
labels=update_kmeans_model.labels_
centroids=update_kmeans_model.cluster_centers_
print("Within the clusters the interia(sum of the squares) of the model is:",update_kmeans_model.inertia_)
```

```
Within the clusters the interia(sum of the squares) of the model is: 58300.443321590676
```

K Means Plotting

In [33]:
```
trace=go.Scatter3d(
x=x['Spending Score (1-100)'],
y=x['Annual Income (k$)'],
z=x['Age'],
mode='markers',
     marker=dict(
        color = labels,
        size= 10,
        line=dict(
            color= labels,
        ),
        opacity = 0.9
     )
)
layout = go.Layout(
    title= 'Clusters',
    scene = dict(
            xaxis = dict(title  = 'Spending_score'),
            yaxis = dict(title  = 'Annual_income'),
            zaxis = dict(title  = 'Age')
        )
)
fig = go.Figure(data=trace, layout=layout)
py.offline.iplot(fig)
```



- **Yellow Cluster**: The yellow cluster groups young people with moderate to low annual income who actually spend a lot.
- **Dark Blue Cluster:** The dark blue clusters groups who are adults pretty medium annual income and spending score
- **Orange Red Cluster**: The orange red cluster groups reasonably young people with pretty decent salaries who spend a lot.
- **Orange Cluster**: The orange cluster groups whose salary is pretty low and don't spend much money in stores, they are people of all ages.
- **Blue Violet Cluster**: The blue Violet cluster groups people who actually have pretty good salaries and barely spend money, their age usually lays between thirty and sixty years
- **Purple Cluster**: The purple cluster groups people who are old with pretty medium annual income and spending score

Conclusion

After developing a solution for the given here are some the conclusions

- Customer segmentation is a good way to understand the behavior of different customers and plan a good marketing strategy accordingly.
- K-Means Clustering is a powerful technique in order to achieve a decent customer segmentation.
- According to the plots there is no big difference between spending score of men & women
- According to cluster plots we tell that young people spend more money on malls maybe due to mall being monopoly for trending clothes ,movies ,place to hangout or food chains
- Promoting discounts on some shops can be something of interest to those who don't actually spend a lot and they may end up spending more!