**Streaming Data Analytics**

**Group 059:**

Avi Krishna Srivastava – 2020fc04492

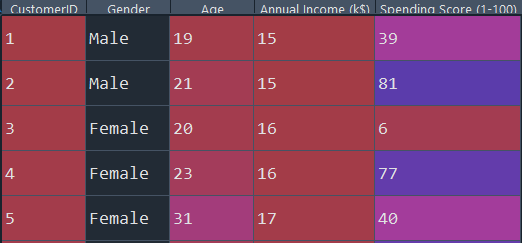
Aditya Bikram – 2020fc04534

Ashwini BR – 2020fc04180

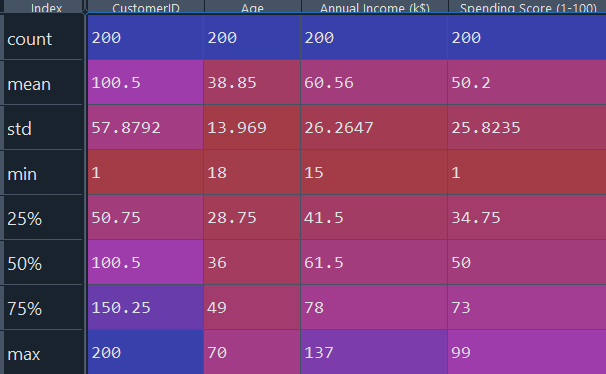
**Data Understanding:**

Data set is of the dimension 200 x 5 (200 rows and 5 columns). Columns contains the CustomerID, Gender, Age, Annual Income (k$) and Spending Score (1-100).

Below is the snippet of sample data:

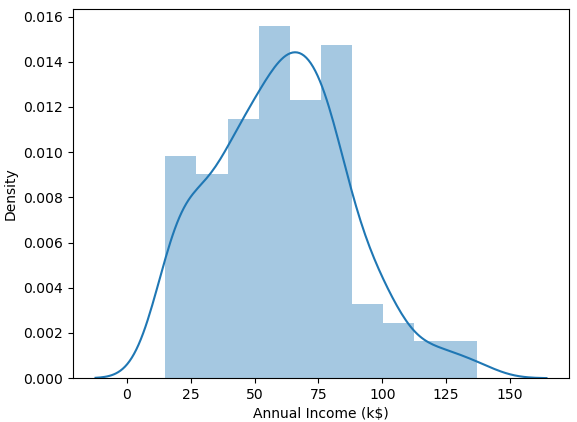
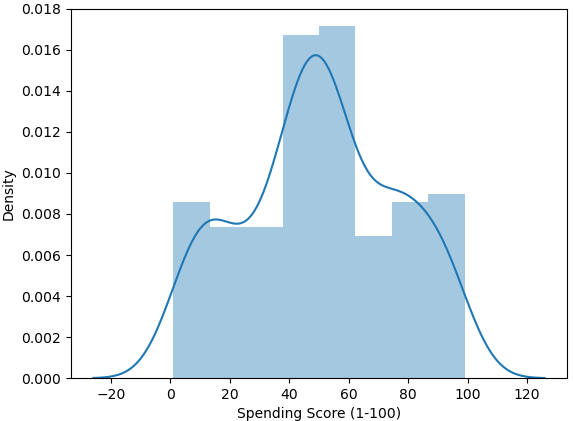


Below are the details of the data which includes it’s mean, median and mode along with min and max.



