Credit card segmentation

REPORT

Himanshu behl | Edwisor |21/02/2020

Chapter

Introduction

In this project it requires to develop a customer segmentation to define marketing strategy through machine learning technique. The sample dataset summarizes the usage behavior of about 9000 active credit card holders during the last 6 months. The file is at a customer level with 18 behavioral variables.

PROBLEMS NEED TO ADDRESS:

Advanced data preparation: Build an 'enriched' customer profile by deriving "intelligent" KPIs such as:

- Monthly average purchase and cash advance amount
- Purchases by type (one-off, installments)
- Average amount per purchase and cash advance transaction,
- Limit usage (balance to credit limit ratio),
- Payments to minimum payments ratio etc.

ADVANCED REPORTING:

- Use the derived KPIs to gain insight on the customer profiles.
- Identification of the relationships/ affinities between services.
- Clustering: Apply a data reduction technique factor analysis for variable reduction technique and a clustering algorithm to reveal the behavioral segments of credit card holders
- Identify cluster characteristics of the cluster using detailed profiling.
- Provide the strategic insights and implementation of strategies for given set of cluster characteristics

DATA DICTIONARY:

- CUST_ID: Credit card holder ID
- BALANCE: Monthly average balance (based on daily balance averages)
- BALANCE_FREQUENCY: Ratio of last 12 months with balance
- PURCHASES: Total purchase amount spent during last 12 months

- ONEOFF_PURCHASES: Total amount of one-off purchases
- INSTALLMENTS_PURCHASES: Total amount of installment purchases
- CASH_ADVANCE: Total cash-advance amount
- PURCHASES_ FREQUENCY: Frequency of purchases (Percent of months with at least one purchase)
- ONEOFF_PURCHASES_FREQUENCY: Frequency of one-off-purchases
- PURCHASES_INSTALLMENTS_FREQUENCY: Frequency of installment purchases
- CASH_ADVANCE_ FREQUENCY: Cash-Advance frequency
- AVERAGE_PURCHASE_TRX: Average amount per purchase transaction
- CASH_ADVANCE_TRX: Average amount per cash-advance transaction
- PURCHASES_TRX: Average amount per purchase transaction
- CREDIT_LIMIT: Credit limit
- PAYMENTS: Total payments (due amount paid by the customer to decrease their statement balance) in the period
- MINIMUM_PAYMENTS: Total minimum payments due in the period.
- PRC_FULL_PAYMEN: Percentage of months with full payment of the due statement balance
- TENURE: Number of months as a customer

