EXPOSYS DATA LABS

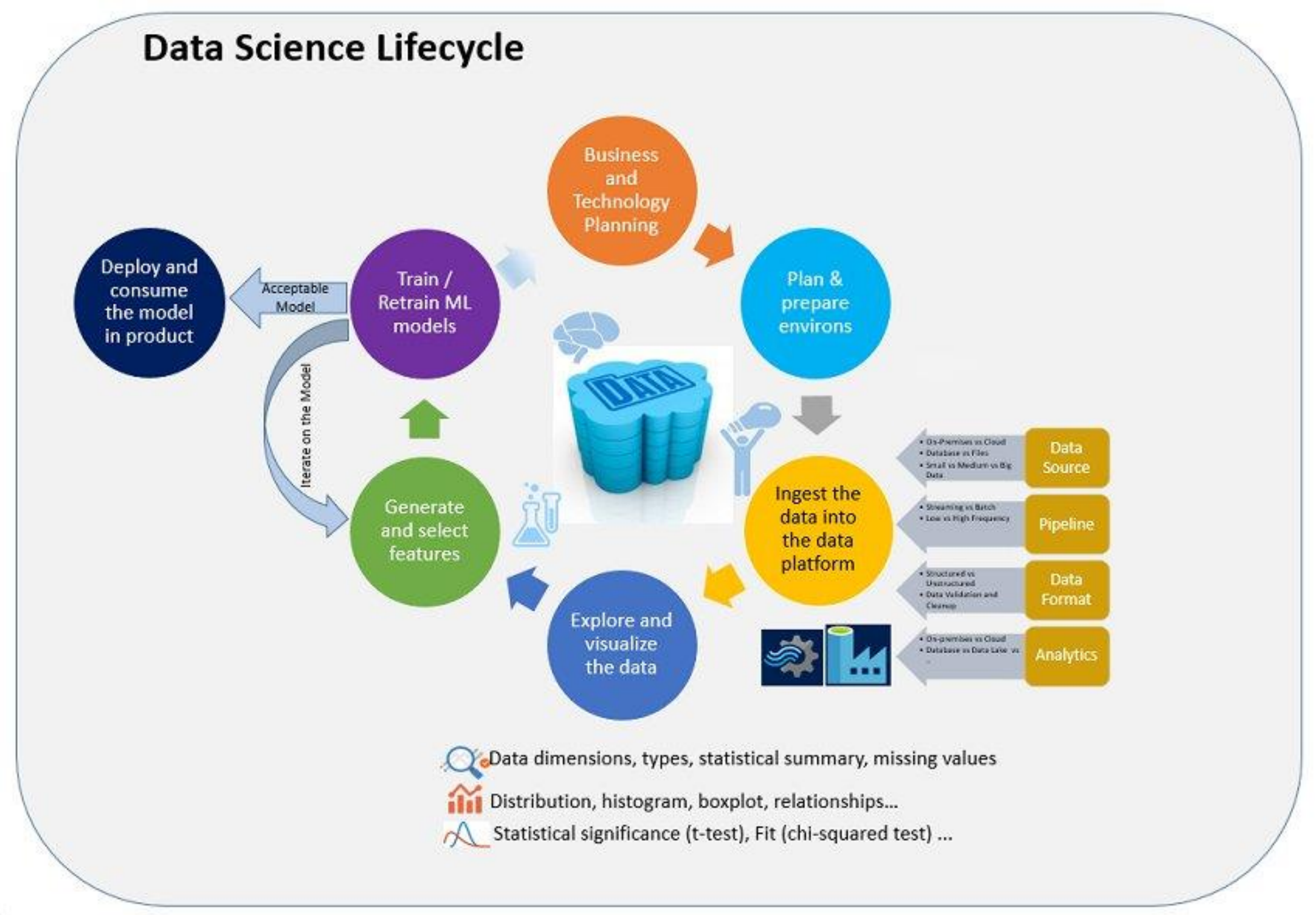
CUSTOMER SEGMENTATION PROJECT

Presenter's name : Bahaduri Prachiti Jagdish

Position : Data Science Intern

DATA SCIENCE

DATA SCIENCE IS AN INTER – DISCIPLINARY THAT USES SCIENTIFIC METHODS, PROCESSES, ALGORITHMS TO EXTRACT KNOWLEDGE AND INSIGHTS FROM MANY STRUCTURAL AND UNSTRUCTURED DATA.



