**Abstract**

In this project of Customer Segmentation, we were given a Dataset of Customers. We have first explained the theory of Unsupervised Learning, Customer Segmentation, Clustering and Different Clustering algorithms. We then explained the different Existing Methods and later Proposed some methods that can be used along with Architecture of those methods in the Customer Segmentation project. In later part of report, we have explained our Methodology to analyse and reach the desired result. Also, In the end we have included the Implementation of the project consisting of Code Snippets and Plots of different graphs.

**Table of Contents**

* **Introduction**
* **Existing Method**
* **Proposed method with Architecture**
* **Methodology**
* **Implementation**
* **Conclusion**

**Introduction**

Unsupervised Learning: In some pattern recognition problems, the training data consists of a set of input vectors x without any corresponding target values. The goal in such unsupervised learning problems may be to discover groups of similar examples within the data, where it is called **clustering**, or to determine how the data is distributed in the space, known as **density estimation**. To put forward in simpler terms, for a n-sampled space x1 to xn, true class labels are not provided for each sample, hence known as **learning without teacher**.

**Issues with Unsupervised Learning:**

* Unsupervised Learning is harder as compared to Supervised Learning tasks.
* How do we know if results are meaningful since no answer labels are available?
* Let the expert look at the results (external evaluation)
* Define an objective function on clustering (internal evaluation)

**Why Unsupervised Learning is needed despite of these issues?**

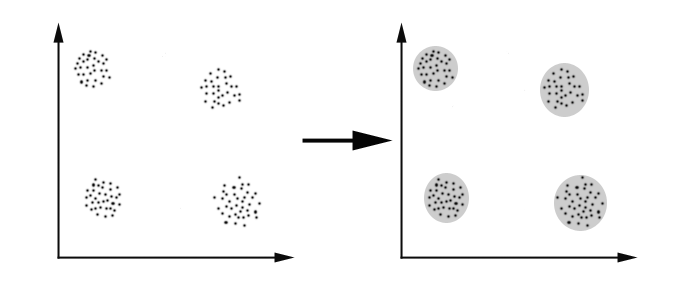
* Annotating large datasets is very costly and hence we can label only a few examples manually. Example: Speech Recognition
* There may be cases where we don’t know how many/what classes is the data divided into. Example: Data Mining
* We may want to use clustering to gain some insight into the structure of the data before designing a classifier.

Unsupervised Learning can be further classified into two categories:

* **Parametric Unsupervised Learning**: In this case, we assume a parametric distribution of data. It assumes that sample data comes from a population that follows a probability distribution based on a fixed set of parameters. Theoretically, in a normal family of distributions, all members have the same shape and are parameterized by mean and standard deviation. That means if you know the mean and standard deviation, and that the distribution is normal, you know the probability of any future observation. Parametric Unsupervised Learning involves construction of Gaussian Mixture Models and using Expectation-Maximization algorithm to predict the class of the sample in question. This case is much harder than the standard supervised learning because there are no answer labels available and hence there is no correct measure of accuracy available to check the result.
* **Non-parametric Unsupervised Learning**: In non-parameterized version of unsupervised learning, the data is grouped into clusters, where each cluster(hopefully) says something about categories and classes present in the data. This method is commonly used to model and analyse data with small sample sizes. Unlike parametric models, nonparametric models do not require the modeller to make any assumptions about the distribution of the population, and so are sometimes referred to as a distribution-free method.

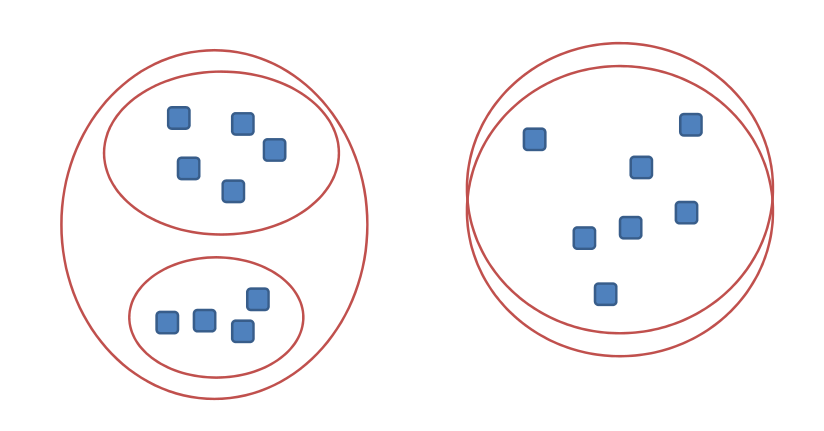
### Customer Segmentation: Customer Segmentation is the process of division of customer base into several groups of individuals that share a similarity in different ways that are relevant to marketing such as gender, age, interests, and miscellaneous spending habits. Companies that deploy customer segmentation are under the notion that every customer has different requirements and require a specific marketing effort to address them appropriately. Companies aim to gain a deeper approach of the customer they are targeting. Therefore, their aim has to be specific and should be tailored to address the requirements of each and every individual customer. Furthermore, through the data collected, companies 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, they can strategize their marketing techniques more efficiently and minimize the possibility of risk to their investment.

The technique of customer segmentation is dependent on several key differentiators that divide customers into groups to be targeted. Data related to demographics, geography, economic status as well as behavioural patterns play a crucial role in determining the company direction towards addressing the various segments.

**Clustering:** Clustering can be considered the most important **unsupervised learning** problem; so, as every other problem of this kind, it deals with finding a **structure** in a collection of unlabelled data. A loose definition of clustering could be “the process of organizing objects into groups whose members are similar in some way”. A **cluster** is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters.

**Distance-based clustering**: Given a set of points, with a notion of distance between points, grouping the points into some number of clusters, such that

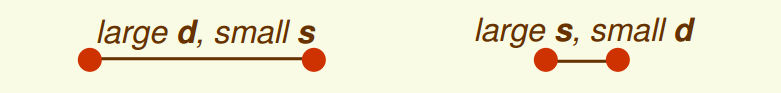
* internal (within the cluster) distances should be small i.e. members of clusters are close/similar to each other.
* external (intra-cluster) distances should be large i.e. members of different clusters are dissimilar.

**The Goals of Clustering**: The goal of clustering is to determine the internal grouping in a set of unlabelled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute “best” criterion which would be independent of the final aim of the clustering. Consequently, it is the user who should supply this criterion, in such a way that the result of the clustering will suit their needs.

In the above image, how do we know what is the best clustering solution?

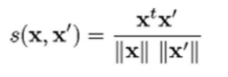
To find a particular clustering solution, we need to define the similarity measures for the clusters.

**Proximity Measures**: For clustering, we need to define a proximity measure for two data points. Proximity here means how similar/dissimilar the samples are with respect to each other.

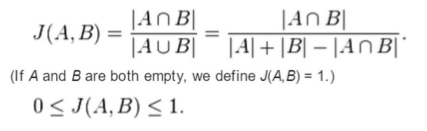
* Similarity measure S(xi,xk): large if xi,xk are similar
* Dissimilarity(or distance) measure D(xi,xk): small if xi,xk are similar

There are various similarity measures which can be used.

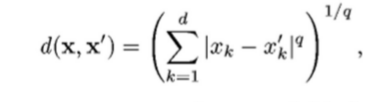
* *Vectors: Cosine Distance*



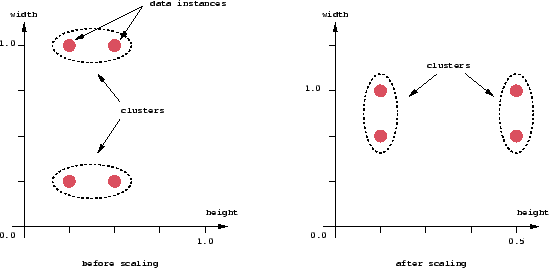
* *Sets: Jaccard Distance*



* *Points: Euclidean Distance* (q=2)



A “good” proximity measure is VERY application dependent. The clusters should be invariant under the transformations “natural” to the problem.



**Problems associated with clustering**: There are a number of problems with clustering. Among them:

* Dealing with large number of dimensions and large number of data items can be problematic because of time complexity;
* The effectiveness of the method depends on the definition of “distance” (for distance-based clustering). If an obvious distance measure doesn’t exist we must “define” it, which is not always easy, especially in multidimensional spaces;
* The result of the clustering algorithm (that in many cases can be arbitrary itself) can be interpreted in different ways.

**Possible Applications**: Clustering algorithms can be applied in many fields, for instance:

* Marketing: Finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records;
* Biology: Classification of plants and animals given their features;
* Insurance: Identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds.

**Existing Method**

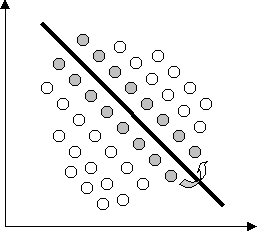
The problems in Pattern Recognition and Machine Learning can be of various types, they can be broadly classified into three categories:

* **Supervised Learning:**  
  The system is presented with example inputs and their desired outputs, given by a “teacher”, and the goal is to learn a general rule that maps inputs to outputs.
* **Unsupervised Learning:**  
  No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).
* **Reinforcement Learning:**  
  A system interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). The system is provided feedback in terms of rewards and punishments as it navigates its problem space.

Clustering Algorithms: Clustering algorithms may be classified as:

* Exclusive Clustering
* Overlapping Clustering
* Hierarchical Clustering
* Probabilistic Clustering

In the first case data are grouped in an exclusive way, so that if a certain data point belongs to a definite cluster then it could not be included in another cluster. A simple example of that is shown in the figure below, where the separation of points is achieved by a straight line on a bi-dimensional plane.



On the contrary, the second type, the overlapping clustering, uses fuzzy sets to cluster data, so that each point may belong to two or more clusters with different degrees of membership. In this case, data will be associated to an appropriate membership value.

A hierarchical clustering algorithm is based on the union between the two nearest clusters. The beginning condition is realized by setting every data point as a cluster. After a few iterations it reaches the final clusters wanted.

Finally, the last kind of clustering uses a completely probabilistic approach.

The four of the most used clustering algorithms:

* K-means
* Fuzzy K-means
* Hierarchical clustering
* Mixture of Gaussians

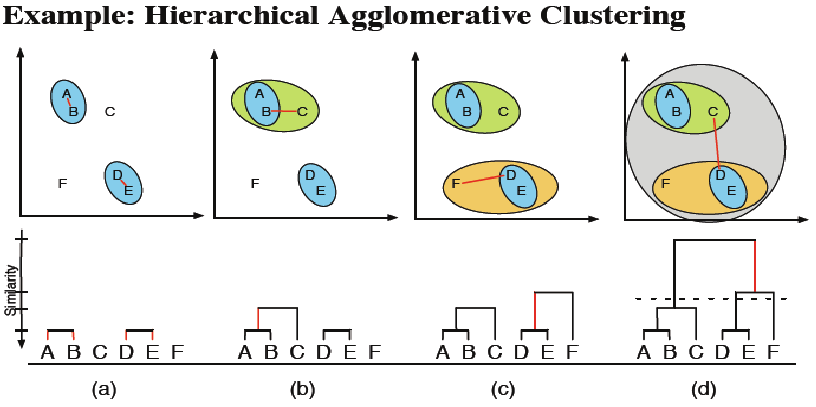
While, K-means is an exclusive clustering algorithm, Fuzzy K-means is an overlapping clustering algorithm, Hierarchical clustering is obvious and lastly Mixture of Gaussians is a probabilistic clustering algorithm. We will discuss about each clustering method in the following paragraphs.

Fuzzy K-Means Clustering: In fuzzy clustering, each point has a probability of belonging to each cluster, rather than completely belonging to just one cluster as it is the case in the traditional k-means. Fuzzy k-means specifically tries to deal with the problem where points are somewhat in between centres or otherwise ambiguous by replacing distance with probability, which of course could be some function of distance, such as having probability relative to the inverse of the distance. Fuzzy k-means uses a weighted centroid based on those probabilities. Processes of initialization, iteration, and termination are the same as the ones used in k-means. The resulting clusters are best analysed as probabilistic distributions rather than a hard assignment of labels. One should realize that k-means is a special case of fuzzy k-means when the probability function used is simply 1 if the data point is closest to a centroid and 0 otherwise.

In the Fuzzy k-means approach, instead, the same given data point does not belong exclusively to a well-defined cluster, but it can be placed in a middle way. In this case, the membership function follows a smoother line to indicate that every data point may belong to several clusters with different extent of membership.

Hierarchical Clustering Algorithms: Given a set of N items to be clustered, and an N\*N distance (or similarity) matrix, the basic process of hierarchical clustering is this:

* Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.
* Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
* Compute distances (similarities) between the new cluster and each of the old clusters.
* Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.



Clustering as a Mixture of Gaussians: There’s another way to deal with clustering problems: a model-based approach, which consists in using certain models for clusters and attempting to optimize the fit between the data and the model.

In practice, each cluster can be mathematically represented by a parametric distribution, like a Gaussian. The entire data set is therefore modelled by a mixture of these distributions.

A mixture model with high likelihood tends to have the following traits:

* component distributions have high “peaks” (data in one cluster are tight);
* the mixture model “covers” the data well (dominant patterns in the data are captured by component distributions).

Main advantages of model-based clustering:

* well-studied statistical inference techniques available;
* flexibility in choosing the component distribution;
* obtain a density estimation for each cluster;
* a “soft” classification is available.

**Proposed Method and Architecture**

We will focus on **Unsupervised Learning** and **Data Clustering**.

The libraries we will use in order to develop a solution for this problem are:

* numpy/pandas: Will help us treat and explore the data, and execute vector and matrix operations.
* Matplotlib/seaborn: Will help us plot the information so we can visualize it in different ways and have a better understanding of it.
* sklearn: Will provide all necessary tools to train our models and test them afterwards.