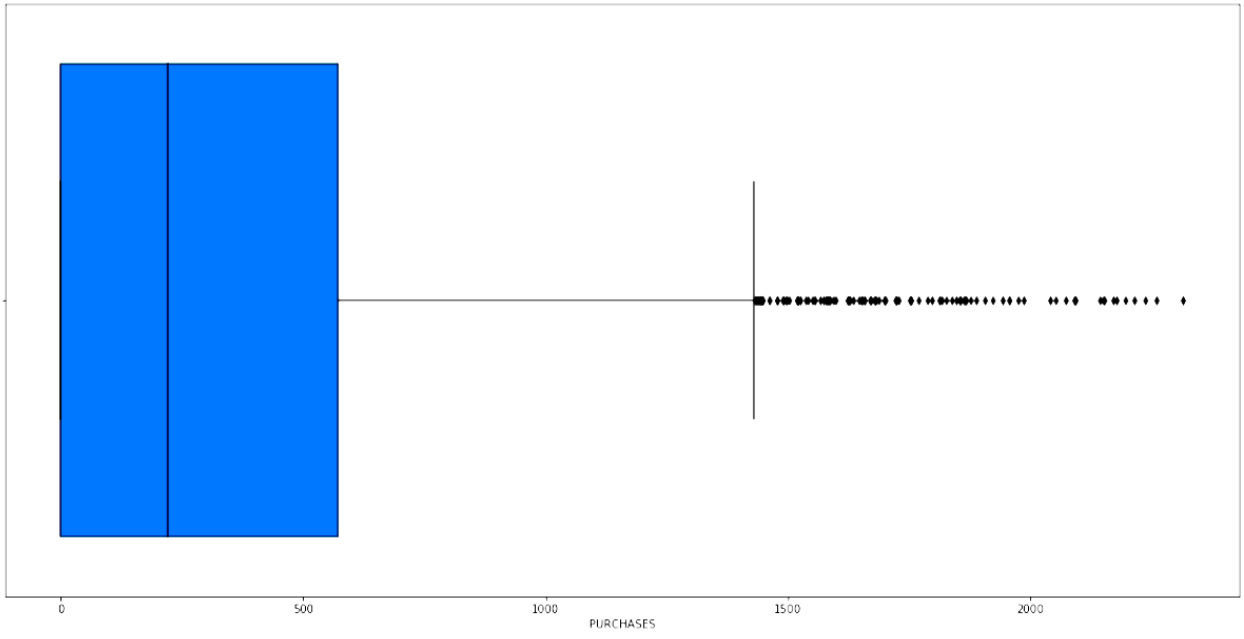
PRE-PROCESSING

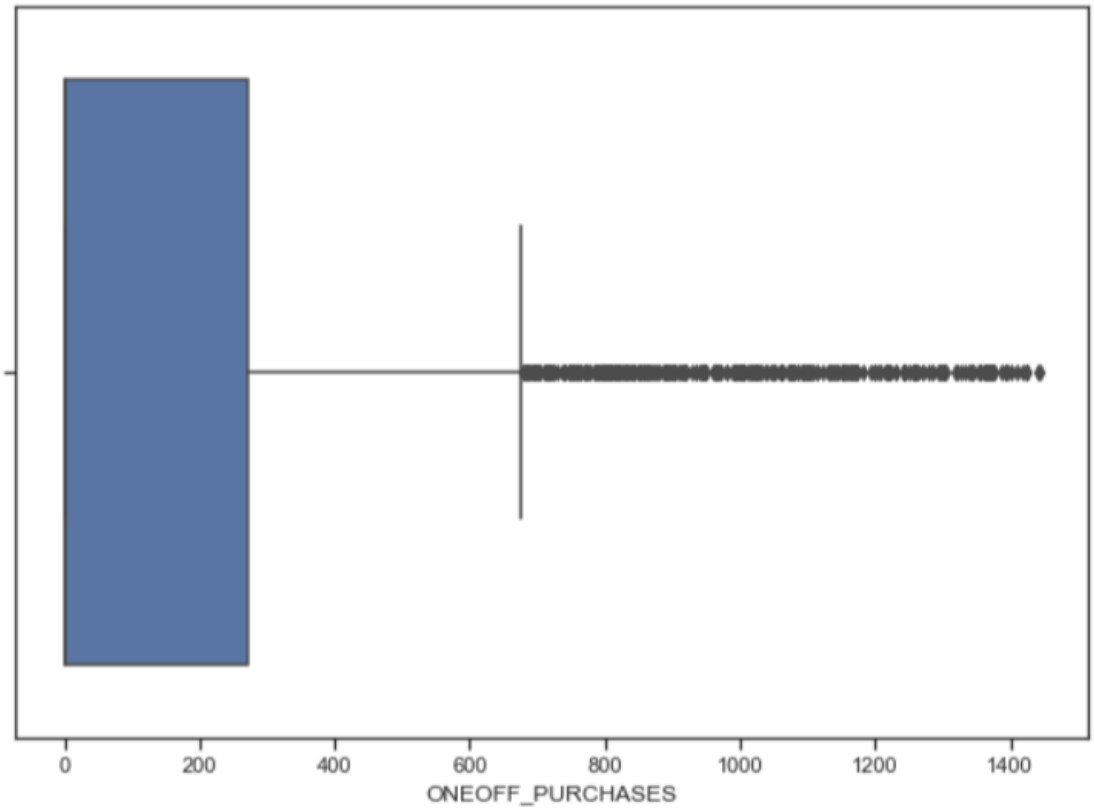
To predict the model, firstly we look the train dataset and manipulate the data before start the modelling. after that cleaning the data as well as visualizing the data through graph and plots. In the other words we explore the data and the data analysis. To data analysis we do following steps:

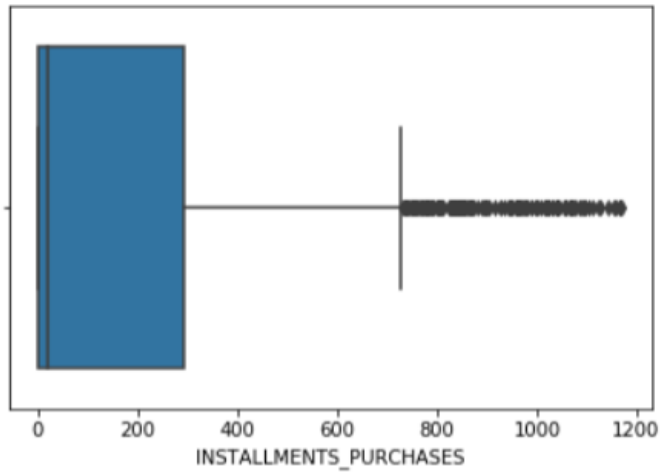
- Data exploration and Cleaning
- Missing values treatment
- Outlier Analysis
- Feature Selection and Scaling
- Visualization

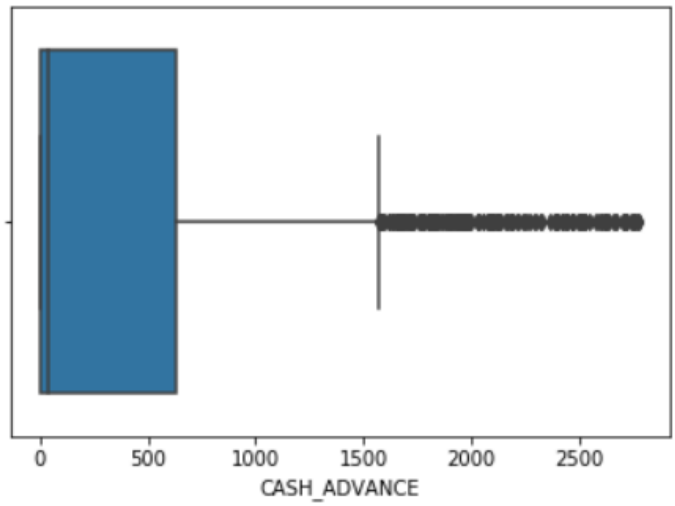
VISUALIZATION

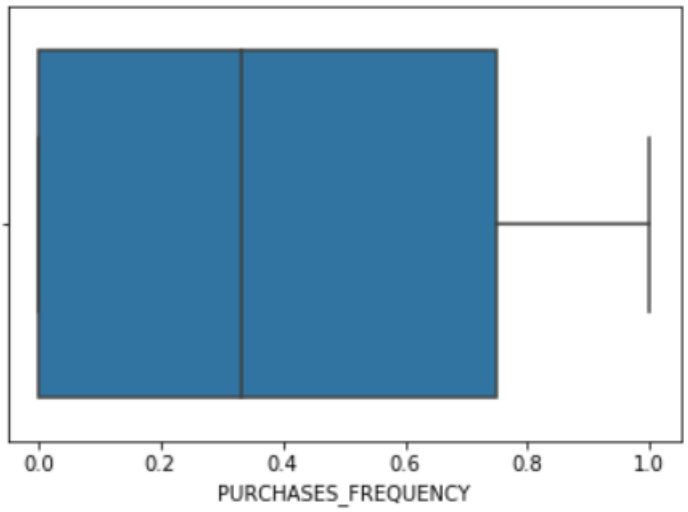
Box plots are a graphical depiction of numerical data through their quantiles. It is a very simple but effective **way to visualize outliers**. Think about the lower and upper whiskers as the boundaries of the data distribution. Any data points that show above or below the whiskers, can be considered **outliers** or anomalous.