Data Science Lifecycle

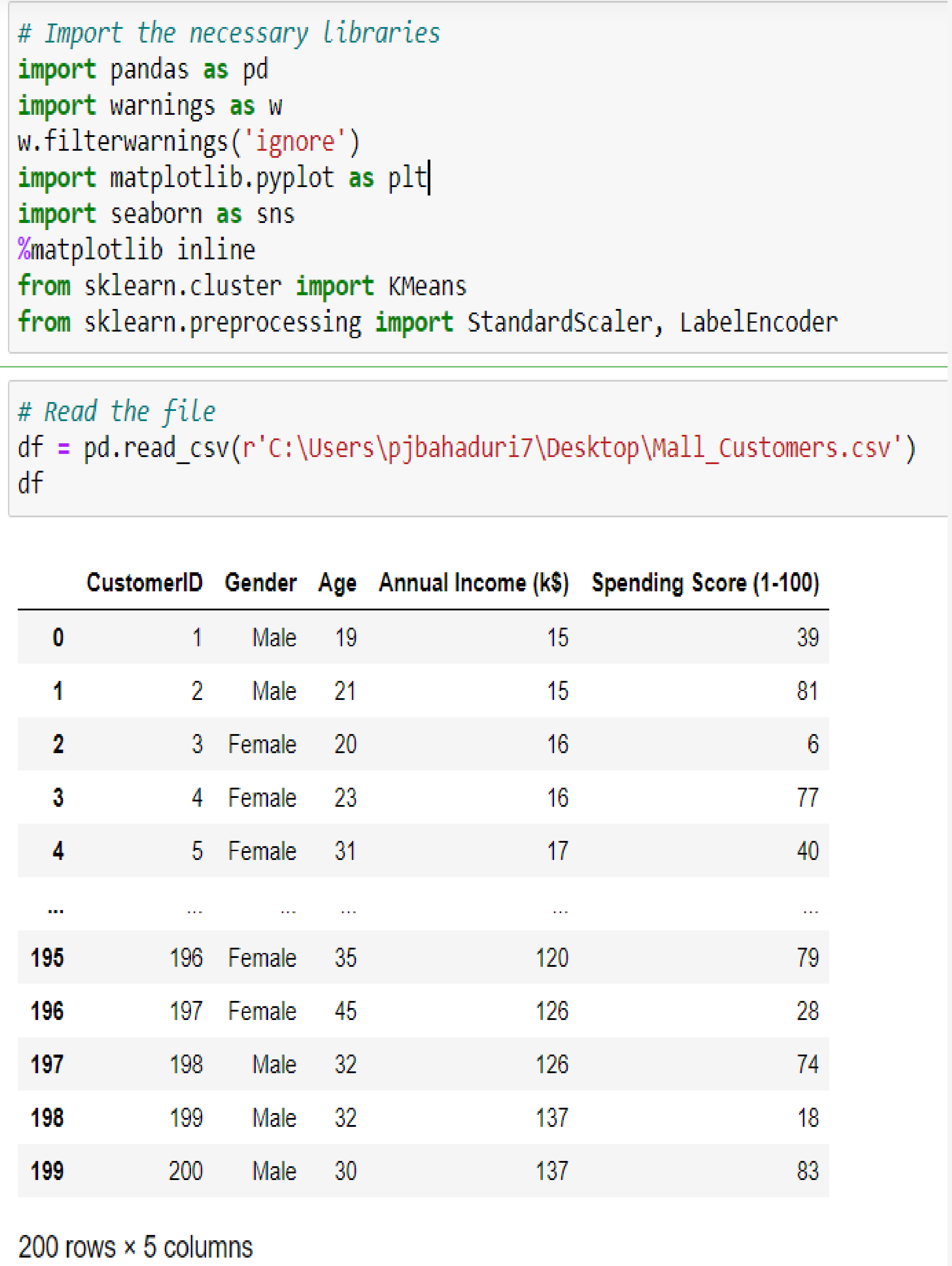
Business and Technology Planning
Plan & prepare environs
Ingest the data into the data platform
Explore and visualize the data
Generate and select features
Train / Retrain ML models
Acceptable Model
Deploy and consume the model in product
Iterate on the Model
DATA
• On-Premises vs Cloud
• Database vs Files
• Small vs Medium vs Big Data
Data Source
• Streaming vs Batch
• Low vs High Frequency
Pipeline
• Structured vs Unstructured
• Data Validation and Cleanup
Data Format
• On-premises vs Cloud
• Database vs Data Lake vs ...
Analytics
Data dimensions, types, statistical summary, missing values
Distribution, histogram, boxplot, relationships...
Statistical significance (t-test), Fit (chi-squared test) ...

CUSTOMER SEGMENTATION

❑ Customer Segmentation enables a company to customize its relationship with its customers.

