Objective of this dataset:

The main objective of this dataset is to group customers based on their age, purchasing behaviour using clustering technique.

Following has been done for the dataset

1) Loaded the data from source
2) Performed Exploratory data analysis on this data to find insights through visual plots and also summarised the data using descriptive statistics
3) Applied K Means Algorithm to group the data points
4) Used Elbow Curve for finding optimal clusters
5) Labelled the customers based on their attributes

STEP 1: Loading of all necessary libraries for doing the analysis

In [1]:
```
import numpy as np
import pandas as pd
import pylab as pl
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sb
import missingno as msno
import plotly.express as px
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans
import warnings
warnings.simplefilter('ignore')
plt.style.use("dark_background")
```

STEP 2: Loading of Data from source file

In [2]:
```
df = pd.read_csv('C:/Users/Karthik/Desktop/Ultra Insights/Mall_Customers.csv')
```

STEP 3: Exploratory Data Analysis on data with summary statistics and plots

In [3]:
```
df.head(10)
```

Out[3]:

	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
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72

In [4]:
```
df.shape
```

Out[4]:
```
(200, 5)
```

In [5]:
```
# To check whether the any of the features has null values
df.isnull().sum()
```

Out[5]:
```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

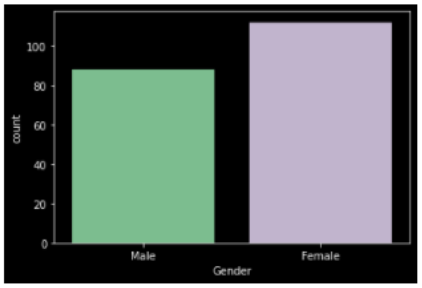
In [6]:
```
df.describe()
```

Out[6]:

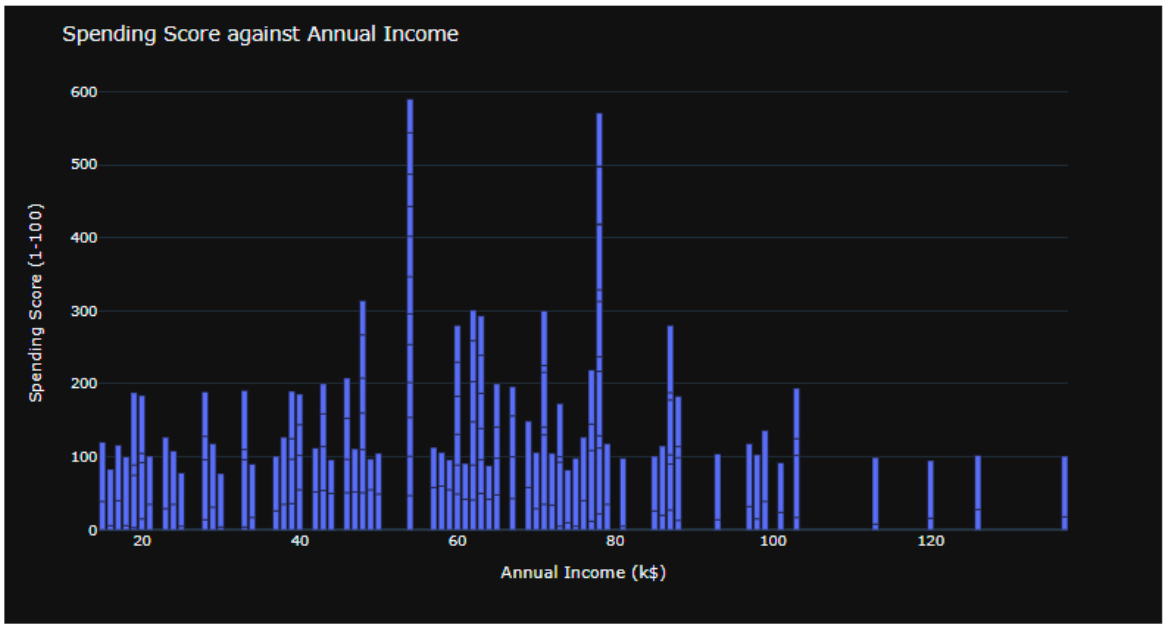
	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000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

In [7]:
```
sb.countplot(df['Gender'], saturation=.66, palette='Accent')
```

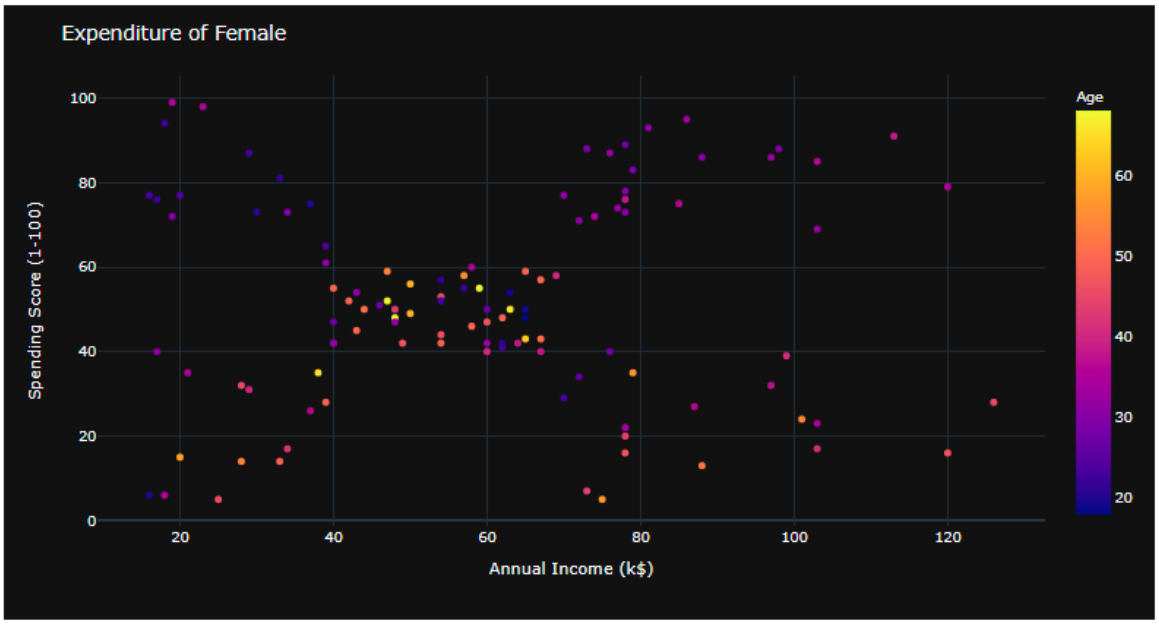
Out[7]:
```
<AxesSubplot:xlabel='Gender', ylabel='count'>
```



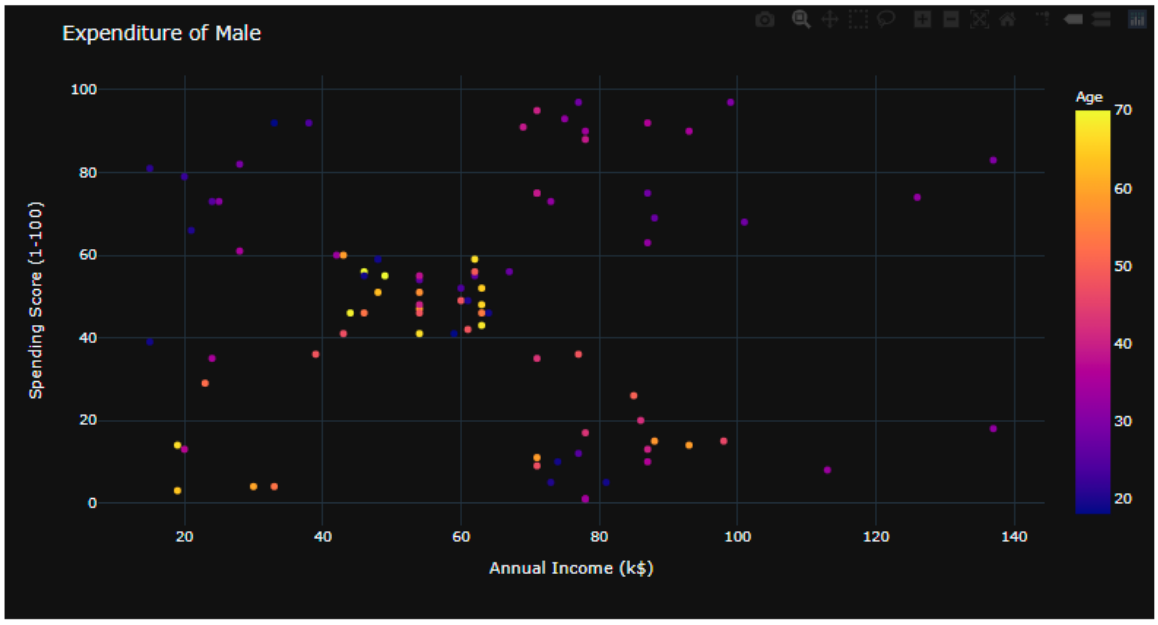
In [8]:
```
# Income wise distribution plot
income = px.bar(df, x="Annual Income (k$)", y="Spending Score (1-100)", template="plotly_dark")
income.update_layout(title_text="Spending Score against Annual Income")
income.show()
```



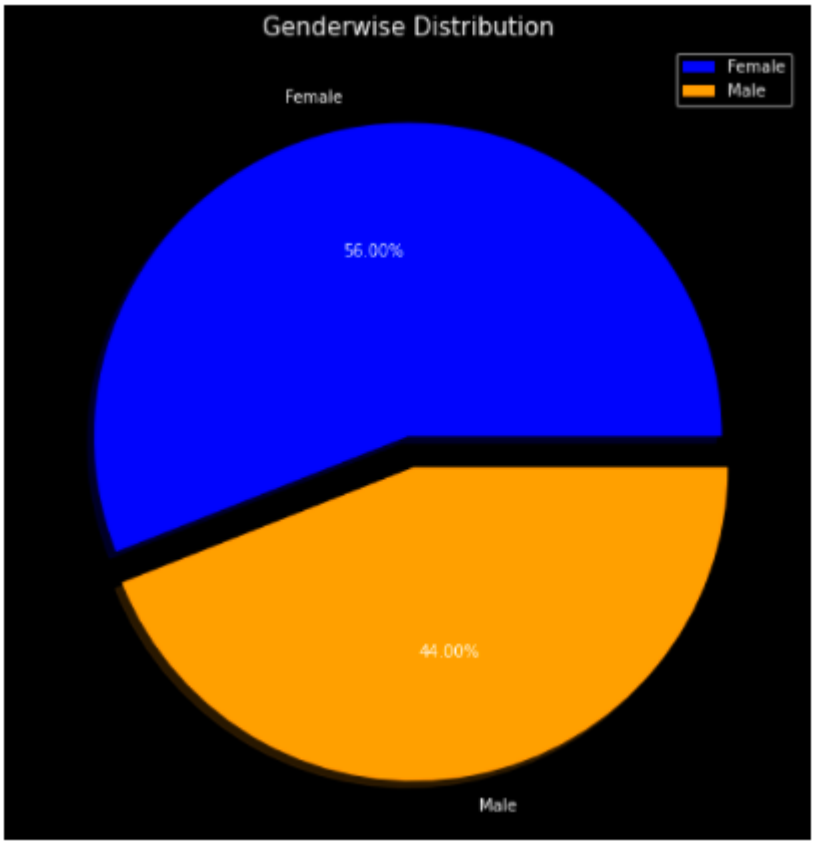
In [9]:
```
# Spending Score distribution for female
female = df[df['Gender'].str.contains("Female")]
female = px.scatter(female, x="Annual Income (k$)", y="Spending Score (1-100)", template="plotly_dark", color="Age")
female.update_layout(title_text="Expenditure of Female")
female.show()
```



In [10]:
```
# Spending Score Distribution for Male
male= df[df['Gender'].str.contains("Male")]
male = px.scatter(male, x="Annual Income (k$)", y="Spending Score (1-100)", template="plotly_dark", color="Age",)
male.update_layout(title_text="Expenditure of Male")
male.show()
```

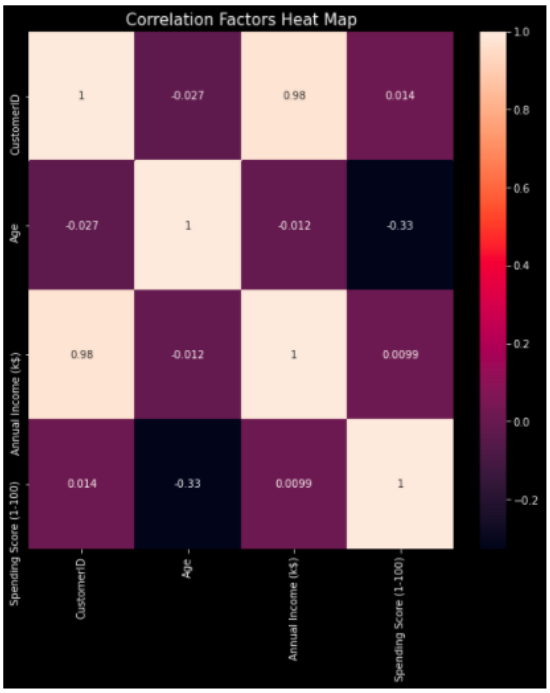


In [11]:
```
## Genderwise Distribution Plot
labels = ['Female', 'Male']
size = df['Gender'].value_counts()
colors = ['blue', 'orange']
explode = [0, 0.1]