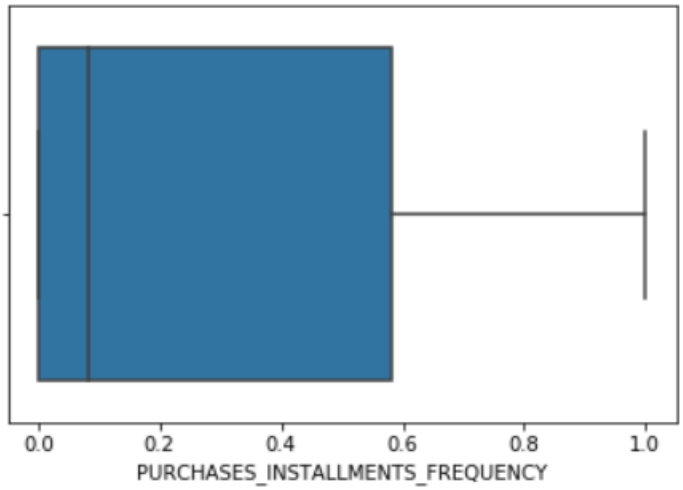


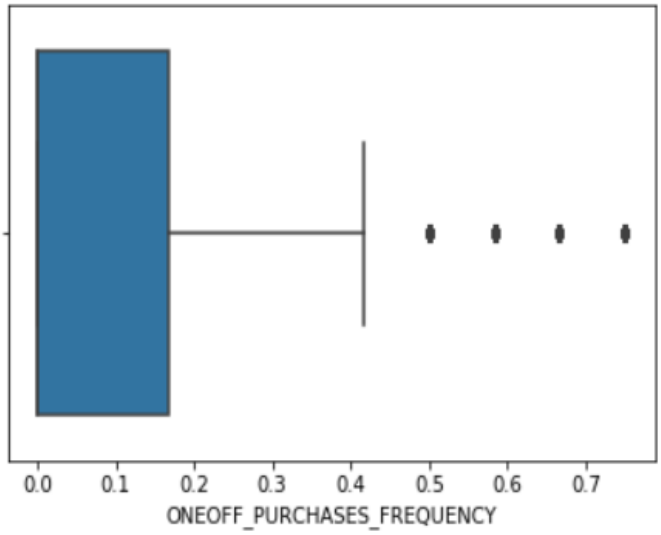

0
200
400
600
800
1000
1200
1400
ONEOFF_PURCHASES

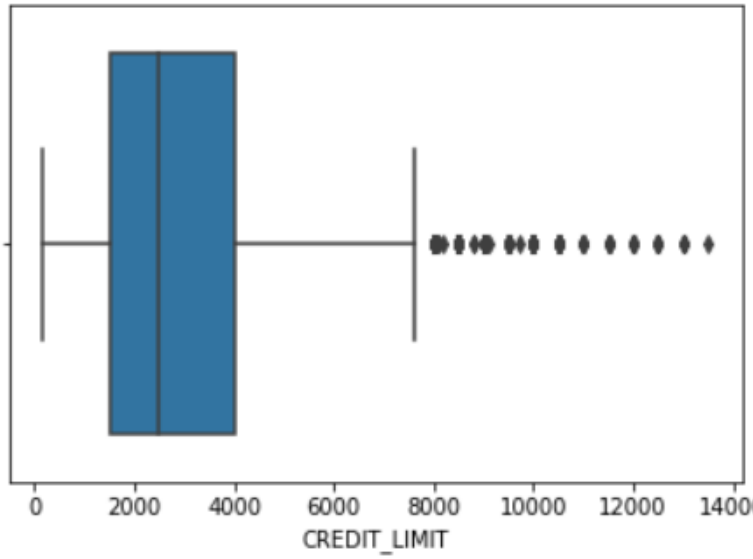

0
200
400
600
800
1000
1200
INSTALLMENTS_PURCHASES

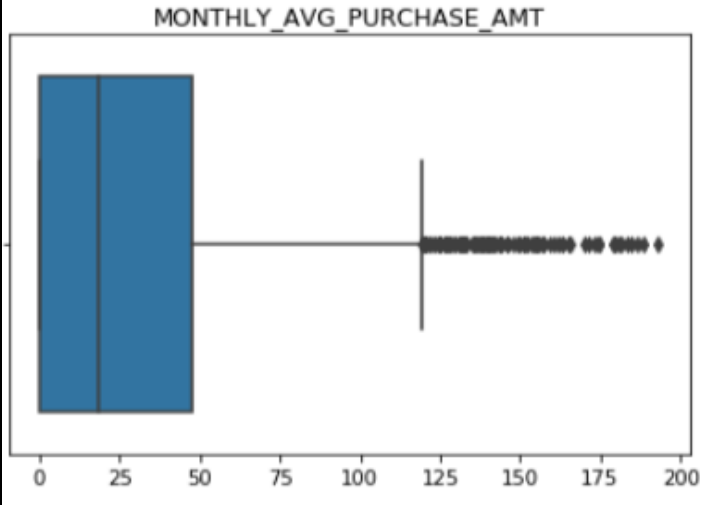

0
500
1000
1500
2000
2500
CASH_ADVANCE

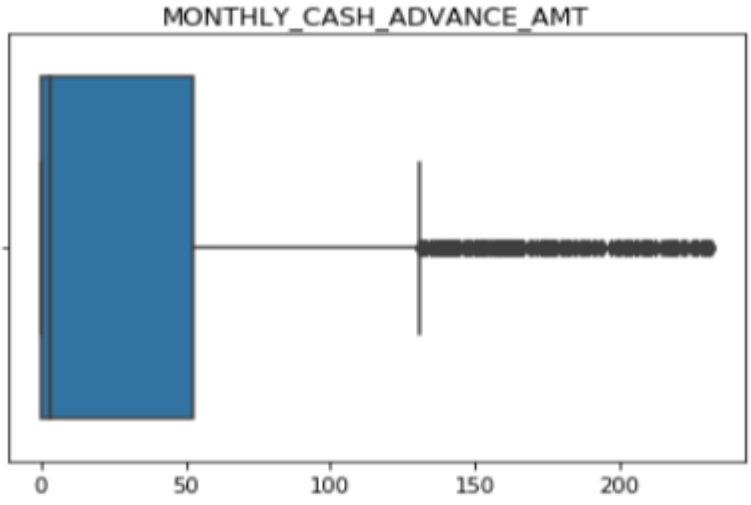

0.0
0.2
0.4
0.6
0.8
1.0
PURCHASES_FREQUENCY


0.0
0.2
0.4
0.6
0.8
1.0
PURCHASES_INSTALLMENTS_FREQUENCY


0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
ONEOFF_PURCHASES_FREQUENCY


0
2000
4000
6000
8000
10000
12000
14000
CREDIT_LIMIT


MONTHLY_AVG_PURCHASE_AMT
0
25
50
75
100
125
150
175
200


MONTHLY_CASH_ADVANCE_AMT
0
50
100
150
200

A **heatmap** is a graphical representation of data in which data values are represented as colors. That is, it uses color in order to communicate a value to the reader. This is a great tool to assist the audience towards the areas that matter the most when you have a large volume of data