❑ The basic characteristics and needs are generalized into groups using various strategies viz,

- Targeted marketing activities to specific groups
- Launch of features aligning with the customer demand
- Development of the product roadmap



```python
# Import the necessary libraries
import pandas as pd
import warnings as w
w.filterwarnings('ignore')
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler, LabelEncoder
```

```python
# Read the file
df = pd.read_csv(r'C:\Users\pjbahaduri7\Desktop\Mall_Customers.csv')
df
```

	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

ABOUT THE DATASET AND PROJECT OBJECTIVE

❑ The code is written in Python of V3.7.6

❑ The required modules are imported with the respective versions

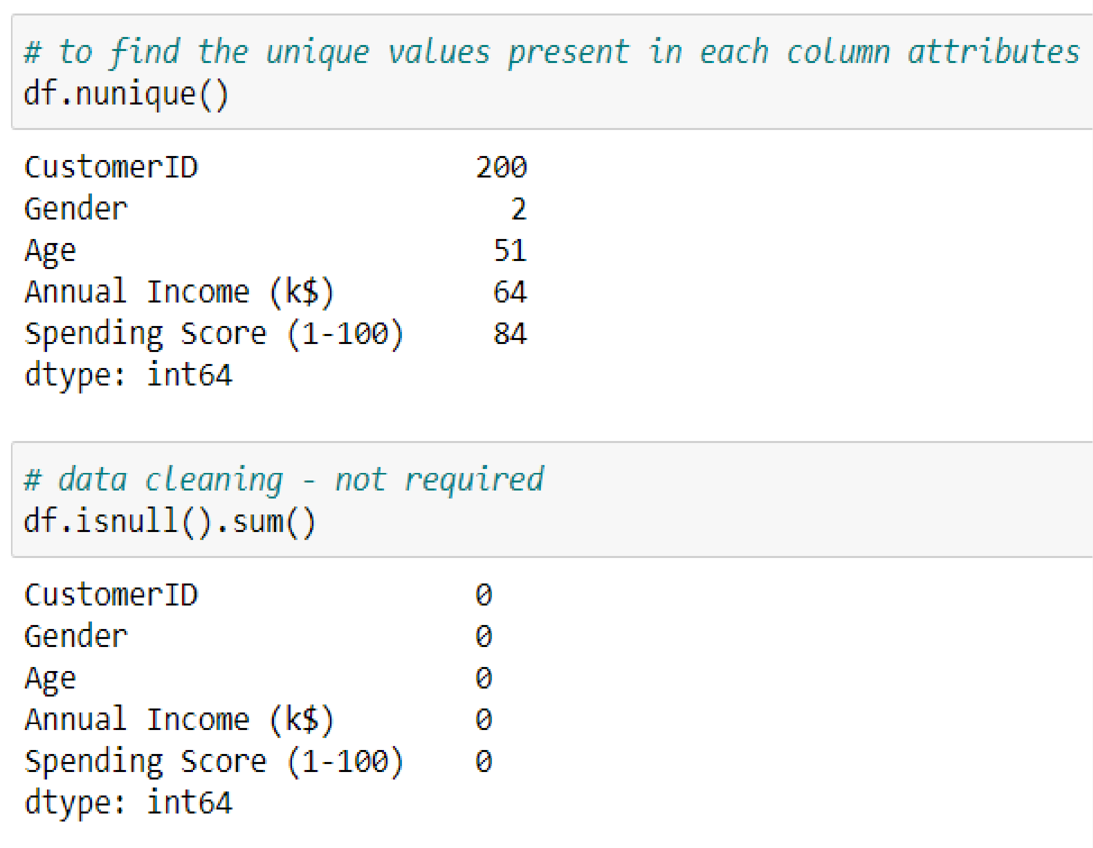
- Pandas – V1.0.1
- Matplotlib – V3.1.3
- Seaborn – V0.10.0
- Scikit – learn – V0.22.1

❑ The attributes of the dataset are the basic characteristics and needs of the customer.

❑ The main aim of the project is to use Unsupervised Learning via K means technique to identify segments of customers using clusters.

DATA CLEANING AND OTHER OPERATIONS

- ❑ The dataset is checked for the presence of null values.
- ❑ Also the number of unique values are obtained for each attributes.
- ❑ The gender is label – encoded for easier analysis by calling the LabelEncoder function.