Among the four mentioned Clustering Algorithms, we will be using K-Means Clustering.

K-Means Clustering: K-means is one of the simplest unsupervised learning algorithms that solves the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centres, one for each cluster. These centroids should be placed in a smart way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycentre’s of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words, centroids do not move any more.

Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function



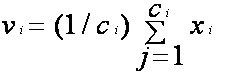
where



is a chosen distance measure between a data point xi and the cluster centre cj, is an indicator of the distance of the n data points from their respective cluster centres.

The algorithm is composed of the following steps:

* Let X = {x1,x2,x3,……..,xn} be the set of data points and V = {v1,v2,…….,vc} be the set of centers.
* Randomly select ‘c’ cluster centers.
* Calculate the distance between each data point and cluster centers.
* Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.
* Recalculate the new cluster center using:

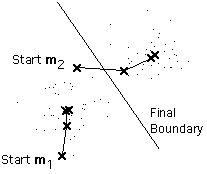


where, ‘ci’ represents the number of data points in ith cluster.

* Recalculate the distance between each data point and new obtained cluster centers.
* If no data point was reassigned then stop, otherwise repeat from step 3

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centres. The k-means algorithm can be run multiple times to reduce this effect.

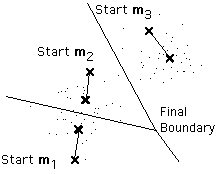
K-means is a simple algorithm that has been adapted to many problem domains. As we are going to see, it is a good candidate for extension to work with fuzzy feature vectors.



The k-means procedure can be viewed as a greedy algorithm for partitioning the n samples into k clusters so as to minimize the sum of the squared distances to the cluster centres. It does have some weaknesses:

* The way to initialize the means was not specified. One popular way to start is to randomly choose k of the samples.
* It can happen that the set of samples closest to mi is empty, so that mi cannot be updated. This is a problem which needs to be handled during the implementation, but is generally ignored.
* The results depend on the value of k and there is no optimal way to describe a best “k”.

This last problem is particularly troublesome, since we often have no way of knowing how many clusters exist. In the example shown above, the same algorithm applied to the same data produces the following 3-means clustering. Is it better or worse than the 2-means clustering?



Summing up the K-means clustering –

* We specify the number of clusters that we need to create.
* The algorithm selects k objects at random from the dataset. This object is the initial cluster or mean.
* The closest centroid obtains the assignment of a new observation. We base this assignment on the Euclidean Distance between object and the centroid.
* k clusters in the data points update the centroid through calculation of the new mean values present in all the data points of the cluster. The kth cluster’s centroid has a length of p that contains means of all variables for observations in the k-th cluster. We denote the number of variables with p.
* Iterative minimization of the total within the sum of squares. Then through the iterative minimization of the total sum of the square, the assignment stop wavering when we achieve maximum iteration. The default value is 10 that the R software uses for the maximum iterations.

### Determining Optimal Clusters: While working with clusters, you need to specify the number of clusters to use. You would like to utilize the optimal number of clusters. To help you in determining the optimal clusters, there are three popular methods –

* Elbow method
* Silhouette method
* Gap statistic

#### We used Elbow Method for determining Optimal Clusters.

**Elbow Method:** The main goal behind cluster partitioning methods like k-means is to define the clusters such that the intra-cluster variation stays minimum.

minimize(sum W(Ck)), k=1…k

where Ck represents the kth cluster and W(Ck) denotes the intra-cluster variation. With the measurement of the total intra-cluster variation, one can evaluate the compactness of the clustering boundary. We can then proceed to define the optimal clusters as follows –

First, we calculate the clustering algorithm for several values of k. This can be done by creating a variation within k from 1 to 10 clusters. We then calculate the total intra-cluster sum of square (**iss**). Then, we proceed to plot **iss** based on the number of k clusters. This plot denotes the appropriate number of clusters required in our model. In the plot, the location of a bend or a knee is the indication of the optimum number of clusters.

**Methodology**

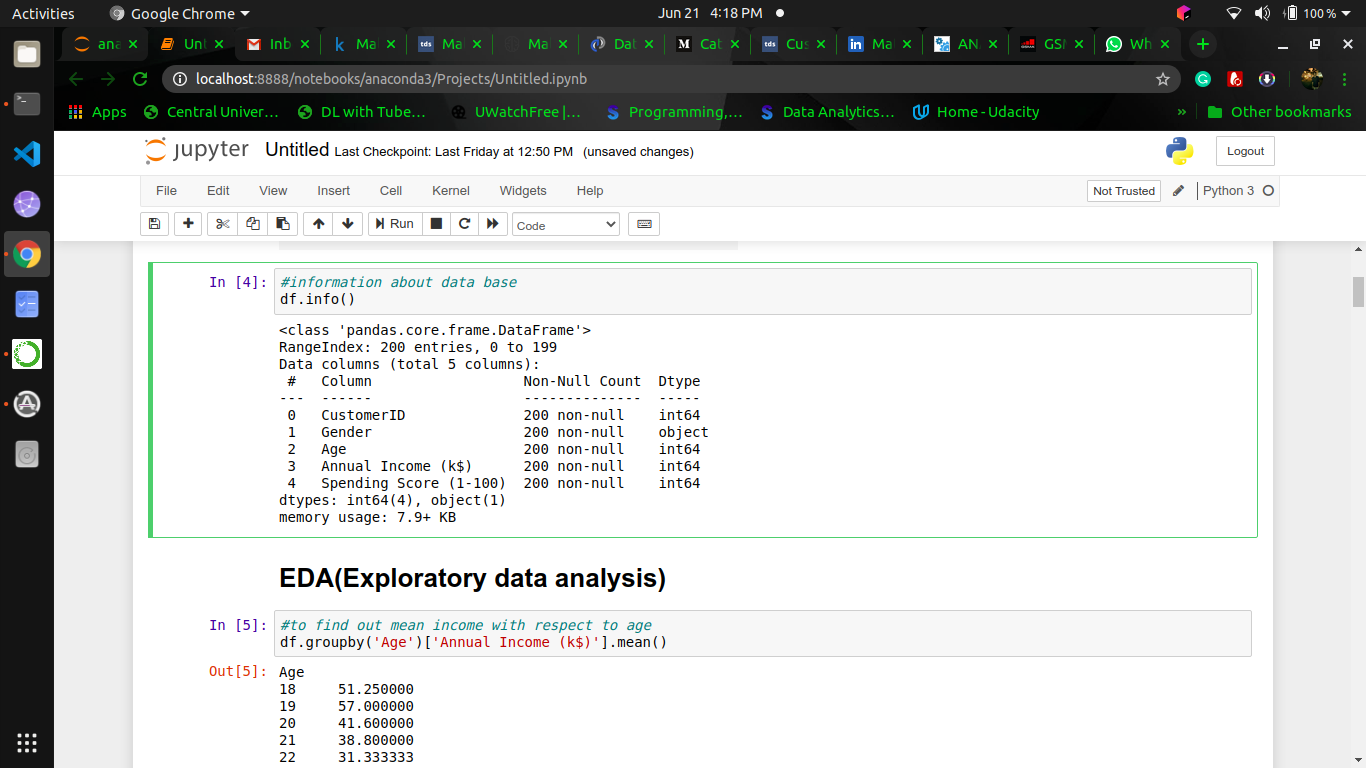
Consider that you have some basic data about your customers like Customer ID, age, gender, annual income and spending score, which is something you assign to the customer based on your defined parameters like customer behaviour and purchasing data.

The main aim of this problem is learning the purpose of the customer segmentation concepts, also known as market basket analysis, trying to understand customers and separate them in different groups according to their preferences, and once the division is done, this information can be given to marketing team so they can plan the strategy accordingly.

Column List

* Customer ID: Unique ID assigned to the customer
* Gender: Gender of the customer
* Age: Age of the customer
* Annual Income: Annual Income of the customer
* Spending Score: Score between (1-100) assigned by the mall based on customer behaviour and spending nature

### Data Exploration: In this section we are doing a little bit of data exploration, checking for null values, object data types and other things we might consider in order to keep our data clean and well structured.

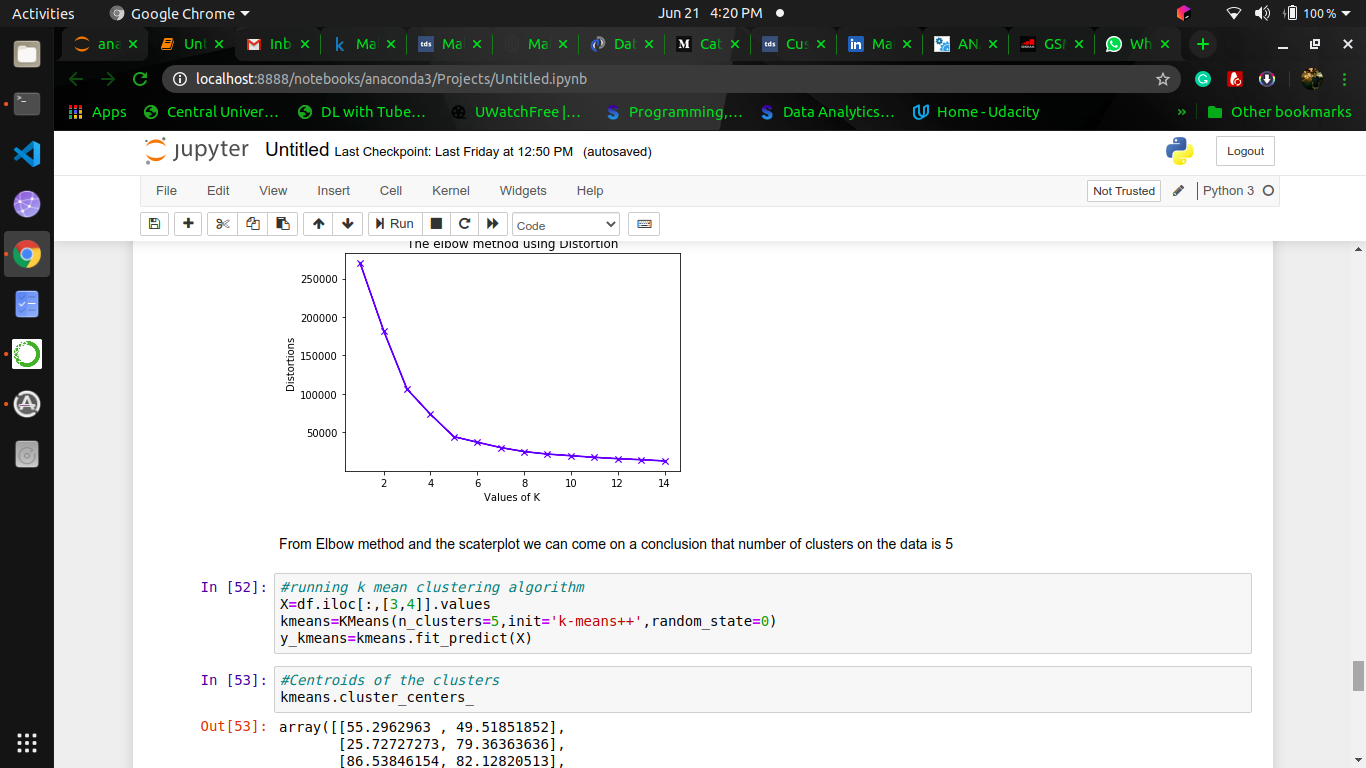


### 

### Data Visualization: Now it's the moment to visualize our data and plot important information so we can see the different values our data has and its behaviour. To do so, we are only going to consider the following features: Annual\_income, Spending\_score and Age. Gender will only be used to make data separation so we can differentiate values for men and women.

### Selecting Number of Clusters: To decide the amount of clusters, we are going to use the Elbow Method.

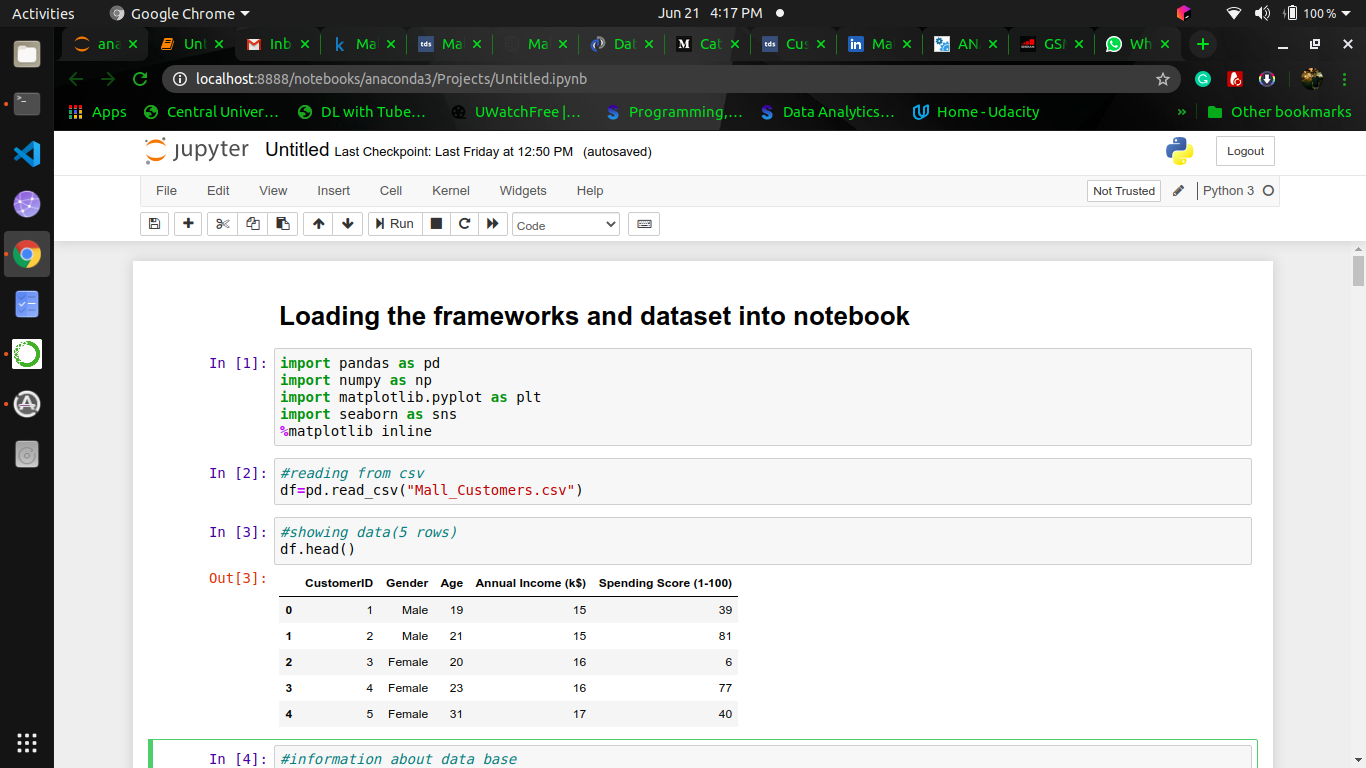
The elbow method is used to determine the optimal number of clusters in k-means clustering. The elbow method plots the value of the cost function produced by different values of k and one should choose a number of clusters so that adding another cluster doesn't give much better modelling of the data. In this problem, we are using the inertia as cost function in order to identify the sum of squared distances of samples to the nearest cluster centre.