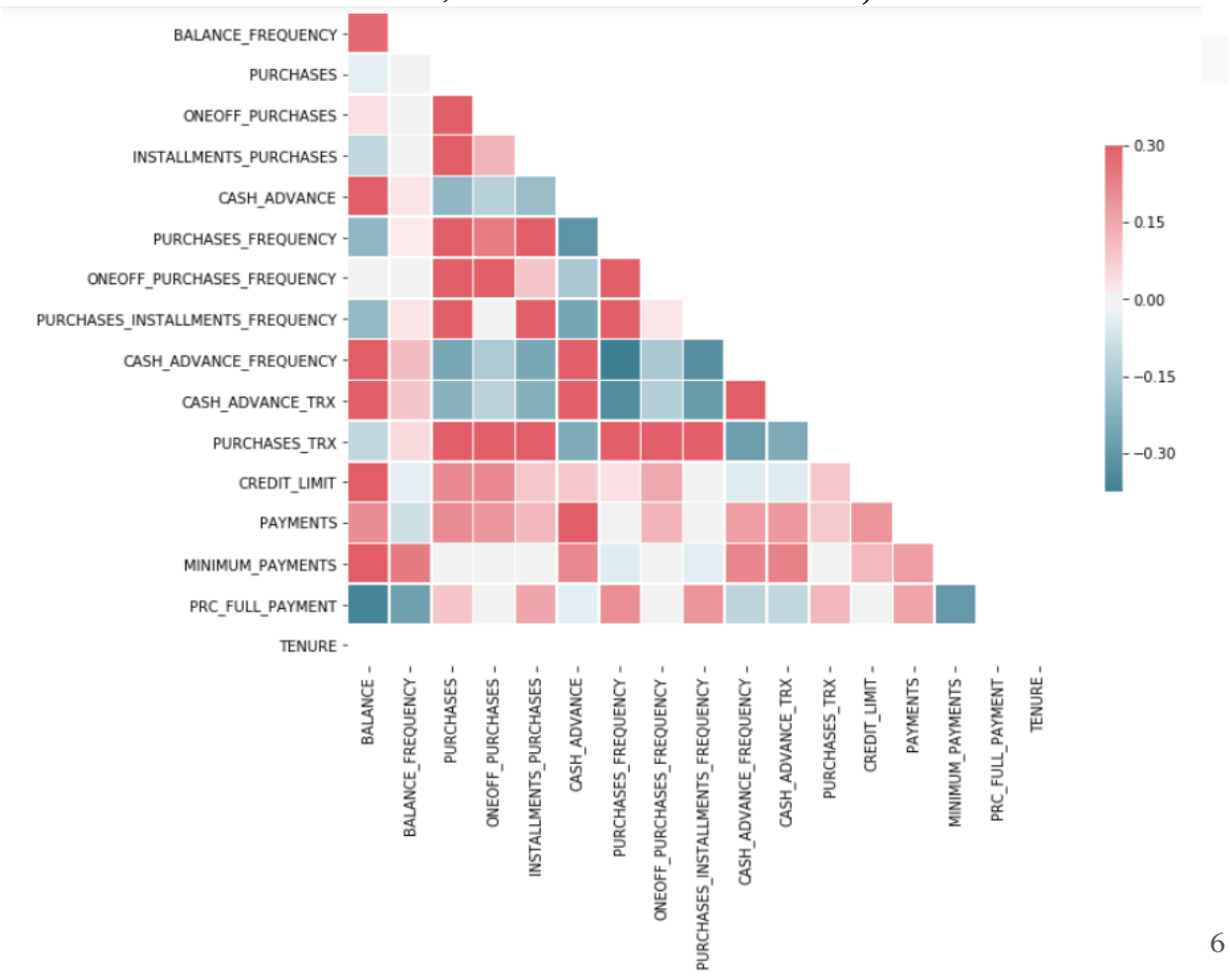
A correlation is a statistical measure of the relationship between two variables. The measure is best used in variables that demonstrate a linear relationship between each other. The fit of the data can be visually represented in a scatterplot. Using a scatterplot, we can generally assess the relationship between the variables and determine whether they are correlated or not.

The correlation coefficient is a value that indicates the strength of the relationship between variables. The coefficient can take any values from -1 to 1. The interpretations of the values are:

-1: Perfect negative correlation. The variables tend to move in opposite directions (i.e., when one variable increases, the other variable decreases).

0: No correlation. The variables do not have a relationship with each other.

1: Perfect positive correlation. The variables tend to move in the same direction (i.e., when one variable increases, the other variable also increases).



KEY PERFORMANCE INDICATOR (KPIs)

A Key Performance Indicator is a measurable value that demonstrates how effectively a company is achieving key business objective. Organizations use KPIs at multiple levels to evaluate their success at reaching targets. High level KPIs may focus on the overall performance of the business while low level KPIs may focus on process in department such as sales, marketing, HR, support and others.