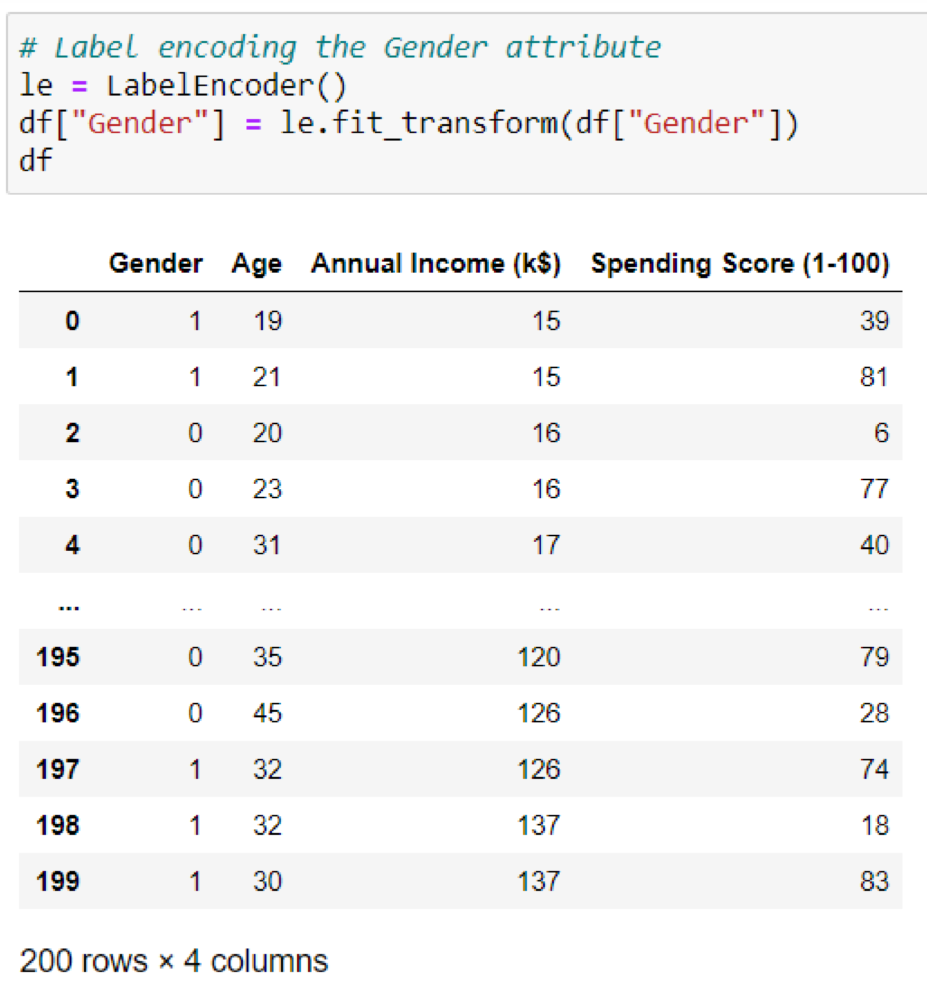


```
# to find the unique values present in each column attributes
df.nunique()
```

```
CustomerID                200
Gender                      2
Age                        51
Annual Income (k$)         64
Spending Score (1-100)     84
dtype: int64
```

```
# data cleaning - not required
df.isnull().sum()
```

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

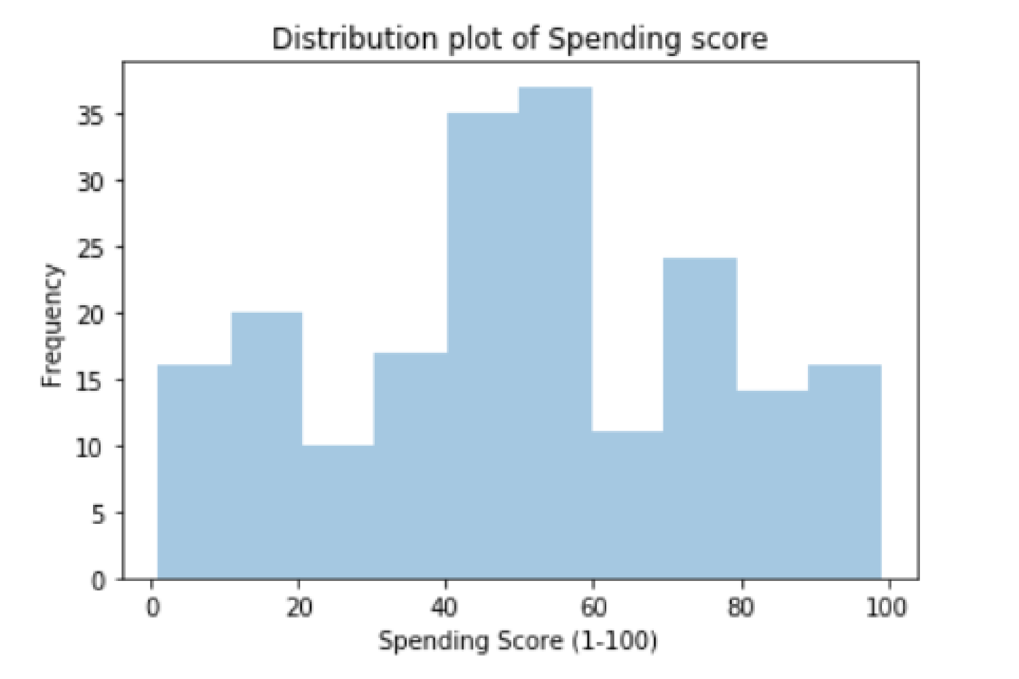


```
# Label encoding the Gender attribute
le = LabelEncoder()
df["Gender"] = le.fit_transform(df["Gender"])
df
```

