**CUSTOMER CHURN ANALYSIS OF A CERTAIN BANK**

**Statistics for Data Science (UE19CS203)**

**ABSTRACT**

Data is all over the place and all over the world. Even if we look at the smallest places and parts, we can obtain a lot of data and collect and analyse it. As is quoted:

***“Data is a precious thing and will last longer than the systems themselves”***

* **Tim Berners-Lee***,*

*Inventor of the World Wide Web*

Information is what drives most of the things right now in 21st century. As someone once said, “*Information is the oil of the 21st century and analytics is the combustion engine*”. Hence our team set out to analyse the data of a certain banking organisation and evaluate its customer attrition rate or customer loss rate. This sort of predictive analysis model is termed as the ‘Churn Model’ or ‘Churn Analysis’. We further went on to draw meaningful inferences from our analysis. The data analysed was a sample provided by the organization for around 5000 customers. The data was analysed, cleaned and then graphs were visualised. This was followed the standardisation and normalization of the values and testing for null hypothesis. We checked for correlation, if any in the data and concluded our findings.

**INTRODUCTION**

The model we have analyses the data towards the end goal of reducing the customer loss rate/customer attrition rate. A churn analysis or churn model would certainly useful in almost all organisations looking for increased profit in any scenarios. So we took the data of this bank and checked for all inconsistencies and proceeded to clean them up before standardisation and/or normalisation of this data.

Our goal is to monitor and analyse the data that we have on our hands and check whether we can generalise our findings from the sample (of 5000 people) to the entire population (all the customers or members of the bank). Also we would be going ahead and checking for correlations found, if any, between two or more of the tuples in the dataset.

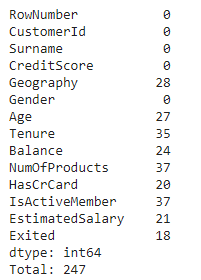
**DATASET**

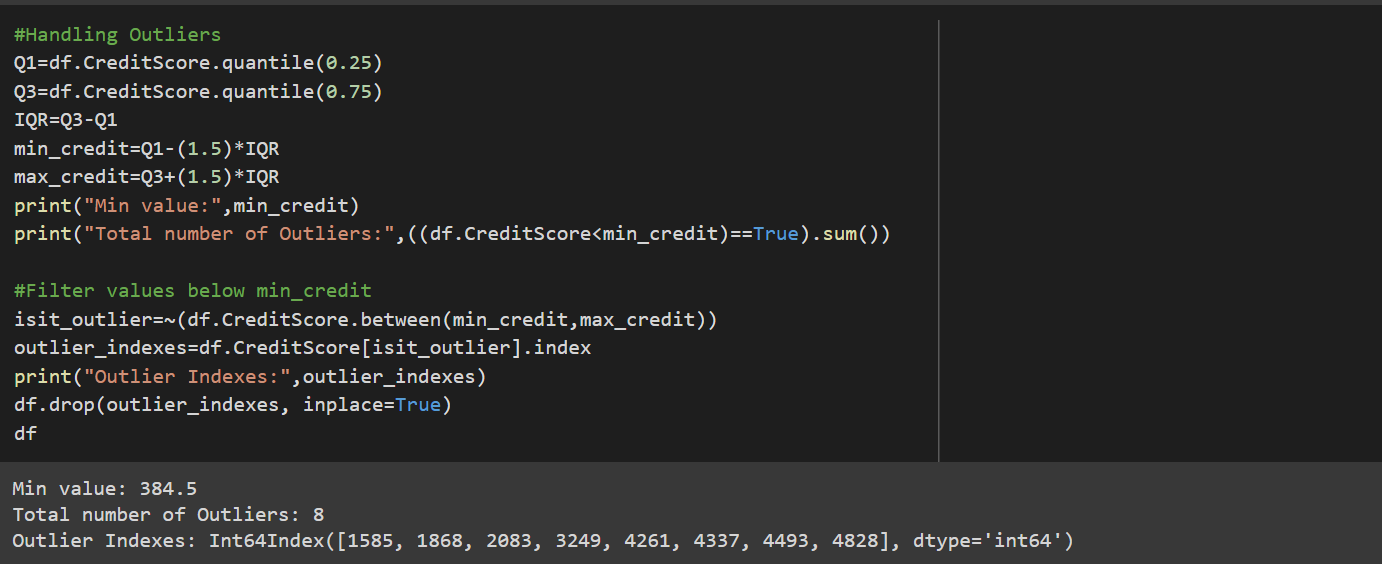
The dataset we will be working during the course of this project is that of a banking organisation. It is a churn model, the source of which can be found at Kaggle, It is in a .csv format and contains the data for around 5000 members in columns like RowNumber, CustomerID, Surname, CreditScore, Geography (or nationality of the user), Balance (in the account of the account holder), Tenure (maturity period of the account) IsActiveMember (tells if the account is active or not, 1 signifies true and 0 signifies false). Then we have a similar column in HasCrCard, specifying if the account holder possesses a credit card or not and Exited, which states whether the account is terminated or not. All three of these columns hold binary values for true or false.