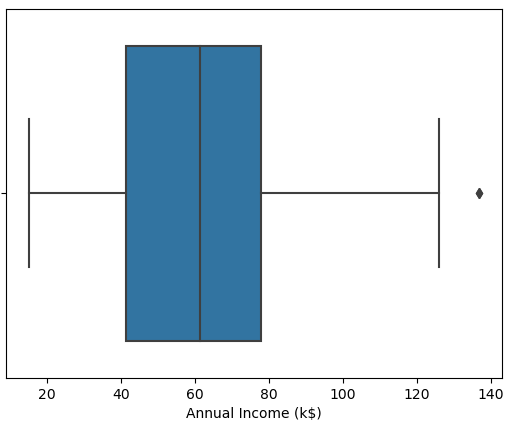
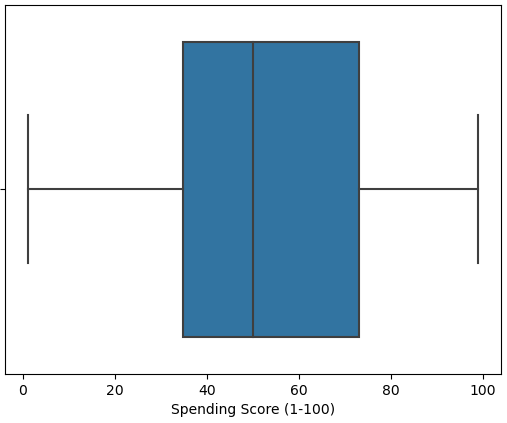
Data Pre-Processing: Data does not contain any null values and looks clean. However, in order to perform clustering, we have to transform gender to numeric value therefore label encoding will be required. Since other numeric fields are in comparable range, therefore, scaling is not required.

Checking the data distribution: Checking the distribution might tell us if there in any outliers and how the data is spread, if there is any skewness in the data?



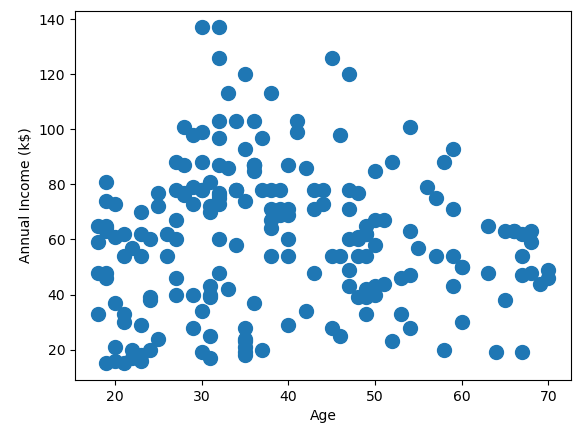
It is observed that numerical columns (annual income and spending score) and close to uniformly distributed without skewness therefore no log transforms are required.

Checking Outliers: Since the problem statement is related to segmentation, therefore if the outliers are very few (4-6 in numbers), we would remove it instead of imputing them. Imputing would cause tampering of data which might lead to wrong segmentations.

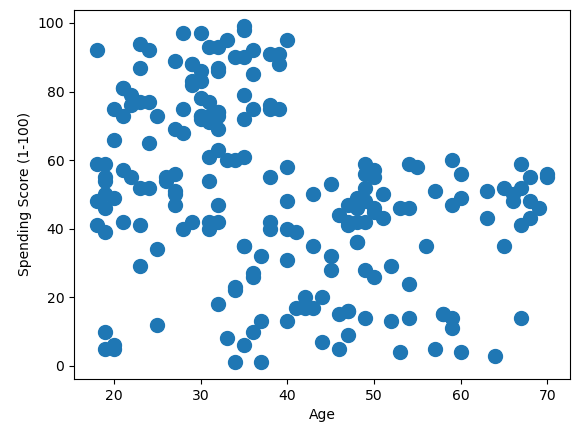
To understand the outliers, we are using box plot:

It is observed that numeric variable Annual income has only 2 outliers and we have removed it from the dataset. Moving onwards, we will be having 198 x 5 dimension of data.

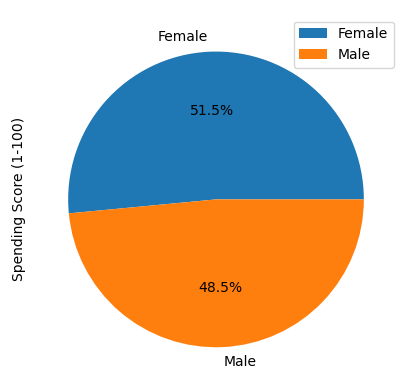
Bi-Variate Analysis: We have planned to perform Bi-Variate analysis in order to understand the relation between the variables.



The above scatter plot shows the relation between age and annual income. Interestingly, it’s been noted that people of the age between 30 to 60 are likely to have high income.