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

200 rows × 4 columns

RANGE OF SPENDING SCORE



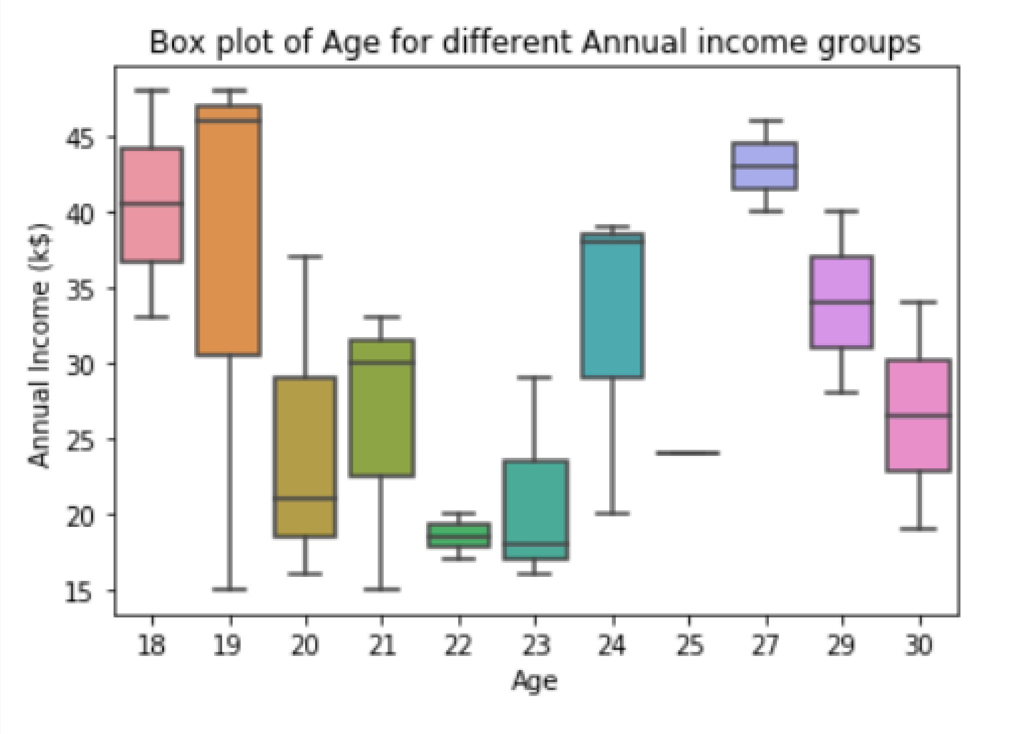
❑ Univariate distribution of Spending Score.

❑ The bin size between each value in the X variable is 20.

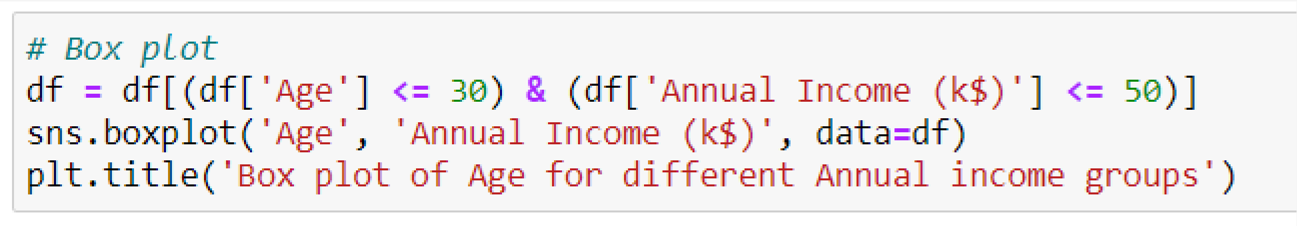
❑ There is a rise in the count when score is in the range of 40 and 60.