plt.rcParams['figure.figsize'] = (9, 9)
plt.pie(size, colors = colors, explode = explode, labels = labels, shadow = True, autopct = '%.2f%%')
plt.title('Genderwise Distribution', fontsize = 15)
plt.axis('off')
plt.legend()
plt.show()
```



In [12]:
```
## Correlation coeffecients heatmap
sns.heatmap(df.corr(), annot=True).set_title('Correlation Factors Heat Map', size='15')
```

Out[12]: Text(0.5, 1.0, 'Correlation Factors Heat Map')



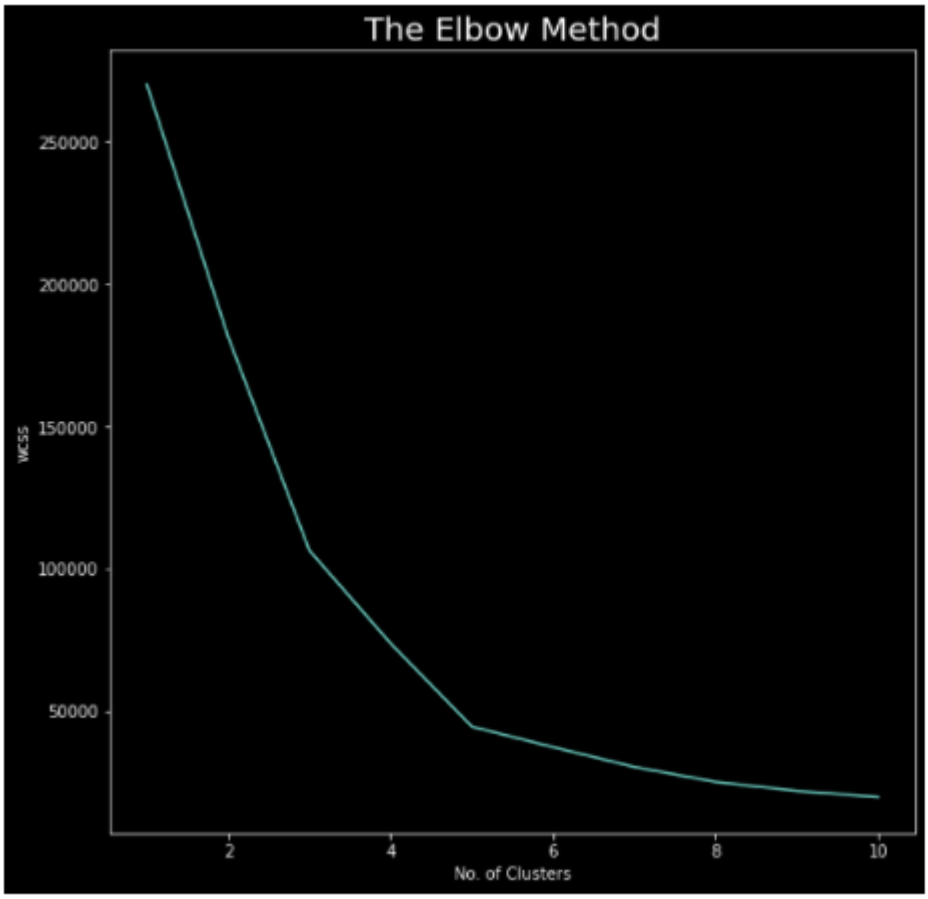
The Above Graph for Showing the correlation between the different attributes of the Mall Customer Segementation Dataset, This Heat map reflects the most correlated features with Orange Color and least correlated features with yellow color.

STEP 4: Building of model with K means clustering

In [15]:
```
x = df.iloc[:, [3, 4]].values