Interestingly it’s also noted that people with age less that 30 have higher spending score.

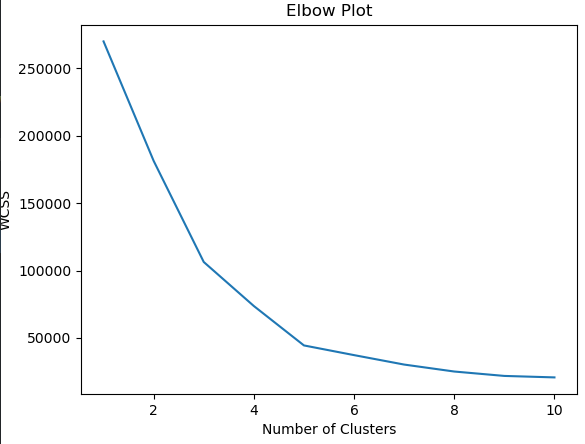
We can create a hypothesis that young people (between the age 20-30) are more likely to spend as compared to others.

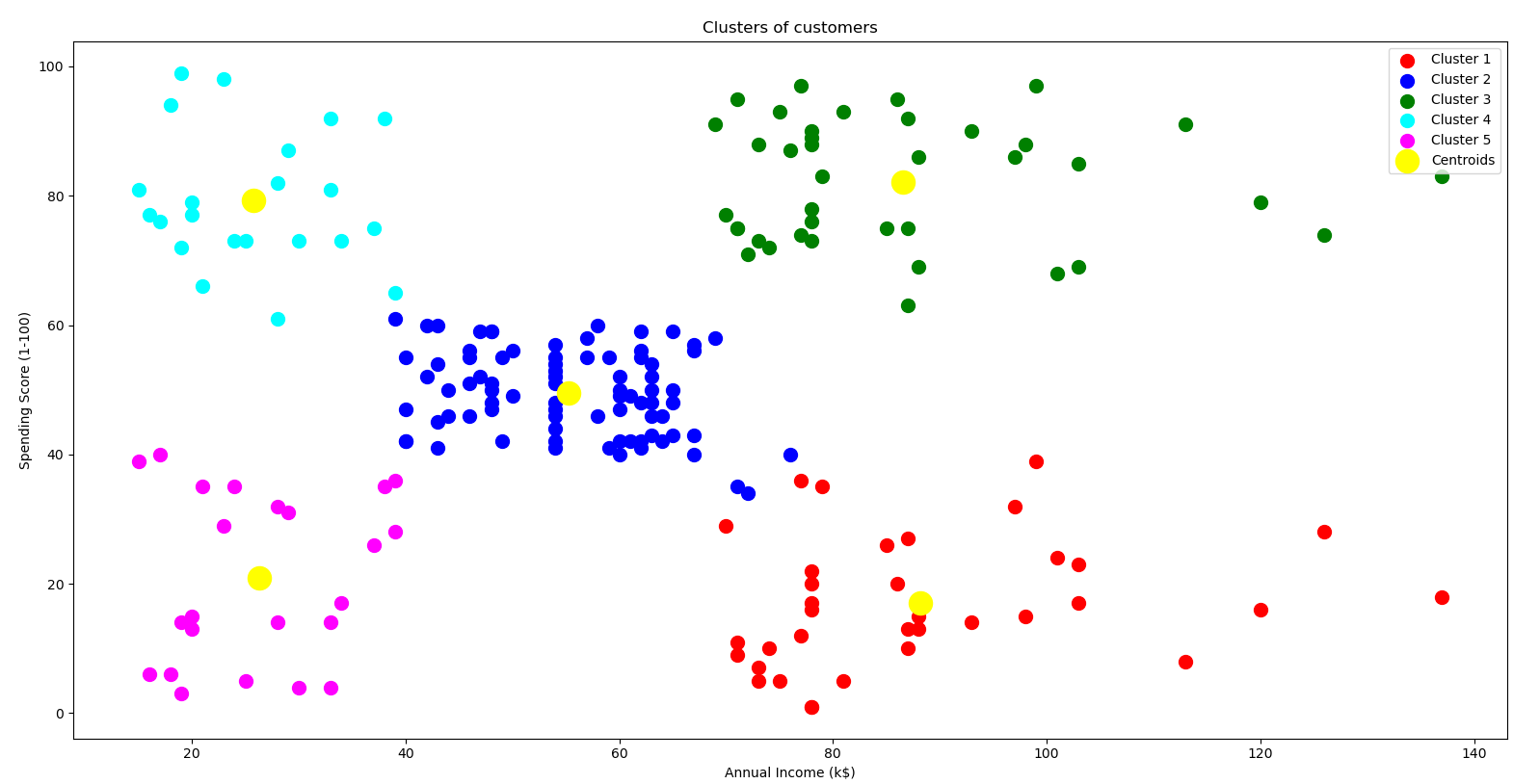
In terms of gender, females have higher average spending scores. Therefore, our hypothesis is - marketing team can focus more on females with the age between 20-30 as they have highest probability to purchase.