❑ The least count of 10 is in the score range of 20 to 30.

BOX PLOT ESTIMATIONS

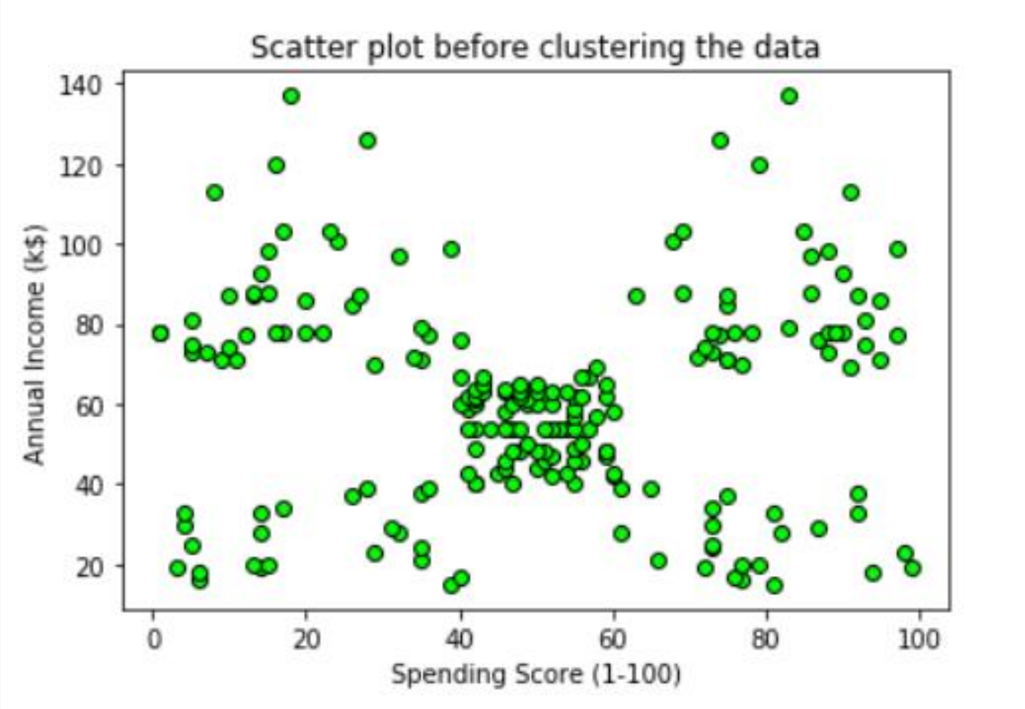


- A standardized way of displaying a dataset.
- Maximum annual income is 48$ - age 18 and 19
  Minimum income is 15$ for the ages 19 and 21.
- Equal number of customers who have surpassed first and third quartile regions are of the age 18, 27, 29, 30.
- No sign of outliers.
- 25 percentile surpassed customers with age 19, 21 and 24 with income 31$, 22$ and 28$ respectively.
- The interquartile region is more for age 20 with income 28$.



```
# Box plot
df = df[(df['Age'] <= 30) & (df['Annual Income (k$)'] <= 50)]
sns.boxplot('Age', 'Annual Income (k$)', data=df)
plt.title('Box plot of Age for different Annual income groups')
```

BEFORE CLUSTERING



- ❑ A type of plot using cartesian coordinates to display values.
- ❑ Scatter plot is used to map out a better relationship between the two numeric variables.
- ❑ Suggests various kinds of correlations between variables with certain confidence interval.