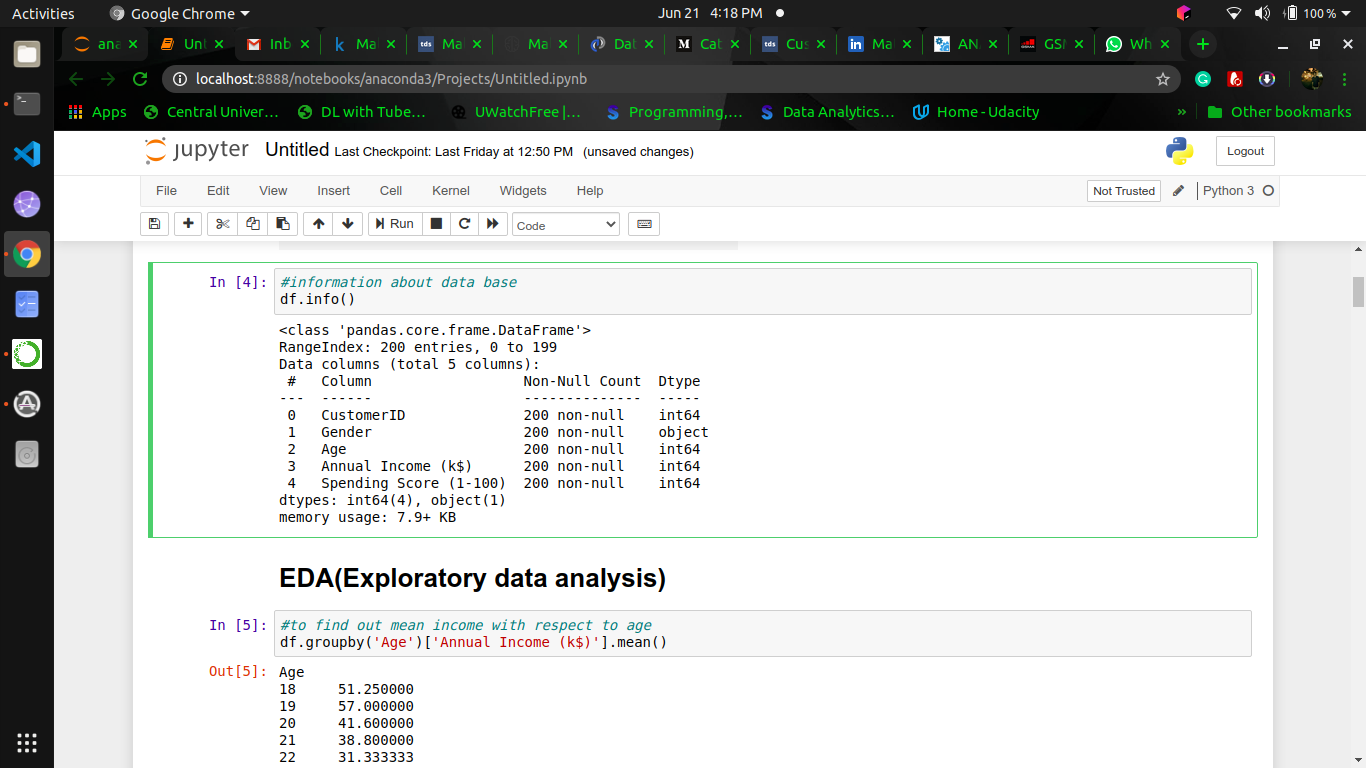


Looking at this particular example, if we imagine the line in the graphic is an arm, the elbow can be found, approximately, where the number of clusters is equal to 5. Therefore, we are selecting 5 as the number of clusters to divide our data in.

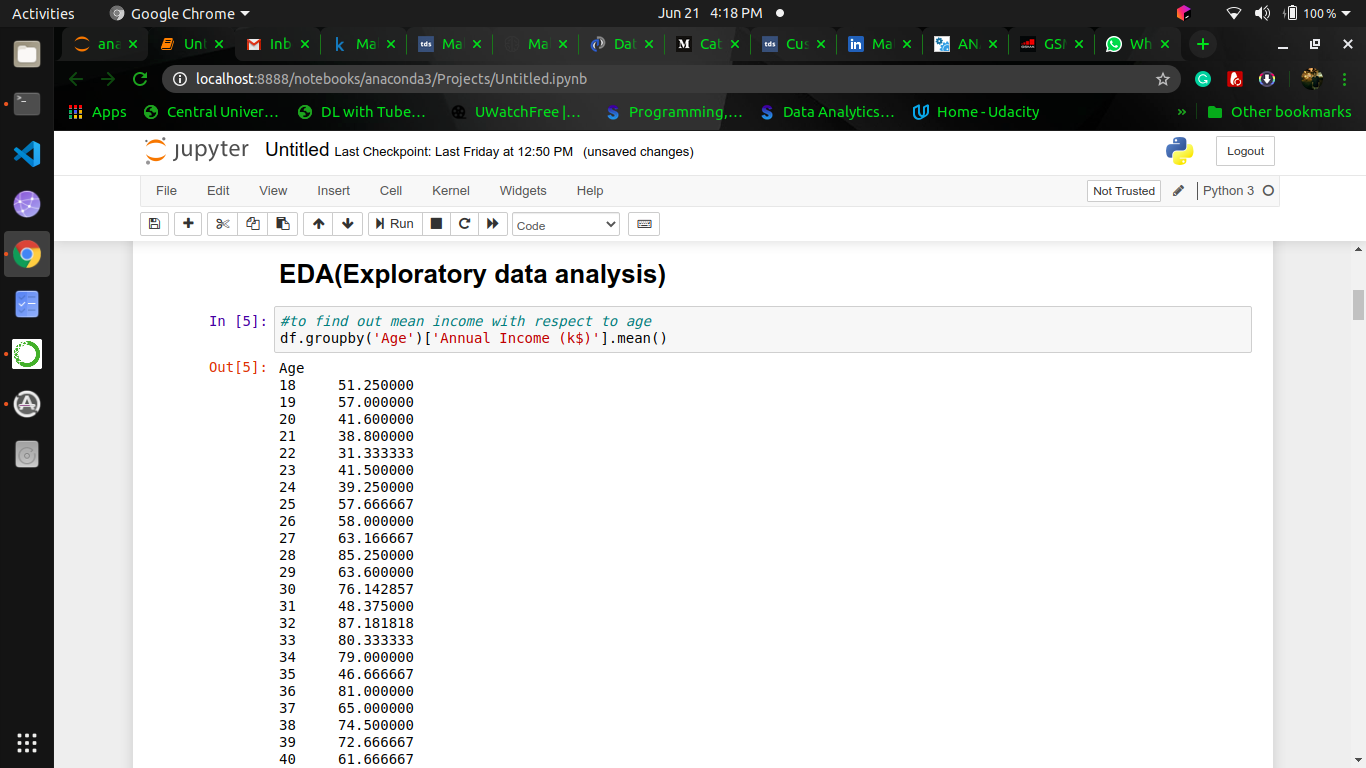
### Clustering: In the process of clustering we will not be considering the gender factor anymore. The first main reason of why we do take this approach is because the difference between male and female in this data is not particularly high and making a gender differentiation won't provide any further information. The second and not least important reason is the fact that stores, in general, hardly ever target a specific gender anymore, in almost every store in a shopping centre male and female products can be found. Additionally we do not want to interfere in the process of unsupervised learning, we will leave the algorithm do its job and once it's finished we will analyse the results and extract conclusions and knowledge.

**Implementation**

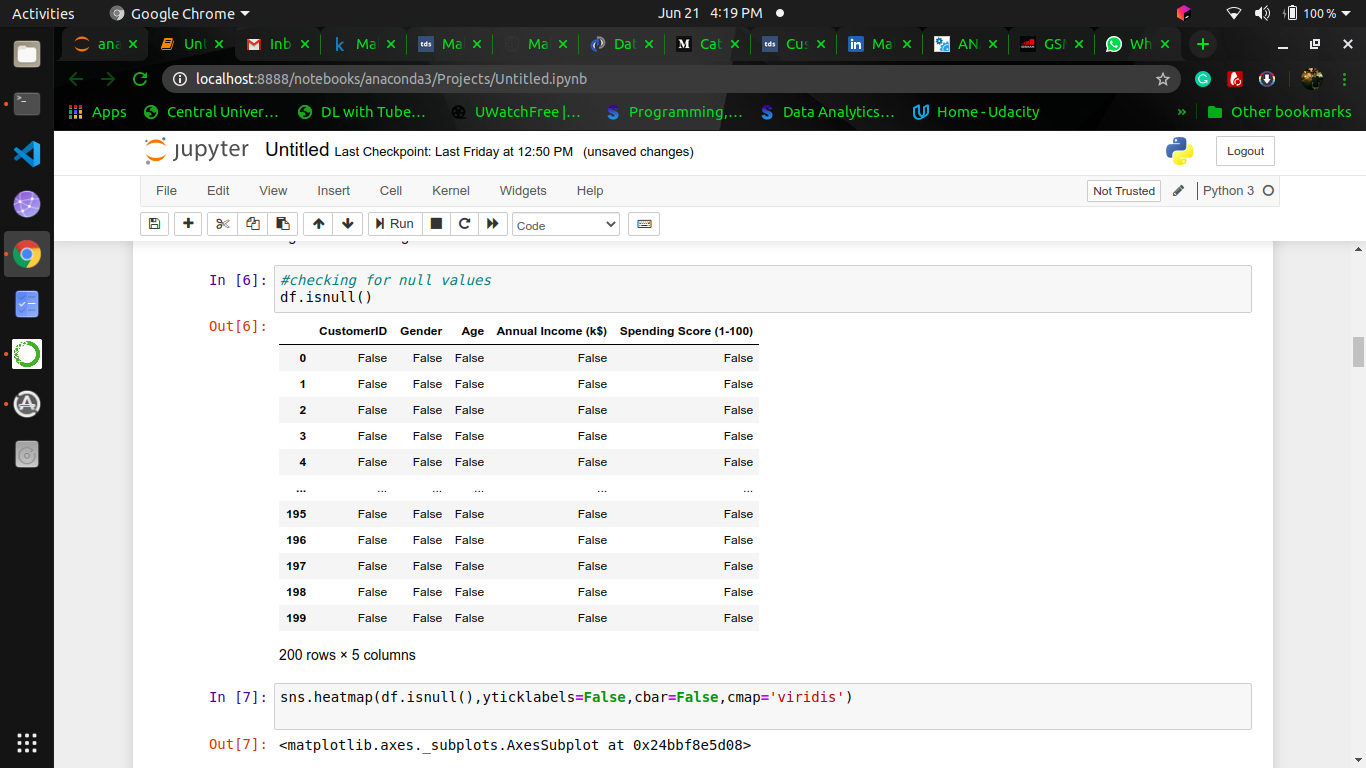
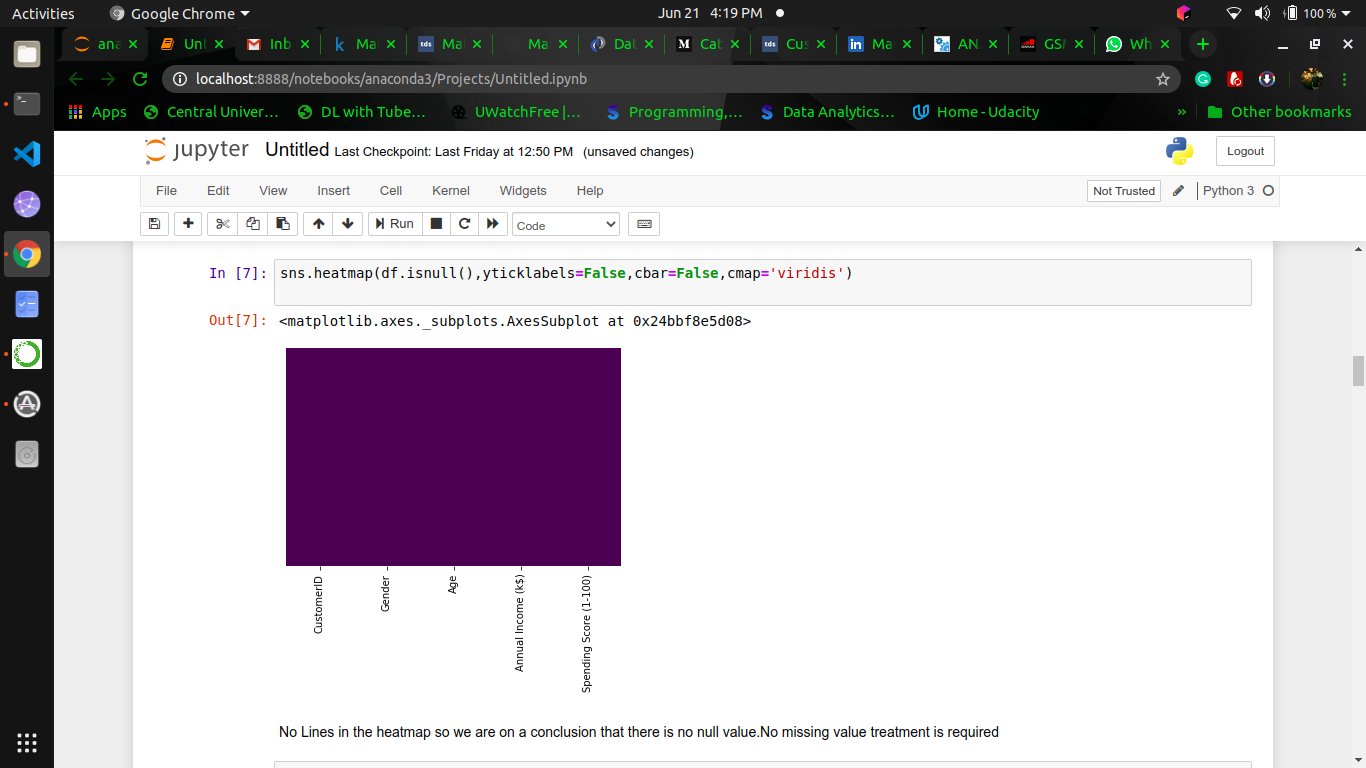
****