Clustering:

Clustering is a technique to group the similar featured data. In this case, we are using K-Means clustering and taking the numeric features of Annual Income (k$) and Spending Score (1-100).

We have used within sum of square method to plot elbow plot to analyze how many clusters would be formed. Below is the elbow plot:



By looking at the elbow plot, we can deduce that the sharpest curve is at the fifth cluster. Therefore, we are choosing the 5 number of clusters in this case.

**Cluster Analysis:**

By looking at the cluster figure we can conclude that annual income and spending scores are forming 5 clusters and we can name them as follows:

Cluster 1: High Income and Low Spending Score – Sleeping

Cluster 2: Average Income and Average Spending Score – Regular

Cluster 3: High Income and High Spending Score – Champions

Cluster 4: Low Income and High Spending Score – Loyal

Cluster 5: Low Income and Low Spending Score – At Risk

**Logic Behind Nomenclature:**

***Sleeping:*** Cluster 1 is sleeping because they annual income is high but spending score is low. If the marketing team would try to call them or message them, they might turn up and purchase the items.

***Regular***: Cluster 2 is regular because they seem to have average income and spending scores. They seem to be regular customer and purchasing very often.

***Champions:*** Cluster 3 is our champion customers as their spending score is the highest with high income. They seem to be the elite members and purchasing at high cost. They should be targeted with high promotions and offers.

***Loyal:*** Cluster 4 is our loyal customers because despite of having low income they are having high spending scores. They are not leaving us and continuing shopping from our store. They should be given rewards for staying with us.

***At Risk:*** Cluster 5 is at risk because they are having low income and low spending scores. It is more likely that they going to churn out soon.

**Offers:**

Different offers should be given to different clusters as they have different buying behavior.

Sleeping customers are having high income and they should be treated with high offers. They are more likely to spend amount on better quality and presentations.

Regular customers should be given small fixed number of points in order to make them special and retain them.

Champions should be given offers like lucky draw and Netflix account with certain discount as they can pay for the value additions.

Loyal customers should be treated specially and should be given additional discounts and referral codes where they can get extra points for referrals.

Customers at risk should be given offers like free home delivery and buy one get another at 50% off.

**Python Program for Customer Movement Simulation: (producer.py)**

from kafka import KafkaProducer

import json

import time

import random

from datetime import datetime,timedelta

def cust\_movement():

cust\_id=random.randint(1, 200)

cust\_time=datetime.now()-timedelta(minutes=random.randint(1,20))

cust\_long=random.randrange(77,79)

cust\_lat=random.randrange(16,18)

return cust\_id,cust\_time,cust\_long,cust\_lat

def json\_serializer(data):

return json.dumps(data,default=defaultconverter).encode("utf-8")

def defaultconverter(o):

if isinstance(o, datetime):

return o.\_\_str\_\_()

producer = KafkaProducer(bootstrap\_servers=['localhost:9092'], value\_serializer=json\_serializer)

if \_\_name\_\_ == "\_\_main\_\_":

while 1==1 :

cust\_data = cust\_movement()

producer.send("cust-movement",cust\_data)

time.sleep(4)

**Overview and Walkthrough of DataPipeline**

Tools and versions

* + Apache Kafka – 2.13
  + Apache Spark – 2.3.4
  + Python – 3.7.6
  + Java - 1.8

Python Modules:

* Kafka-python
* Pyspark
* Sparksql-magic
* Pandas
* findspark

For compatibility purposes, older versions of some of the tools are used.