A quantifiable measure used to evaluate the success of an organization, employee, etc. in meeting objectives for performance.

CLUSTERING

Clustering is one of the most common exploratory data analysis technique used to get an intuition about the structure of the data. It can be defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different. In other words, we try to find homogeneous subgroups within the data such that data points in each cluster are as similar as possible according to a similarity measure such as euclidean-based distance or correlation-based distance. The decision of which similarity measure to use is application-specific.

Kmeans ALGORITHM

Kmeans algorithm is an iterative algorithm that tries to partition the dataset into *K*pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to **only one group**. It tries to make the inter-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

The way kmeans algorithm works is as follows:

1. Specify number of clusters *K*.
2. Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement.

3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.

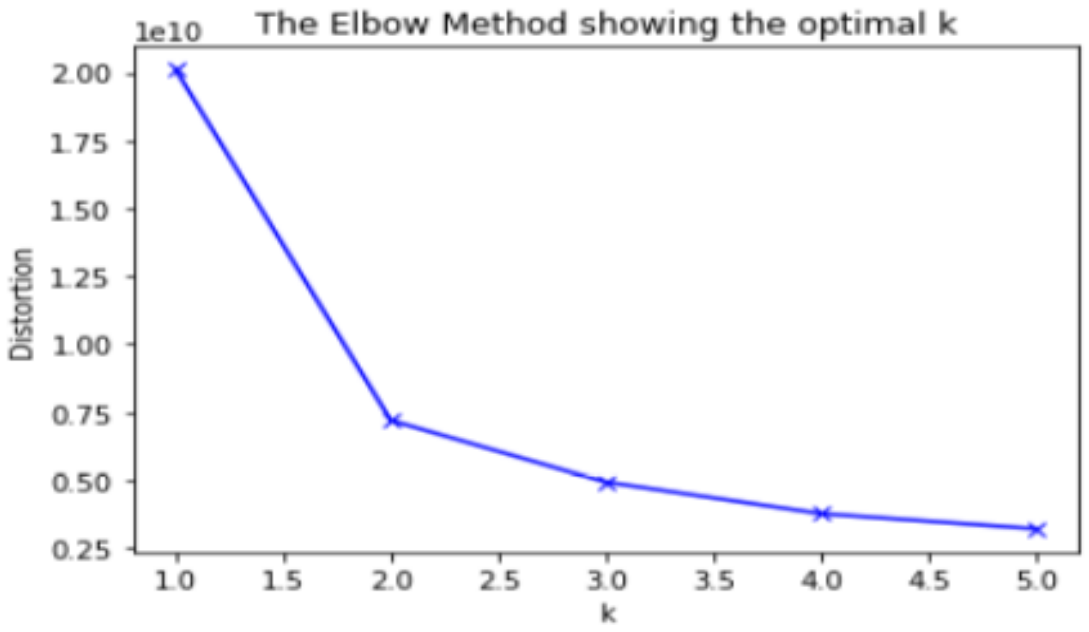
- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).
- Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

HOW MANY CLUSTERS?

I mentioned previously that we need to tell the K-Means algorithm the number of clusters it should use. There are a number of techniques that can be used to find the optimal number. For this example, I am going to use the elbow method so named because the chart that it produces is similar in shape to the curve of an elbow. This method computes the sum of squared distances for clusters k. As more clusters are used the variance will reduce until you reach a point at which increasing clusters no longer results in a better model.

CLUSTERING ANALYSIS¶

```
# Plot the elbow
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```



```
km = KMeans(init="random", n_clusters=4)
km.fit(cluster_df)
```

```
KMeans(algorithm='auto', copy_x=True, init='random', max_iter=300, n_clusters=4,
       n_init=10, n_jobs=None, precompute_distances='auto', random_state=None,
       tol=0.0001, verbose=0)
```

```
km=KMeans(n_clusters=4,random_state=112)
km.fit(cluster_df)
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=4, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=112, tol=0.0001, verbose=0)
```