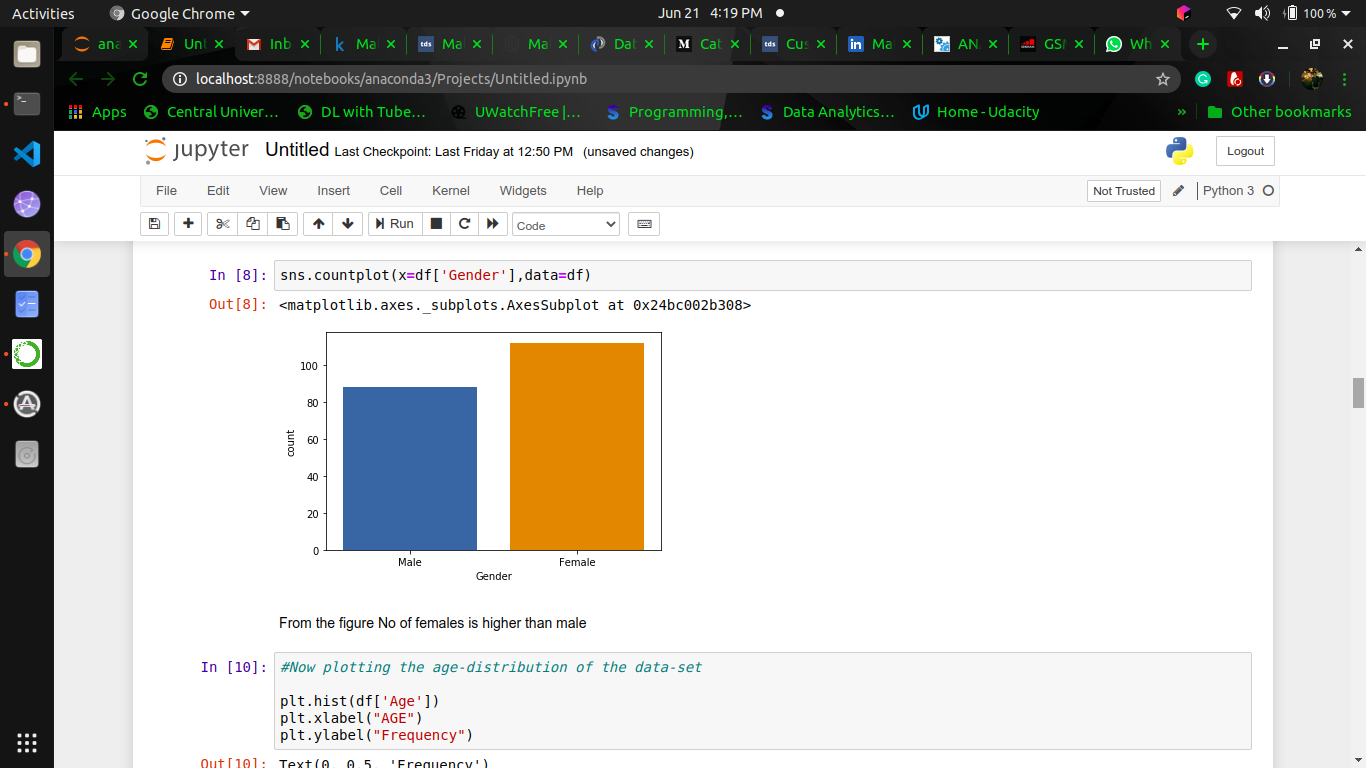
**Fig: Showing Information about database**