# let's check the shape of x
print(x.shape)
```

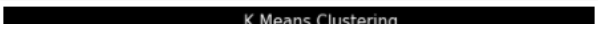
```
(200, 2)
```

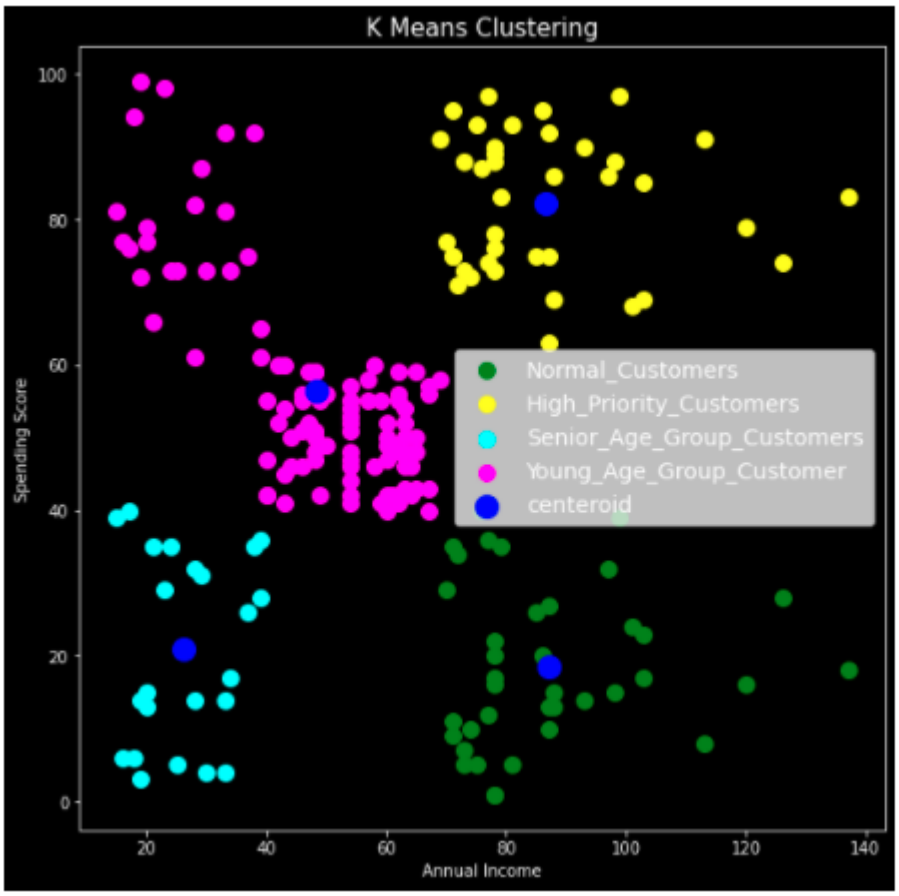
In [23]:
```
from sklearn.cluster import KMeans