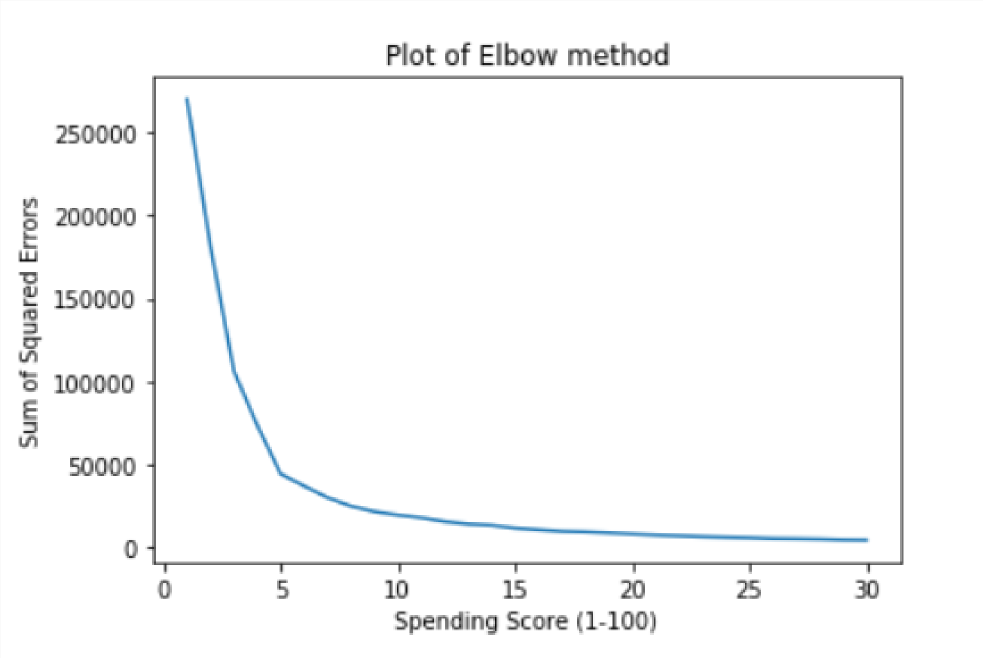
THE ELBOW METHOD

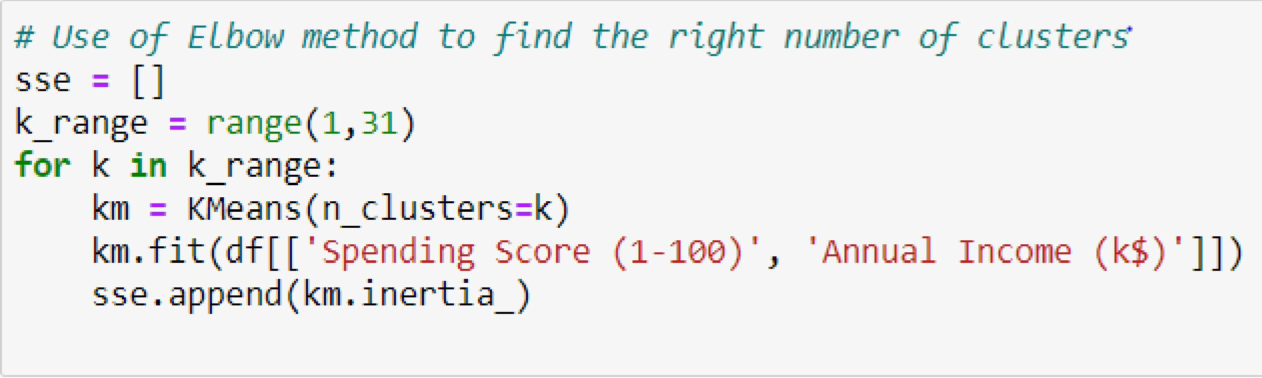
❑ A heuristic used in determining the number of clusters in the dataset.

❑ Pick the elbow of the curve as a number of clusters.

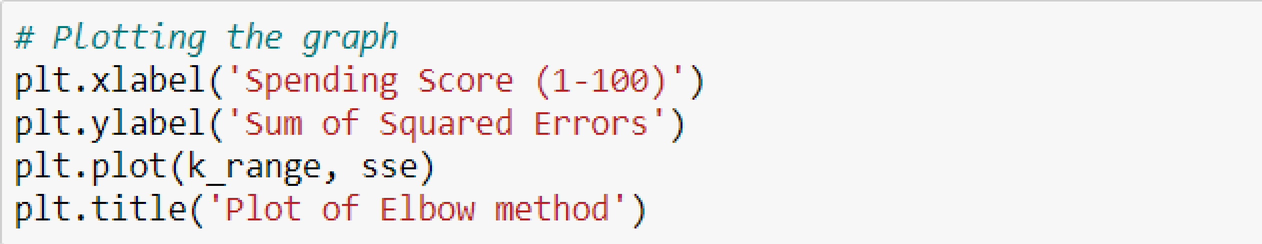
❑ Sum of squared errors has to be minimized for the given set of centroids.

❑ It is evident that the k value is 5.