****

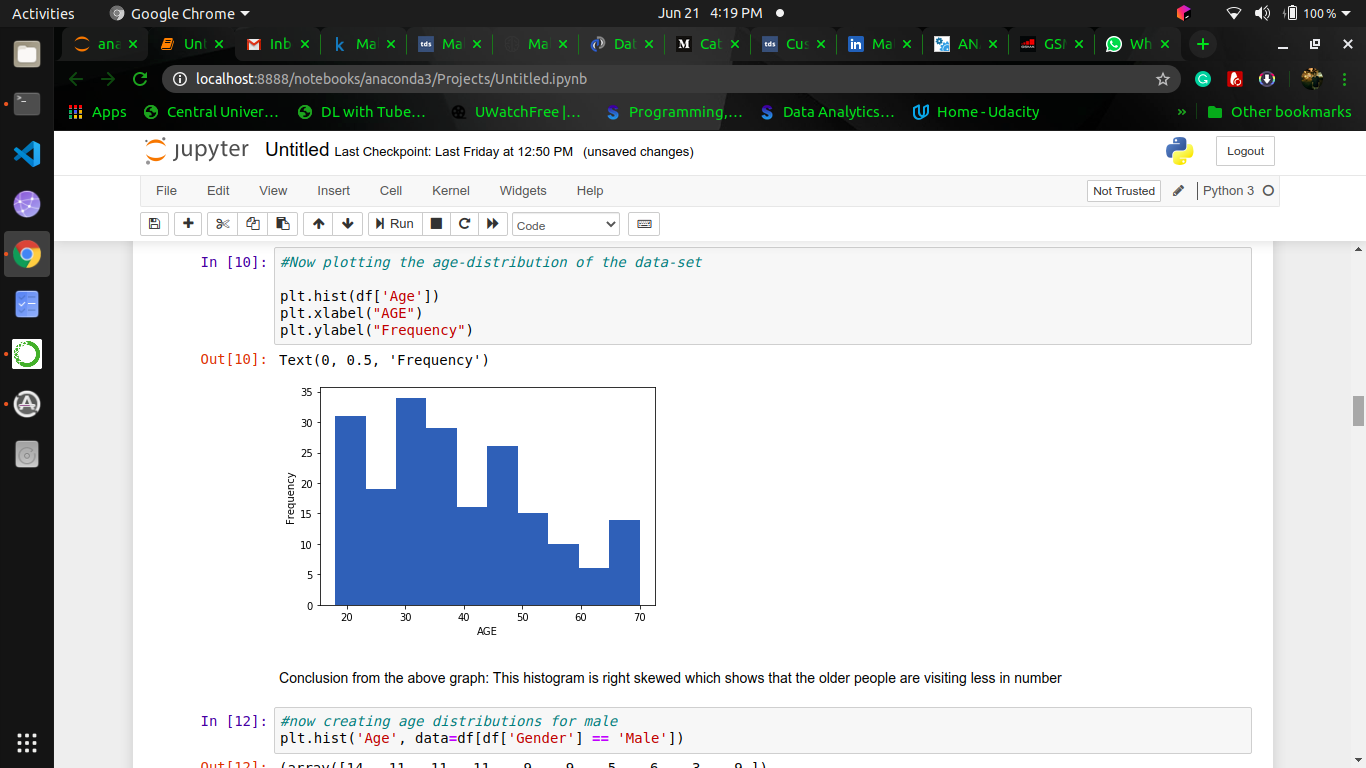
**Fig: Finding Mean income w.r.t age**

****

**Fig: Checking for Null Values in Dataset**

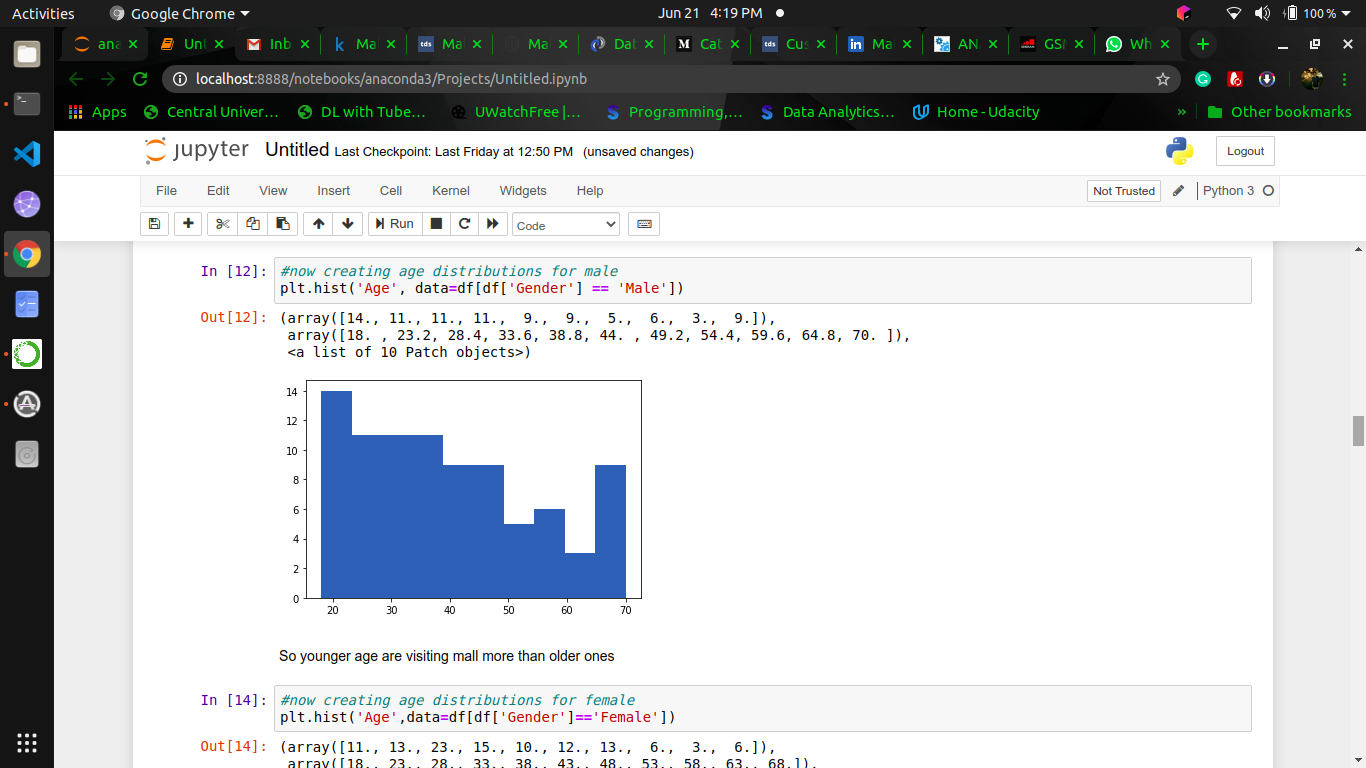
****

**Fig: Comparing Count of Male and Female**

****

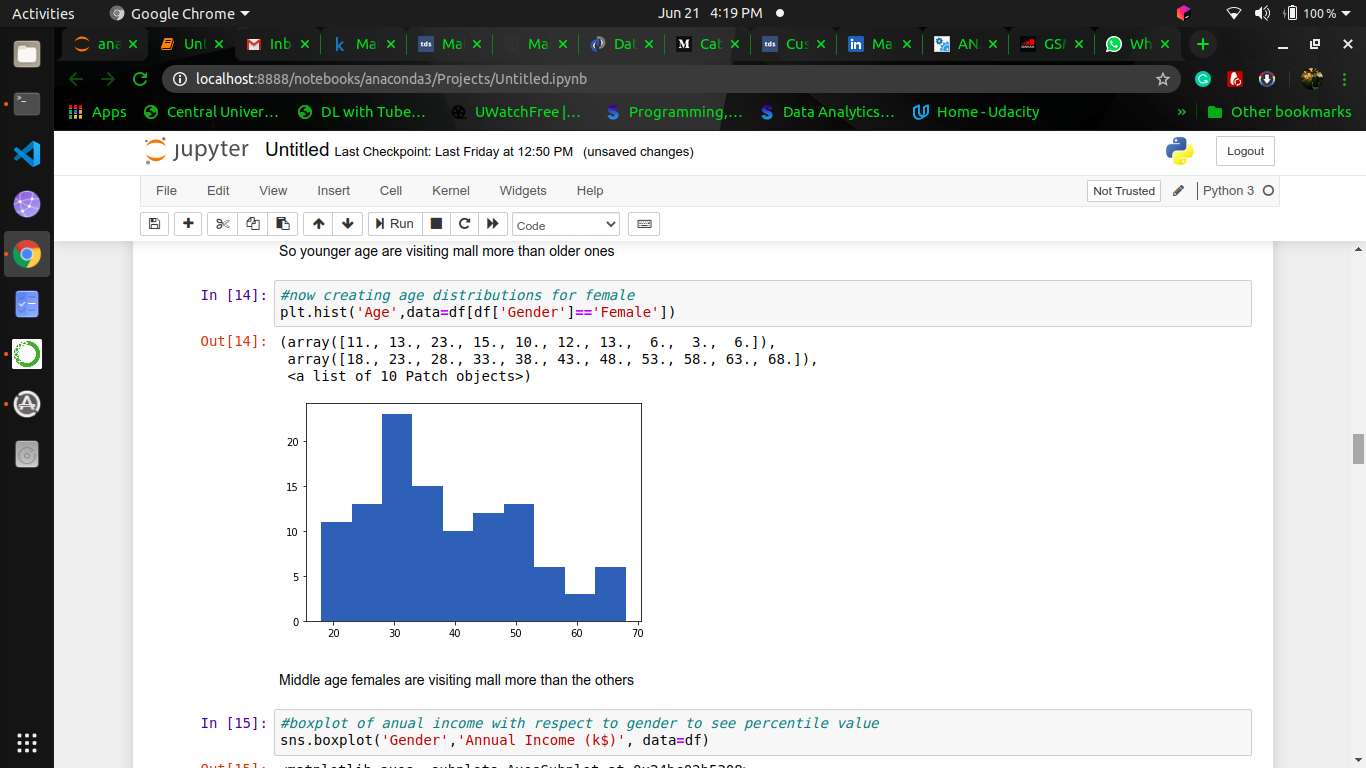
**Fig: Plotting Age Distribution**

**Conclusion: Histogram is Right Skewed meaning; Older people are visiting less.**

****

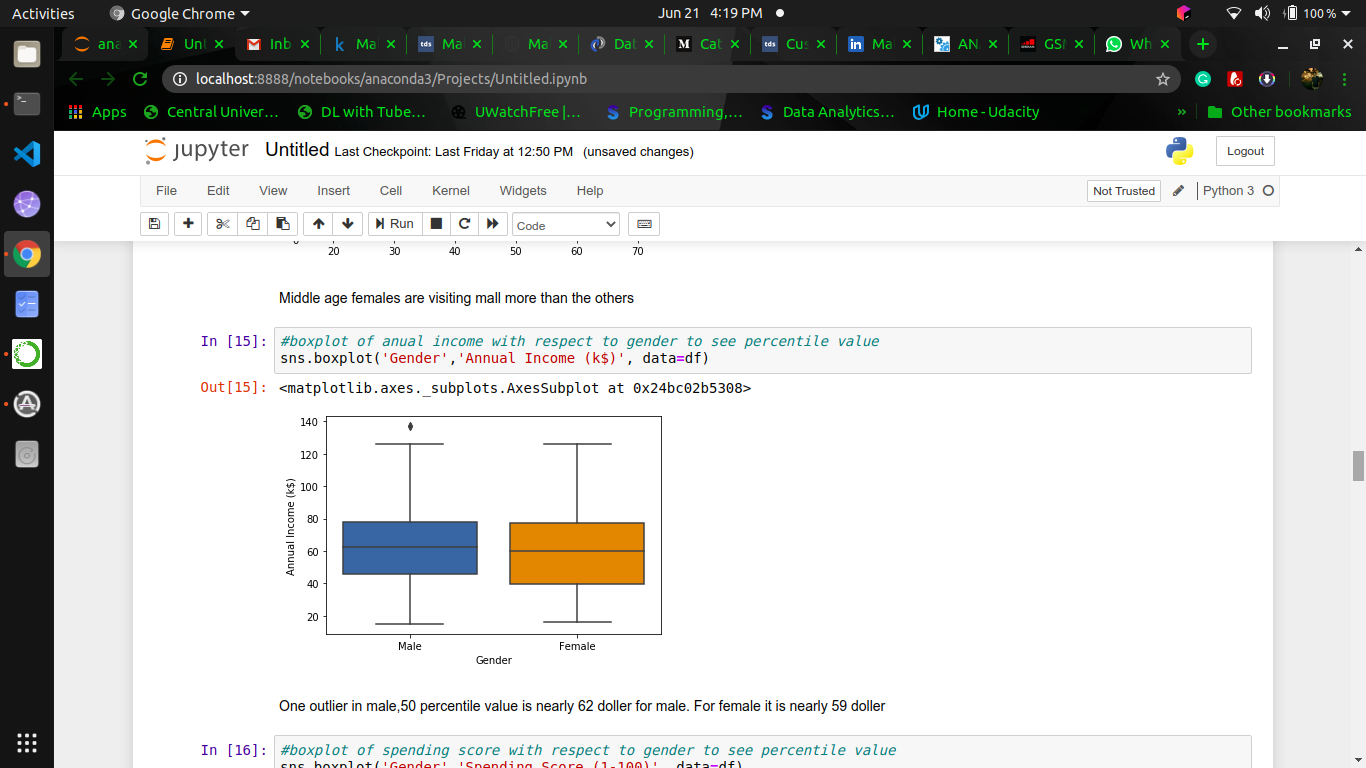
**Fig: Plotting Age Distribution (MALE)**

**Conclusion: Younger people are visiting more.**

****

**Fig: Plotting Age Distribution (FEMALE)**

**Conclusion: Middle Age Females are visiting more than others.**

****

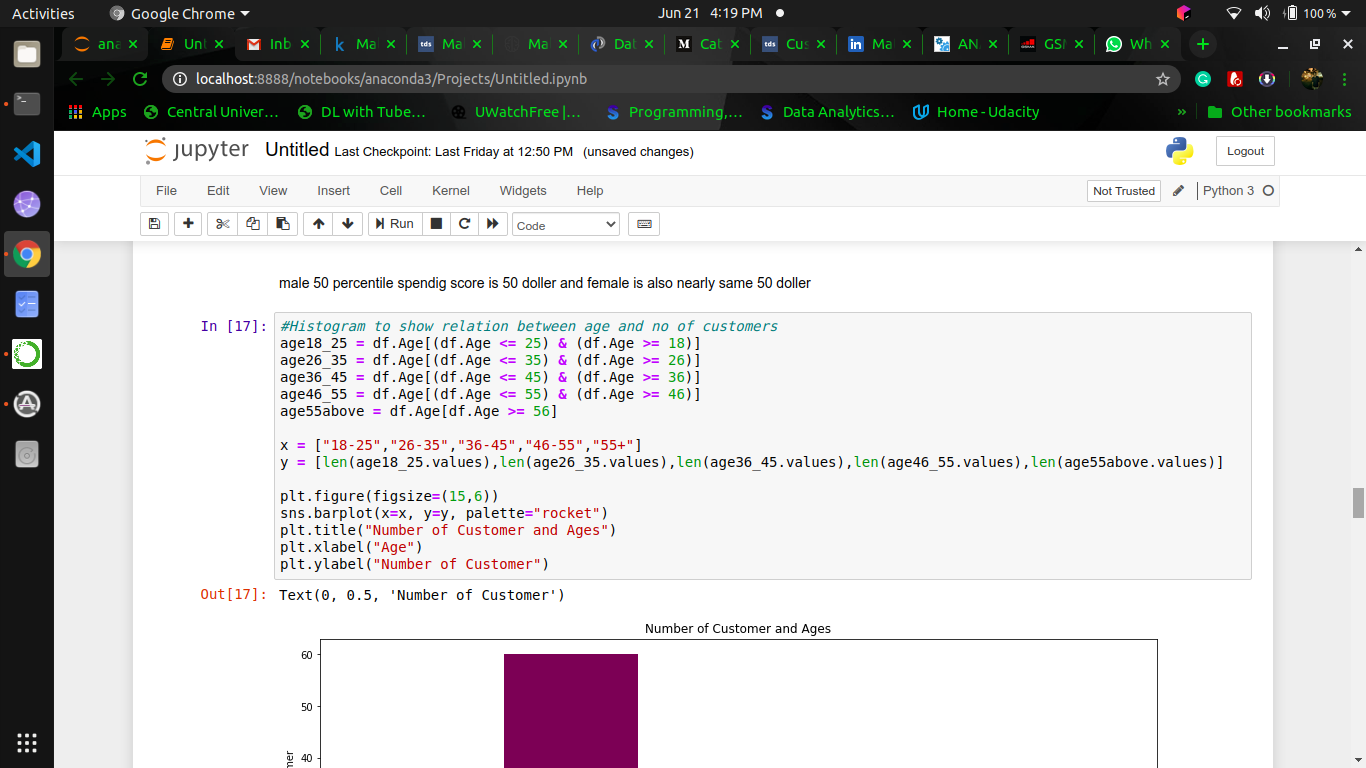
**Fig: Plotting Annual Income w.r.t Gender**

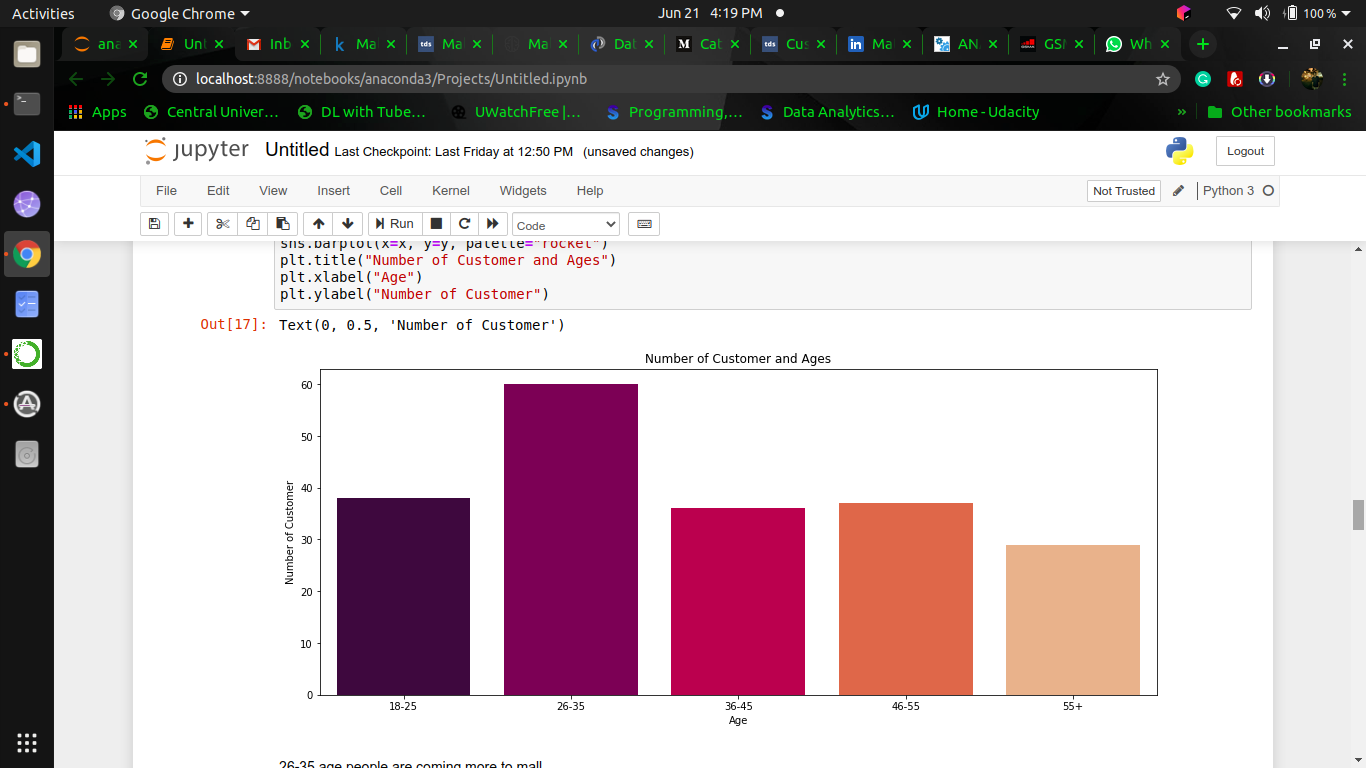
**Conclusion: One Outlier in Male. 50 percentile value is nearly 62k$ for male. For Female it is 59k$ (approx)**