Kafka is used as message queue to hold the continuously flowing streaming data with customer movements. It is also used to hold the output of the Stream Processor on which further actions could be taken.

The Input stream data can be redirected to a database, S3 bucket or a simple text file for archiving purposes.

Spark Streaming is used for StreamProcessing.

**Graphical user interface, diagram, application

Description automatically generated**

**Business Logic:**

* The Streaming Analytics System is designed to track customers movement in and around the mall area in the last 20 minutes.
* Location information in terms of Longitude and Latitude information is assumed to be streamed from customers phone along with customer\_id and timestamp
* The streaming systems checks for customers within a threshold distance of 100 meters and depending on the customer segmentation type, corresponding offers are released to customers
* Customer Segmentation is determined by clustering using on pre-captured data based on Annual Income and Spending Score
* The streaming data is joined with the customer segmentation information to enable businesses strategize different offers and promotions

**Programs and Queries:**

* Producer.py 🡪 Code to simulate customer movement and send messages to kafka topic: cust-movement
* mallStream.py🡪 Code to read the input stream, process it and write output stream to kafka topic: potential-cust
* consumer.py 🡪 Code to read the final stream processed data

**Start/Stop Kafka:**

Open a cmd window and run

%KAFKA\_HOME%/bin/windows/zookeeper-server-start.bat %KAFKA\_HOME%/config/zookeeper.properties

Open another cmd window and run

%KAFKA\_HOME%/bin/windows/kafka-server-start.bat %KAFKA\_HOME%/config/server.properties

To stop, hit Ctrl+C or close the windows command terminal.

**Create Kafka Topics**

%KAFKA\_HOME%/bin/windows/kafka-topics.bat --create --topic cust-movement --bootstrap-server localhost:9092

%KAFKA\_HOME%/bin/windows/kafka-topics.bat --create --topic potential-cust --bootstrap-server localhost:9092

**Description of Topics**

**Graphical user interface

Description automatically generated**

**Read the events in Kafka Topic:**

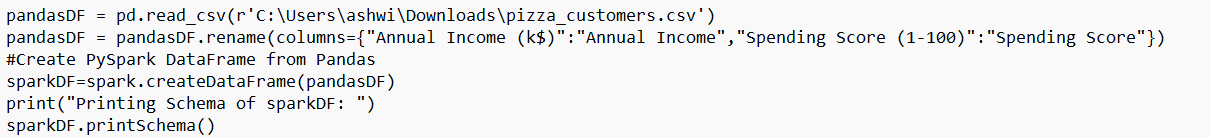
%KAFKA\_HOME%/bin/windows/kafka-console-consumer.bat --topic cust-movement --from-beginning --bootstrap-server localhost:9092

%KAFKA\_HOME%/bin/windows/kafka-console-consumer.bat --topic potential-cust --from-beginning --bootstrap-server localhost:9092

**Stream Processor triggering script:**

spark-submit --packages org.apache.spark:spark-sql-kafka-0-10\_2.11:2.4.3 mallStream.py

**Part of the code where csv is file is read into Spark Dataframe “sparkDF” to be joined with streaming data later**



Schema of sparkDF:

Text

Description automatically generated

Schema of streamed data read from input stream:

Text

Description automatically generated

Schema after combining Stream and csv data:

A picture containing text, plaque

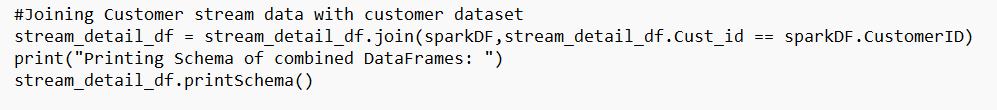
Description automatically generated

**Part of code where customers within 100m range are filtered**

**Text

Description automatically generated**

**Part of the code where Stream Data and csv data are joined**

****

**Part of code where customer segmentation is determined in almost real-time using Stream Processing Analytics.**

**Chart, scatter chart

Description automatically generated**