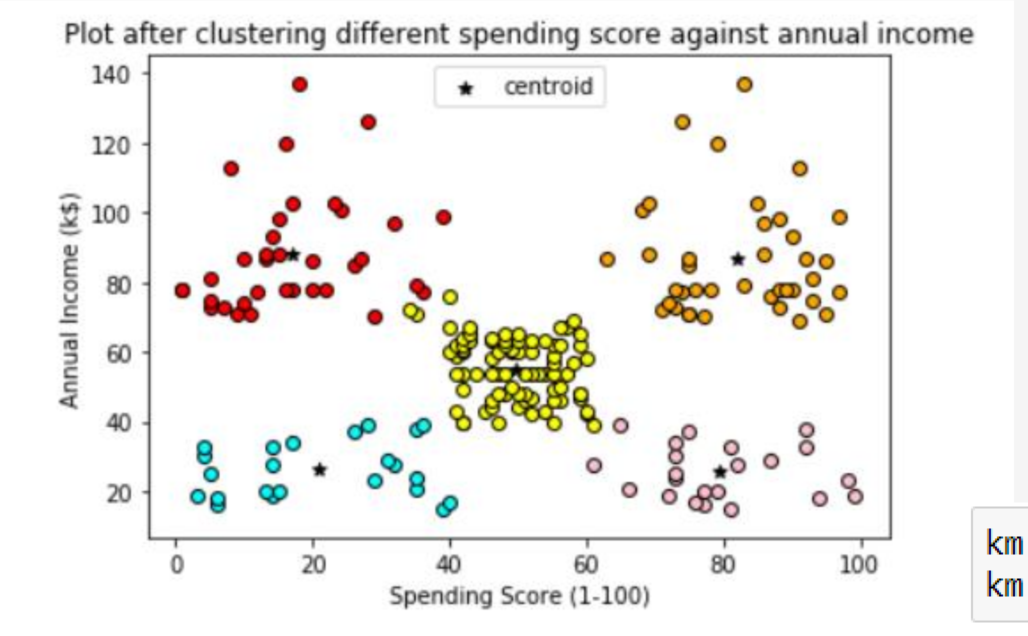


```
# Use of Elbow method to find the right number of clusters
sse = []
k_range = range(1,31)
for k in k_range:
    km = KMeans(n_clusters=k)
    km.fit(df[['Spending Score (1-100)', 'Annual Income (k$)']])
    sse.append(km.inertia_)
```



```
# Plotting the graph
plt.xlabel('Spending Score (1-100)')
plt.ylabel('Sum of Squared Errors')
plt.plot(k_range, sse)
plt.title('Plot of Elbow method')
```

AFTER CLUSTERING



❑ Cyan coloured cluster 🟦

Spending Score – 0 to 40 within 20$ to 40$ of income.

❑ Red coloured custer 🟥

Spending score – 0 to 40 within income 80$ to 140$

❑ Yellow coloured cluster 🟨

Spending score – 40 to 60 within income 40$ to 70$

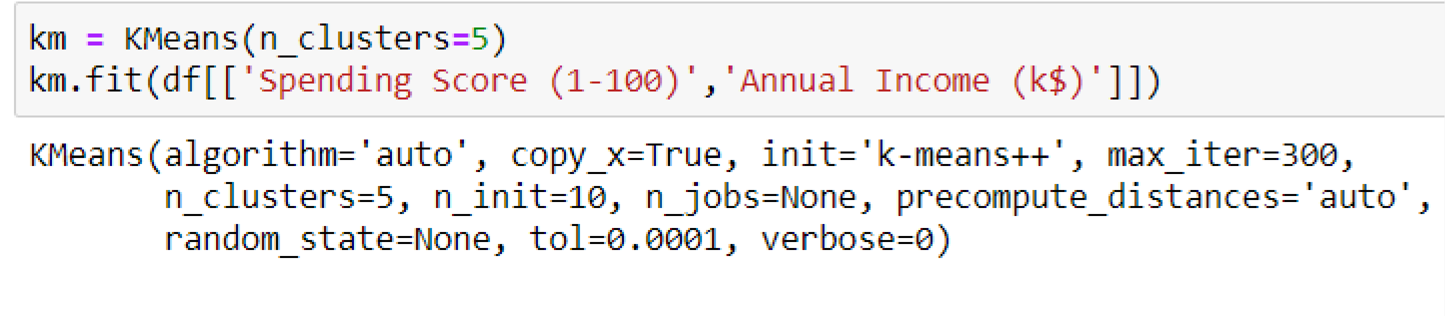
❑ Orange coloured cluster 🟧

Spending score – 60 to 100 within income 20$ to 40$

❑ Pink coloured cluster 🟪

Spending score – 60 to 100 within income 78$ to 140$

❑ Finally the K means method is called and the result is acquired.



```
km = KMeans(n_clusters=5)
km.fit(df[['Spending Score (1-100)','Annual Income (k$)']])
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=5, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

THANK YOU