****

**Fig: Plotting Spending Score w.r.t Gender**

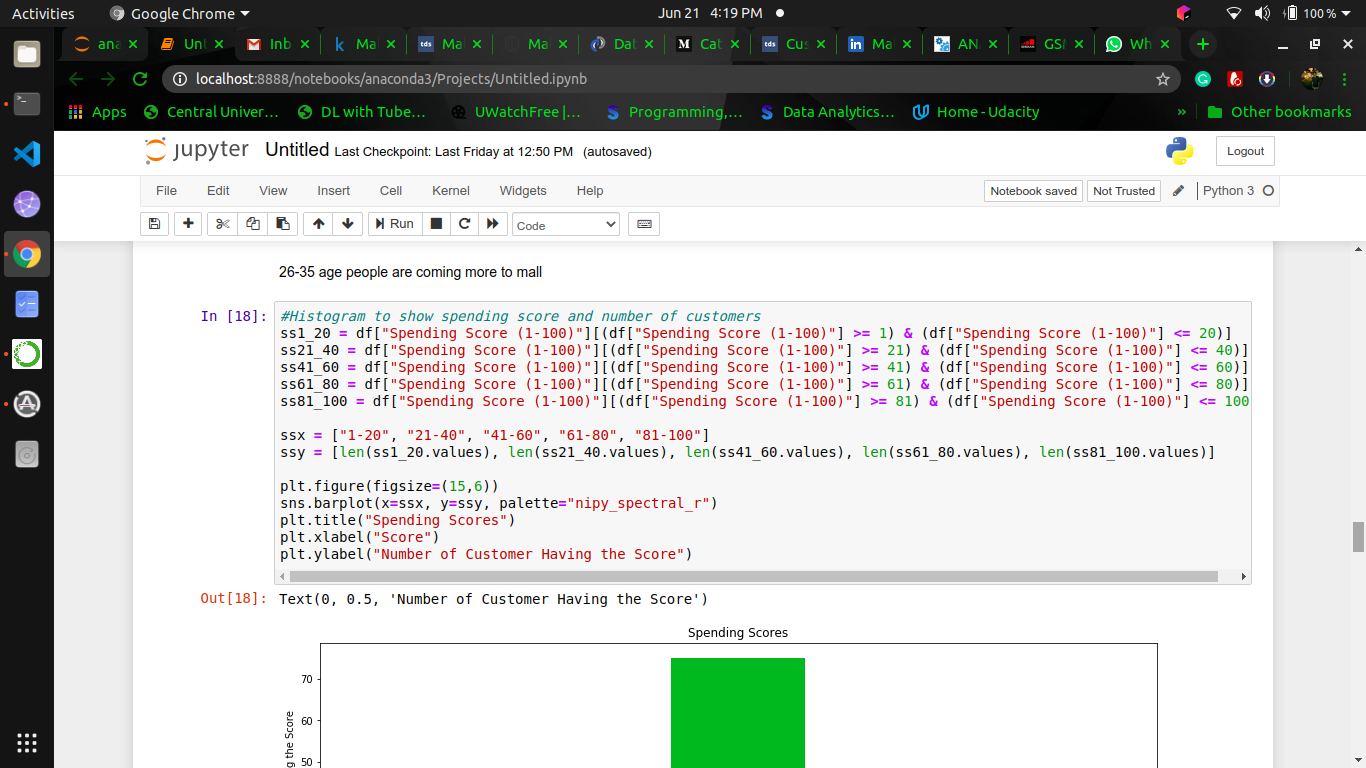
**Conclusion: Male 50 percentile score is 50. For Female it is 50 (approx)**

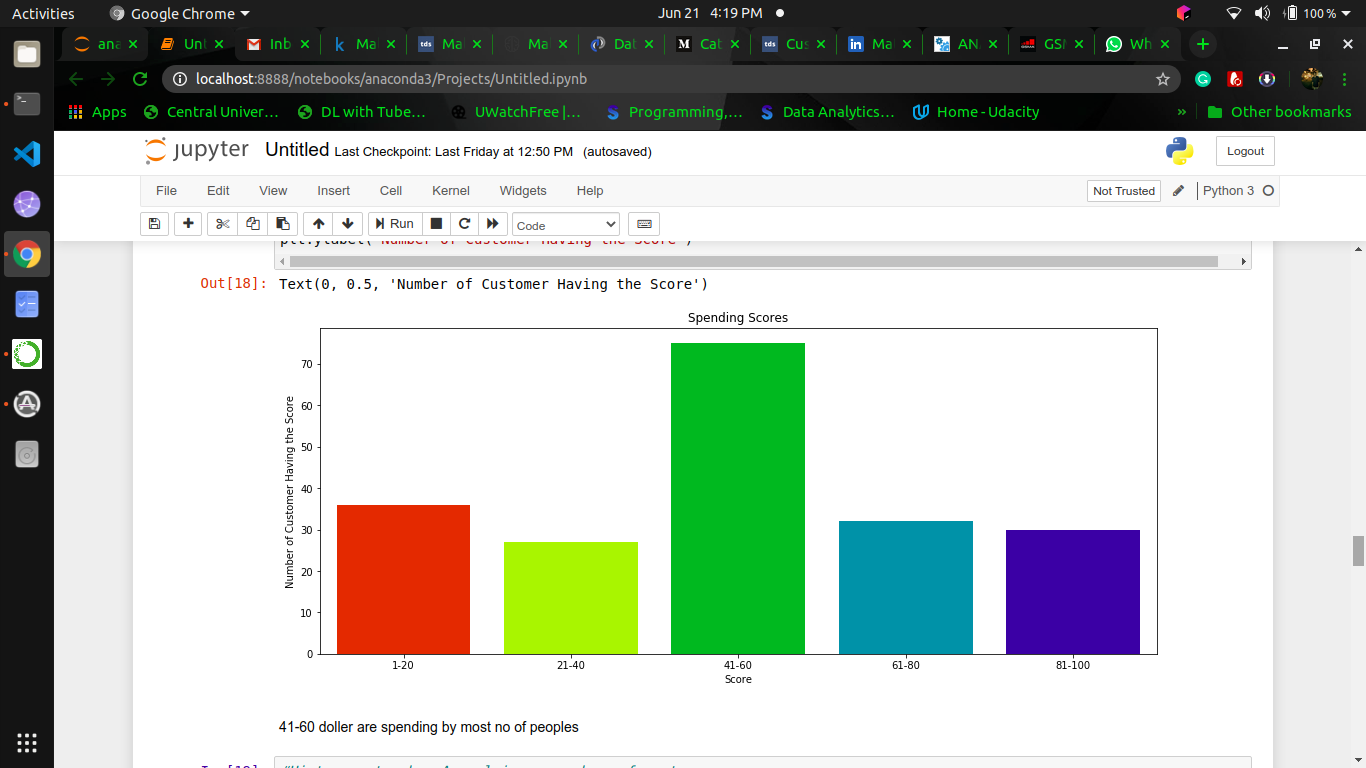
****

****

**Fig: Plotting Relation between Age and Number of Customer**

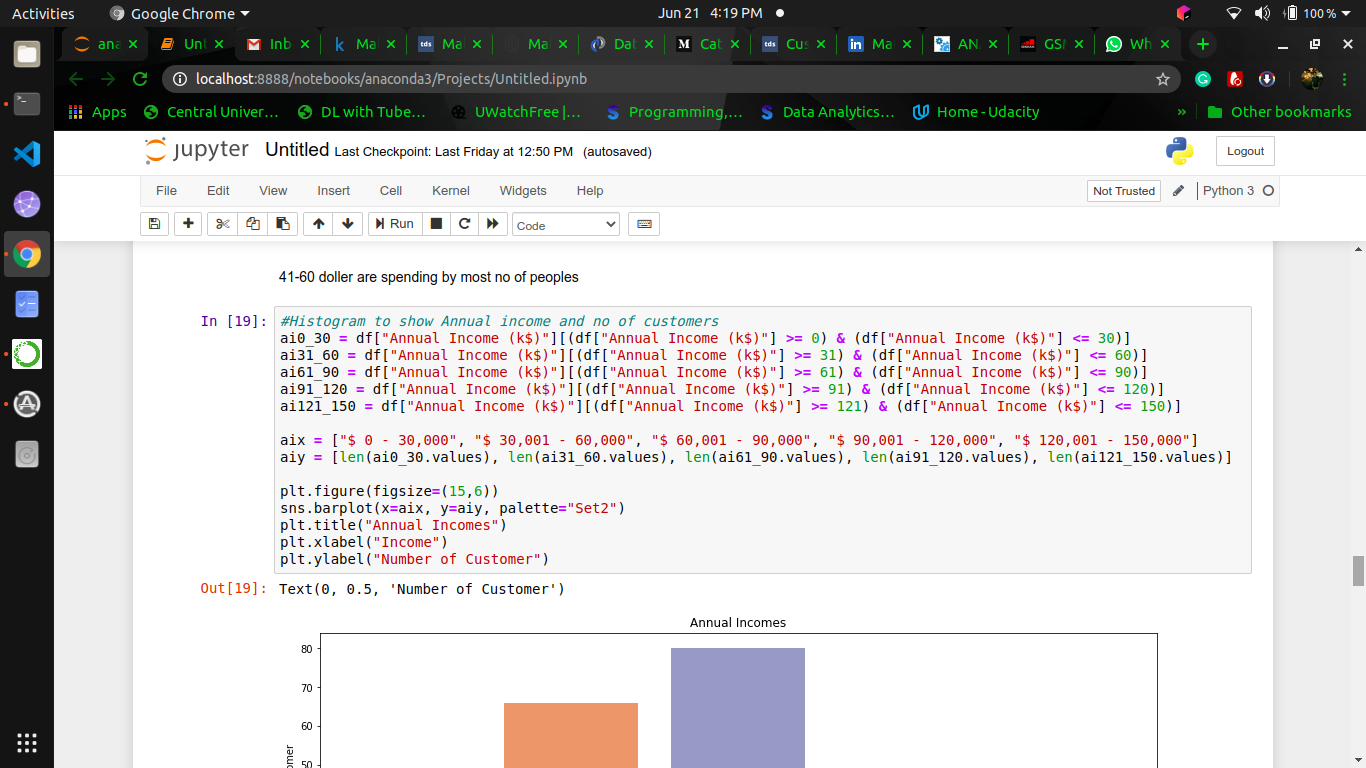
**Conclusion: 26-35 years old people are coming more to mall.**

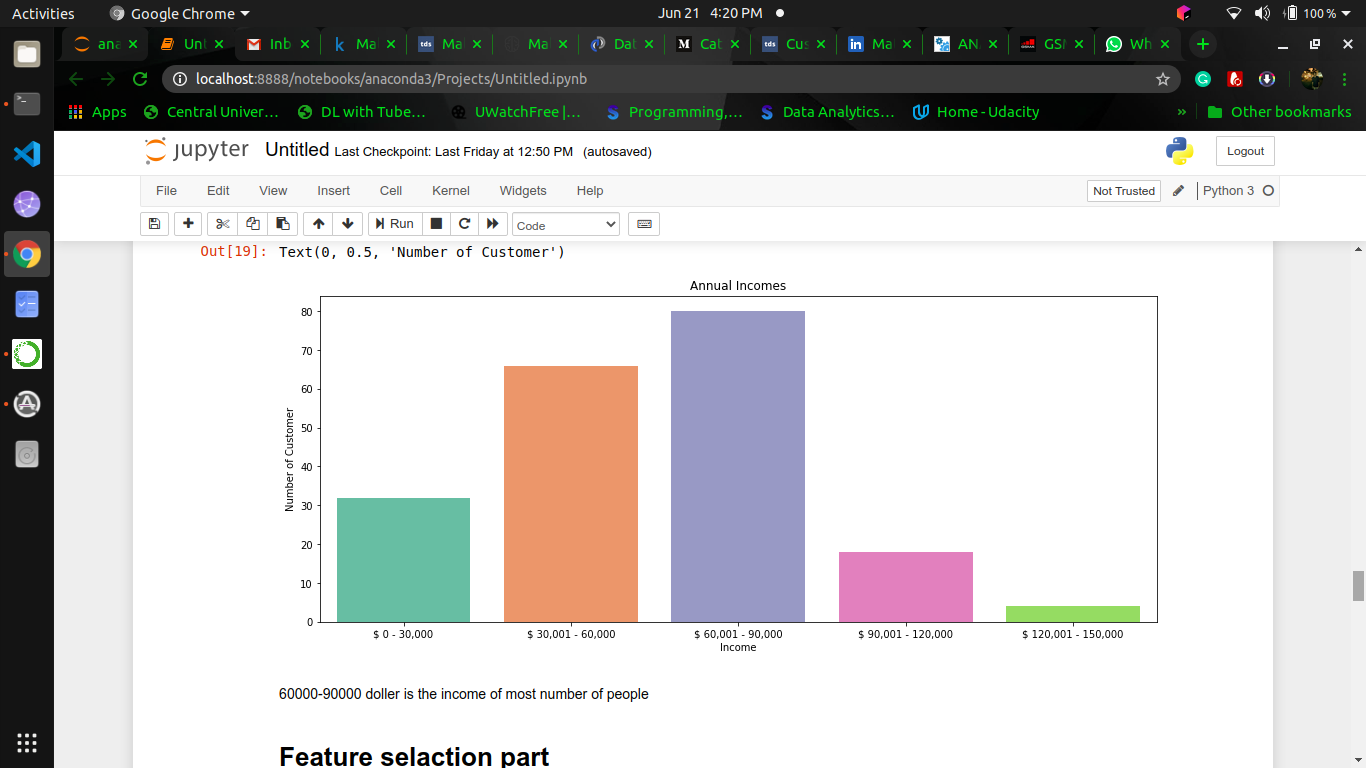
****



**Fig: Plotting Relation between Spending Score and Number of Customer**

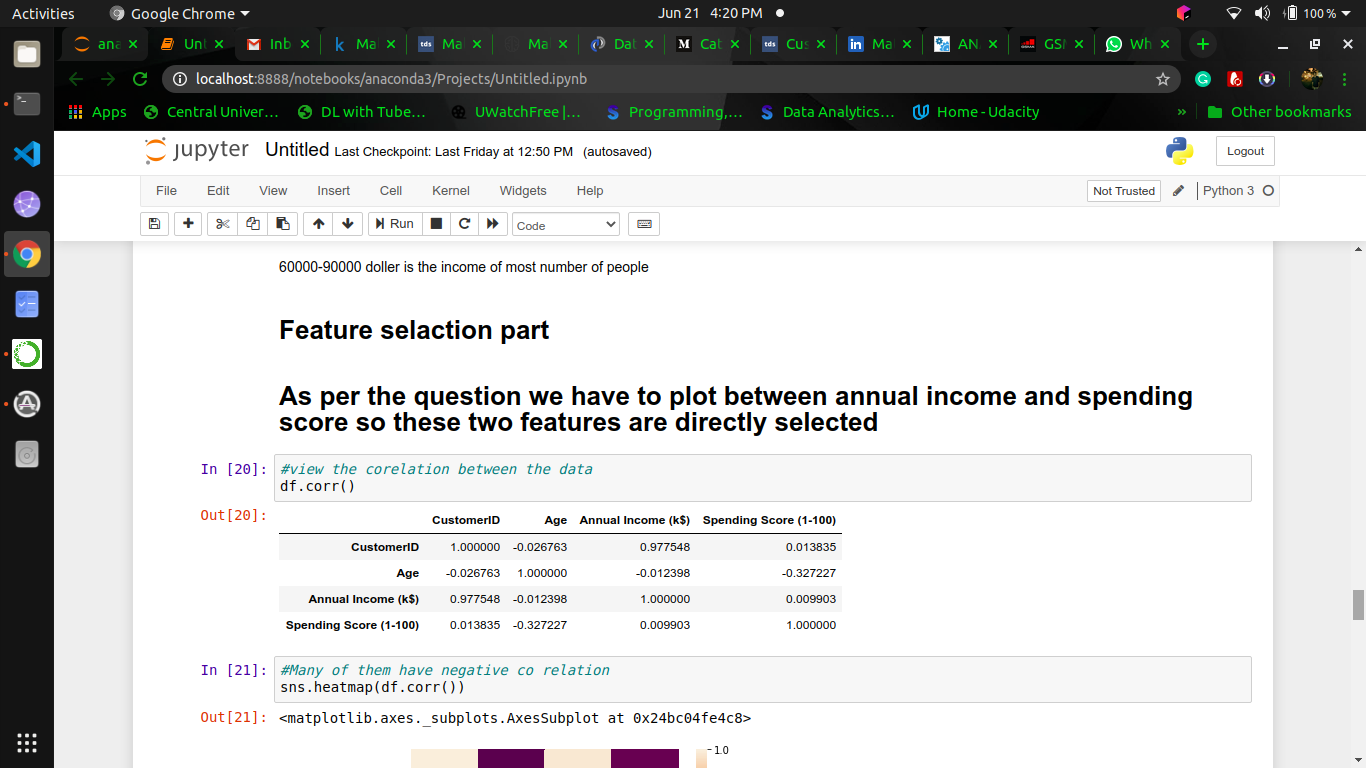
**Conclusion: 41-60 spending score corresponds to maximum number of customers.**