wcss = []
for i in range(1, 11):
    km = KMeans(n_clusters = i, init = 'k-means++', max_iter = 100000, n_init = 12, random_state = 0)
    km.fit(x)
    wcss.append(km.inertia_)

plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method', fontsize = 20)
plt.xlabel('No. of Clusters')
plt.ylabel('wcss')
plt.show()
```



From the above Elbow plot we infer that the dataset needs to grouped under 4 clusters

In [24]:
```
km = KMeans(n_clusters = 4, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
y_means = km.fit_predict(x)

plt.scatter(x[y_means == 0, 0], x[y_means == 0, 1], s = 100, c = 'green', label = 'Normal_Customers')
plt.scatter(x[y_means == 1, 0], x[y_means == 1, 1], s = 100, c = 'yellow', label = 'High_Priority_Customers')
plt.scatter(x[y_means == 2, 0], x[y_means == 2, 1], s = 100, c = 'cyan', label = 'Senior_Age_Group_Customers')
plt.scatter(x[y_means == 3, 0], x[y_means == 3, 1], s = 100, c = 'magenta', label = 'Young_Age_Group_Customer')
plt.scatter(km.cluster_centers_[:,0], km.cluster_centers_[:, 1], s = 200, c = 'blue' , label = 'centeroid')

plt.style.use('fivethirtyeight')
plt.title('K Means Clustering', fontsize = 15)
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.legend()
plt.show()
```




K Means Clustering
Spending Score
Annual Income
Normal_Customers
High_Priority_Customers
Senior_Age_Group_Customers
Young_Age_Group_Customer
centeroid
0
20
40
60
80
100
20
40
60
80
100
120
140