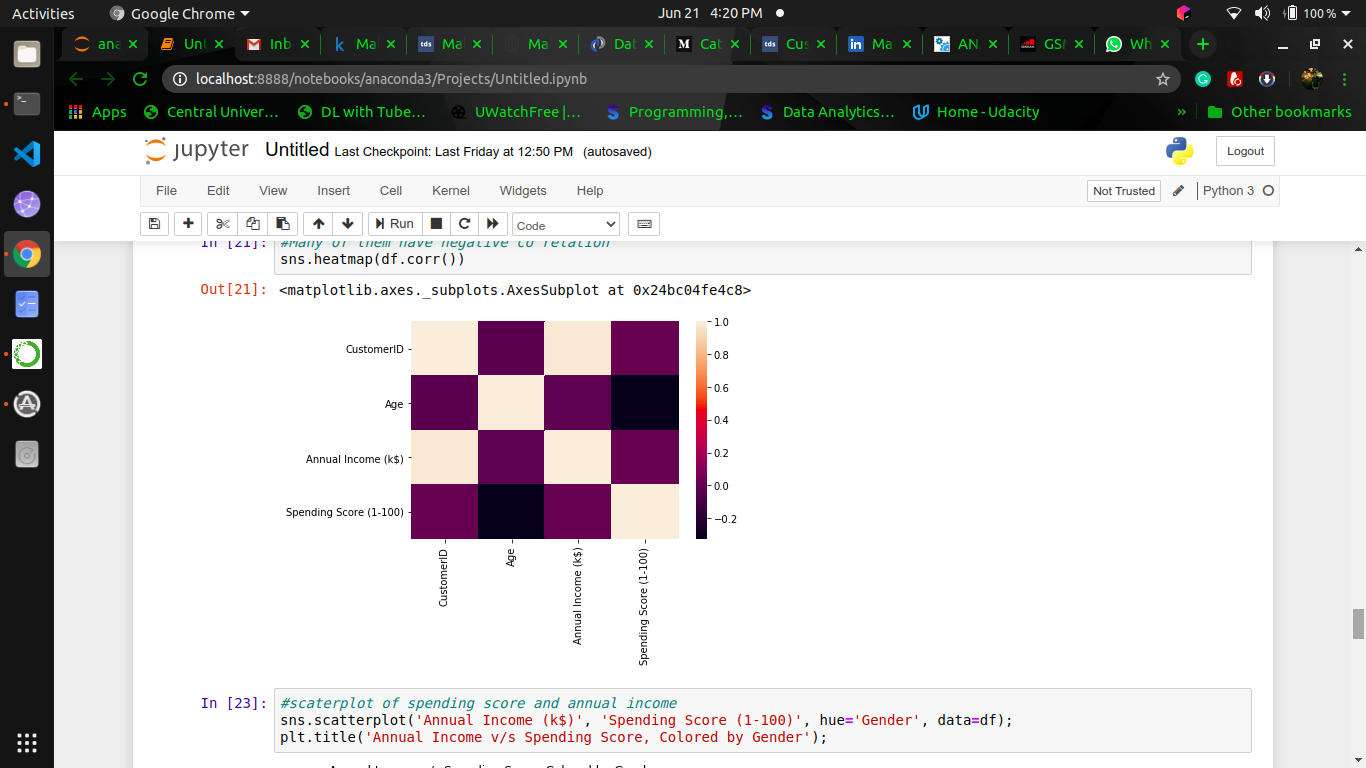
****

****

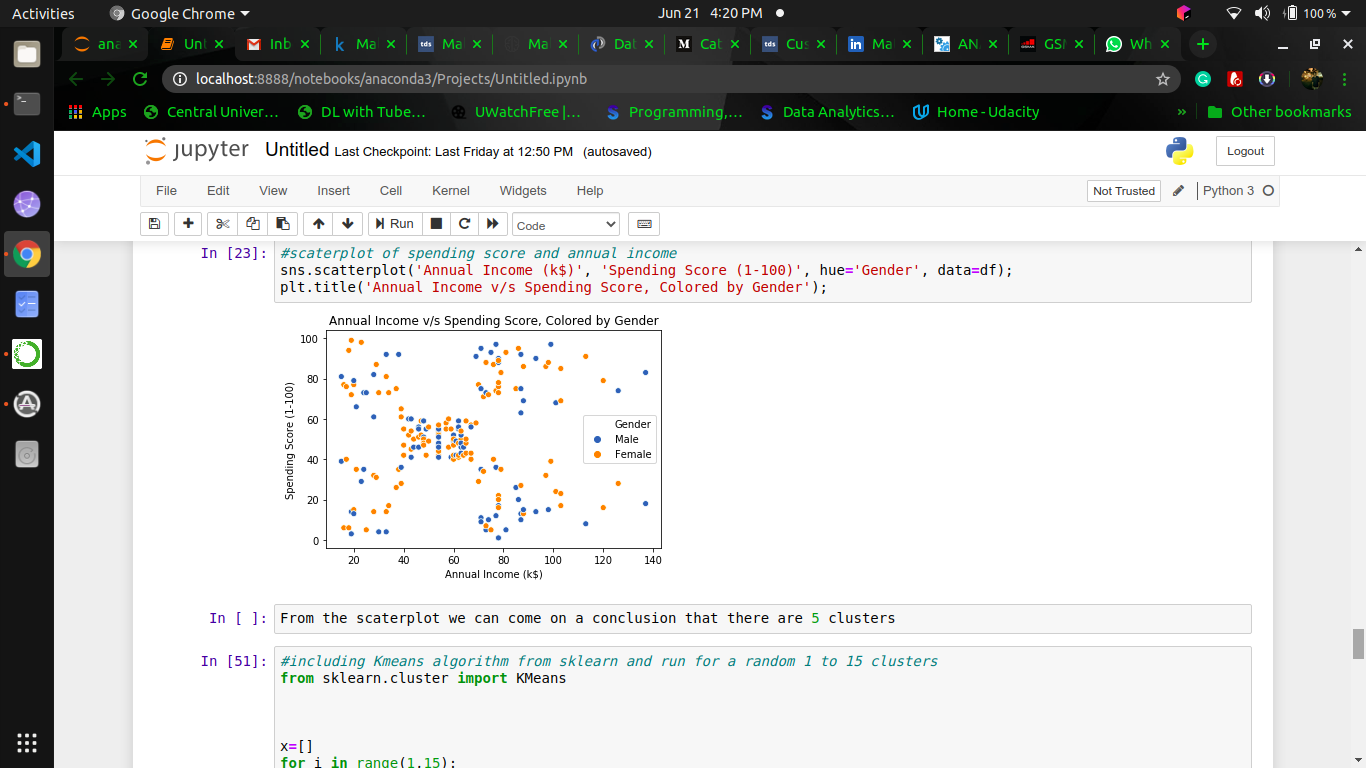
**Fig: Plotting Relation between Annual Income and Number of Customer**

**Conclusion: 60k-90k $ is the income of most of the people.**

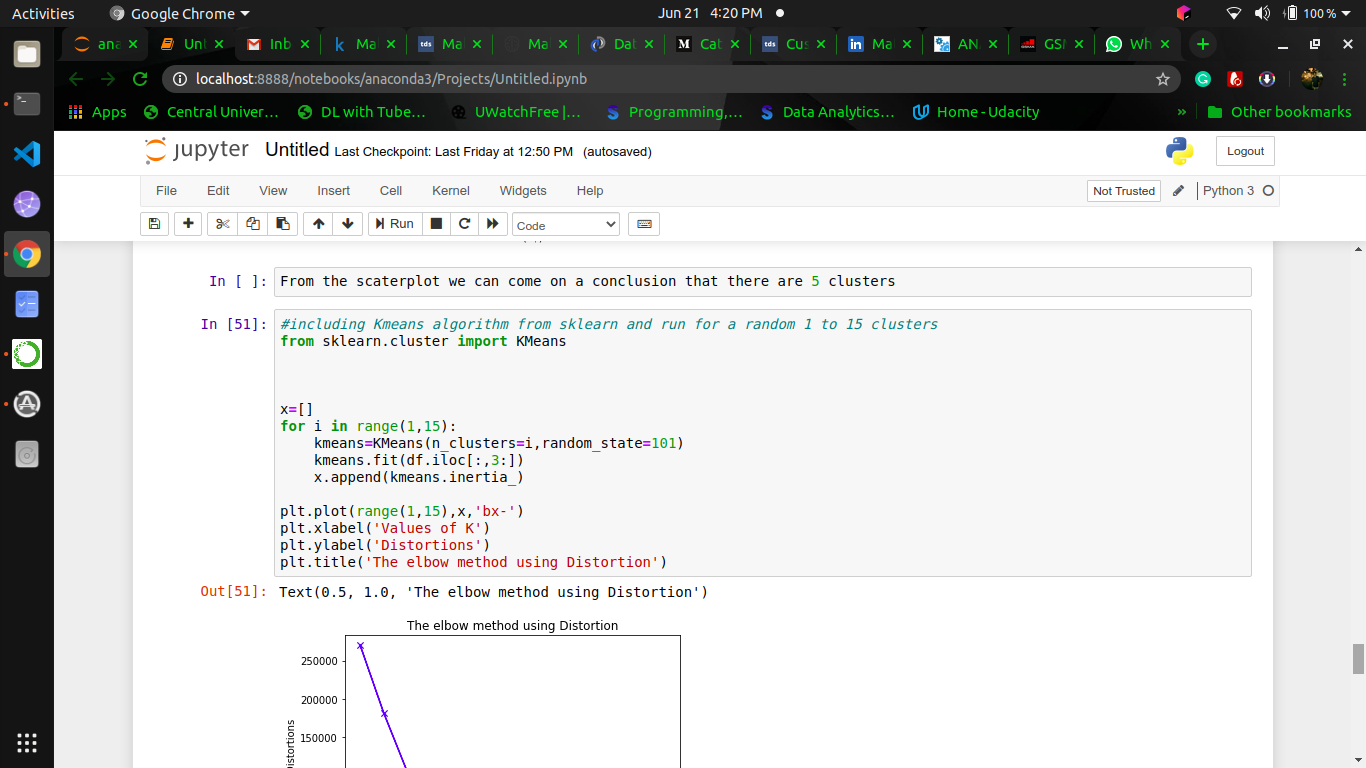
****

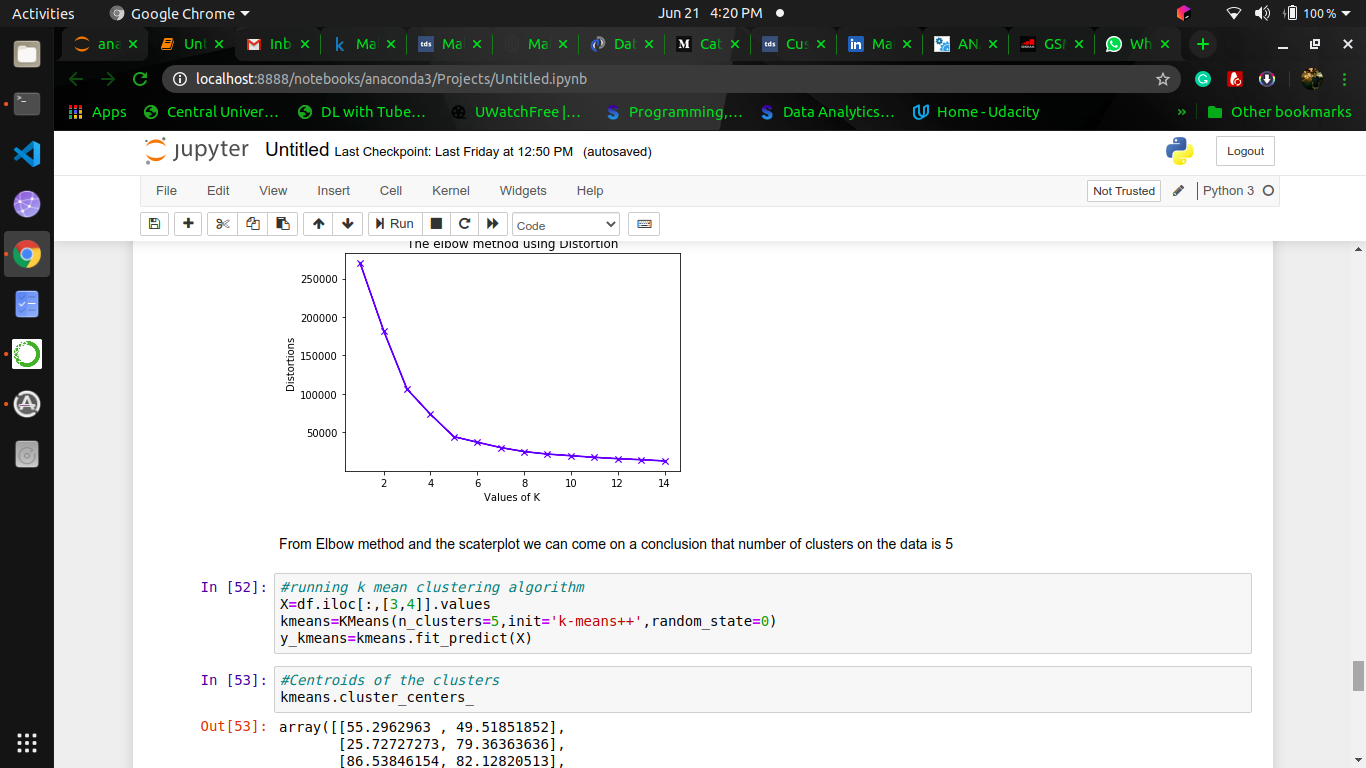
****

**Fig: Plotting correlation between data**

****

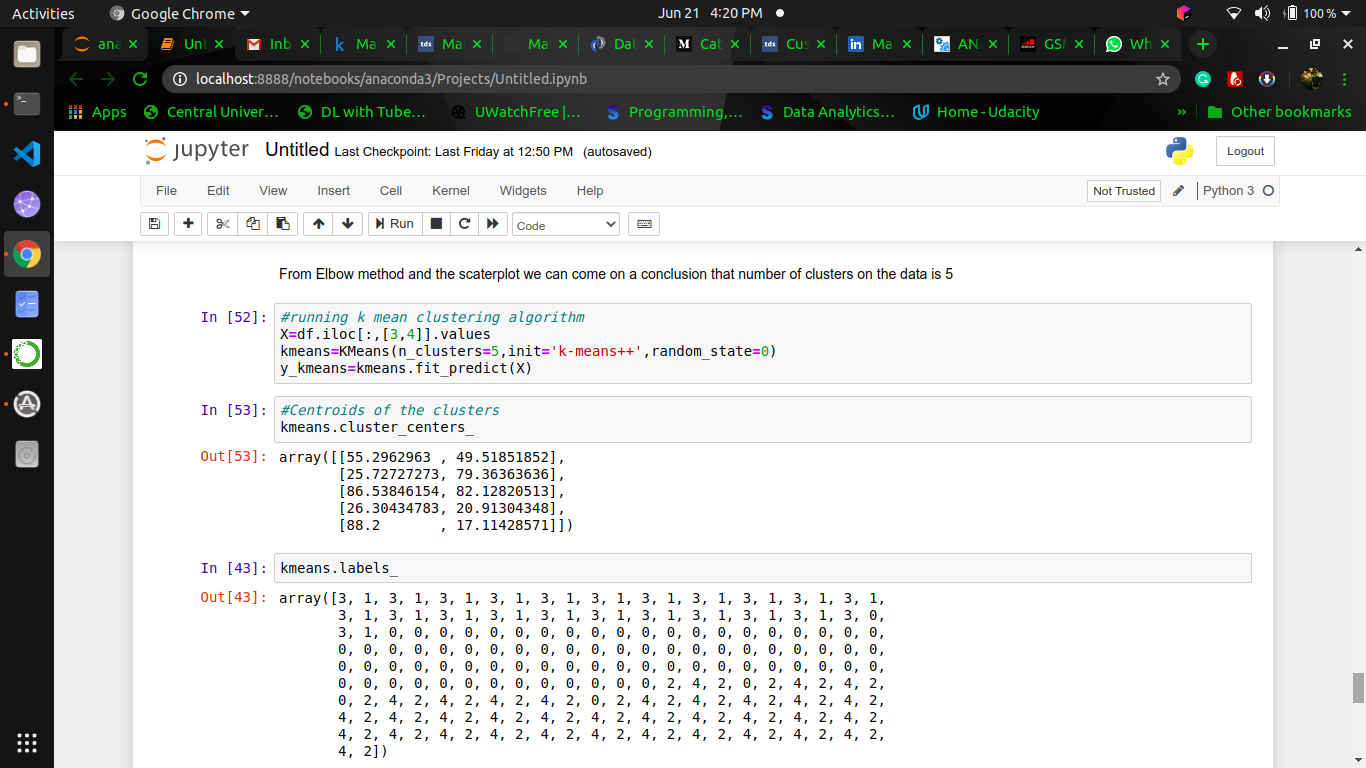
**Fig: Spending Score of Spending Score and Annual Income**

****

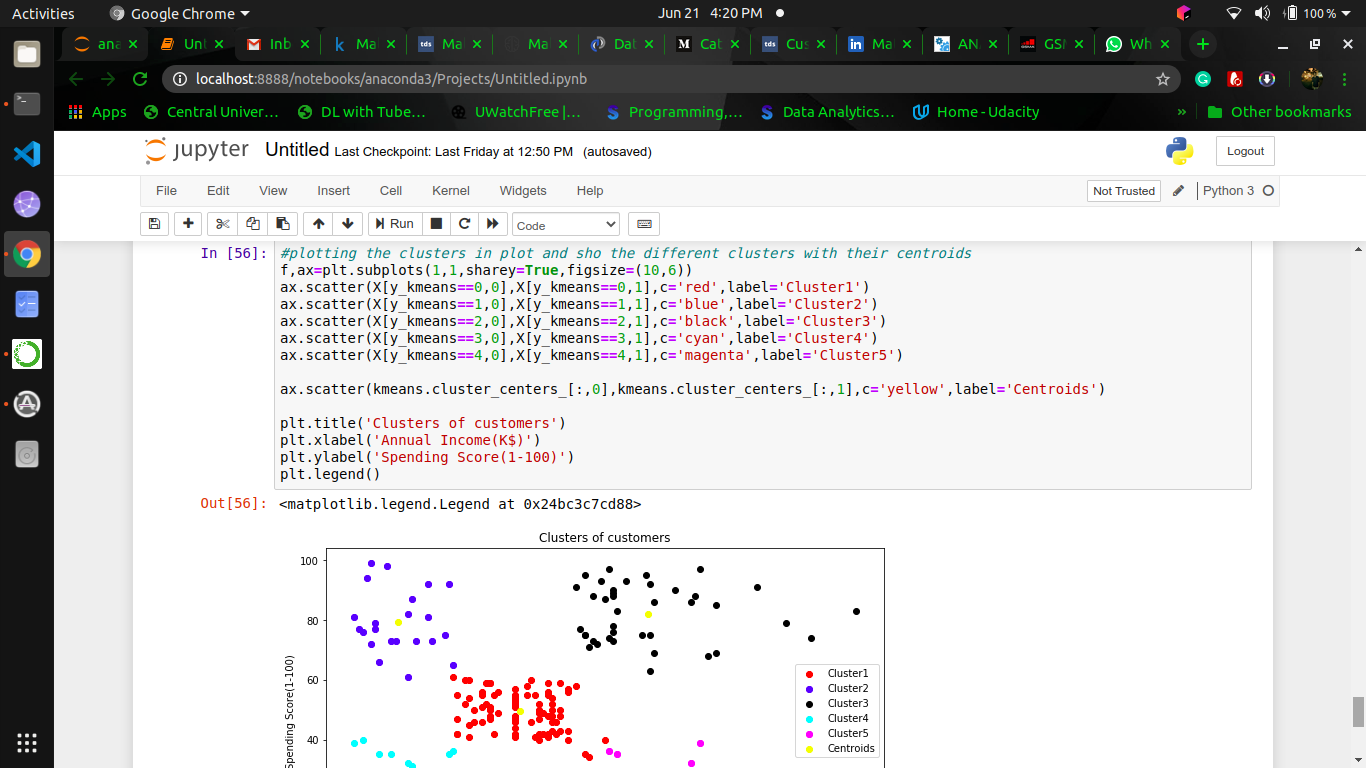


**Fig: Elbow Method**

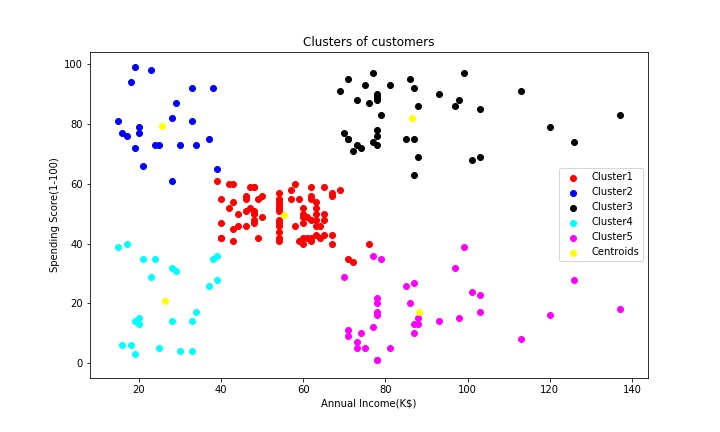
**Conclusion: 5 clusters in Data.**

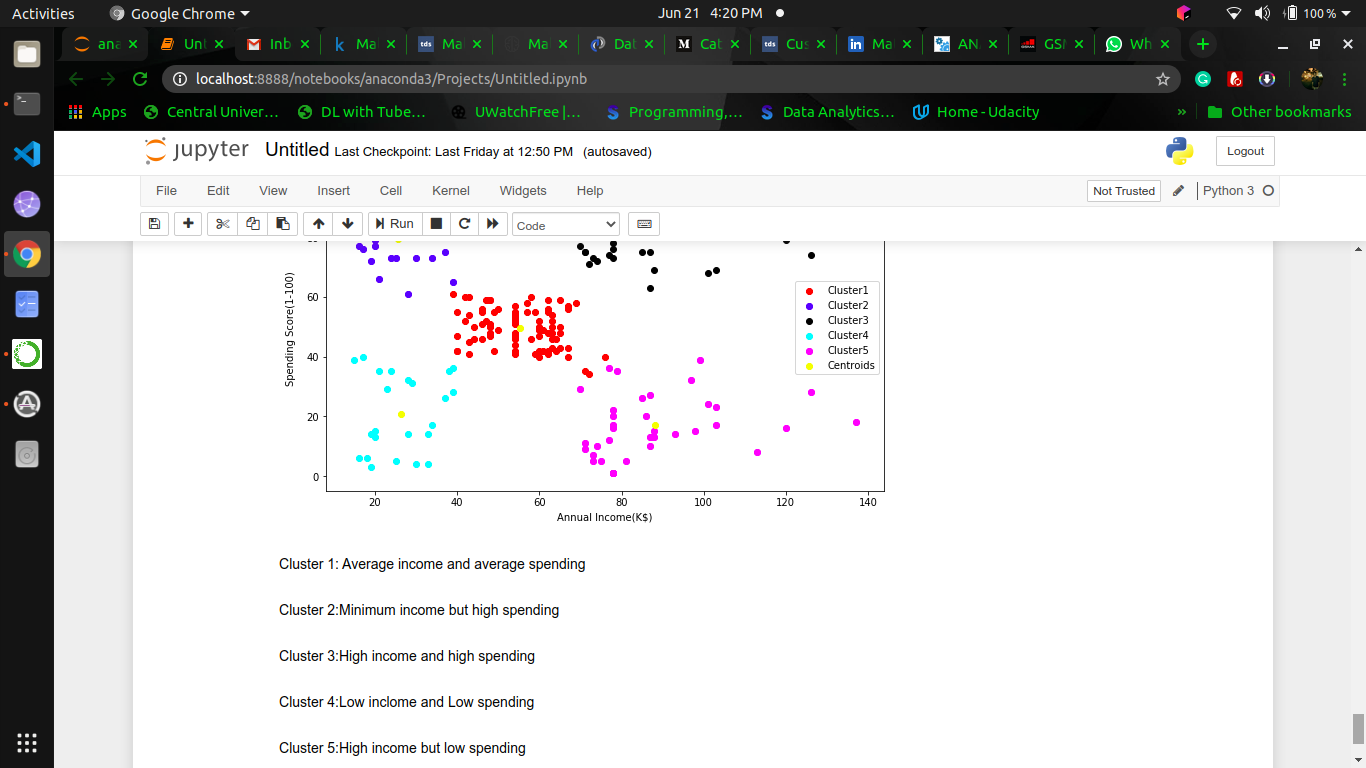


**Fig: K means Algorithm and Centroid of Clusters**



**Fig: Plotting Clusters**





**Fig: Plotting Clusters**

**Conclusion**

Customer Segmentation is a popular application of unsupervised learning. Using clustering, identify segments of customers to target the potential user base. They divide

customers into groups according to common characteristics like gender, age, interests, and spending habits so they can market to each group effectively.

We Used K-means clustering and also visualized the gender and age distributions. Then, we analysed their annual incomes and spending scores.

Language Used: Python

Dataset/Package: https://drive.google.com/file/d/19BOhwz52NUY3dg8XErVYglctpr5sjTy4/view