**PREPROCESSING OR DATA CLEANING**

Data cleaning is the process of preparing data for analysis by removing or modifying data that is incorrect, incomplete, irrelevant, duplicated, or improperly formatted. Data cleaning is not simply about erasing information to make space for new data, but rather finding a way to maximize a data set’s accuracy without necessarily deleting information. There are several methods for cleaning data depending on how it is stored along with the answers being sought.

This portion required us to follow a set of steps in a certain order. This order is listed below:

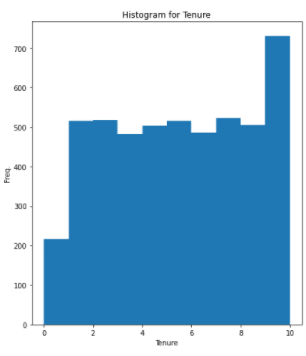
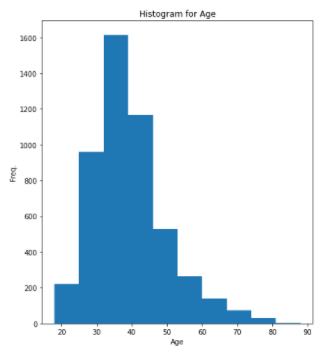
1. Missing values: these were initialised to a list containing the values - ["na","n/a","--","NaN",""," "]. The total number of missing values were found using isnull().sum() and these missing or NaN values were handled by replacing them with appropriate values using the fillna() function.
2. Inconsistent data: We also obtained a lot of inconsistencies in our data like the repetition of countries in different cases for different rows, for example France, was mentioned as france, fRance, and France at three different rows. The issue of inconsistencies in the form of capitalization was tackled using the str.lower() function which makes everything lowercase.
3. Repetitive Data: From the dataset, we observed that the last few rows had been duplicated, i.e., repeated many times. Since Duplicate data could negatively affect analysis of the data at hand, so we use the drop\_duplicates() function to eliminate all duplicate values.
4. Outliers: Outliers are handled by first using the quantile() function to calculate minimum and maximum credit values, following which all values below the minimum credit and above the maximum credit are filtered out using drop() function.

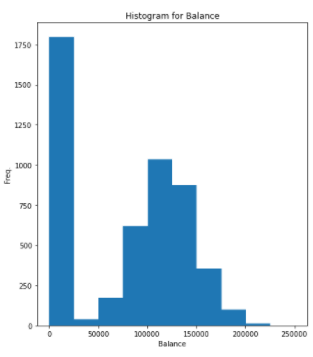


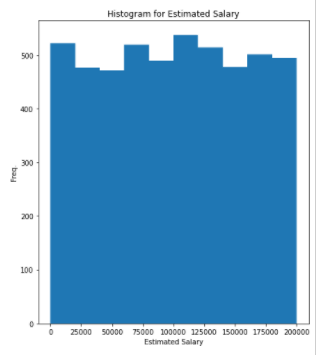
**EXPLORATORY DATA ANALYSIS**

After cleaning the data, there was still much left to do. The detailed analysis and standardisation of the data was what followed.

Graphs:

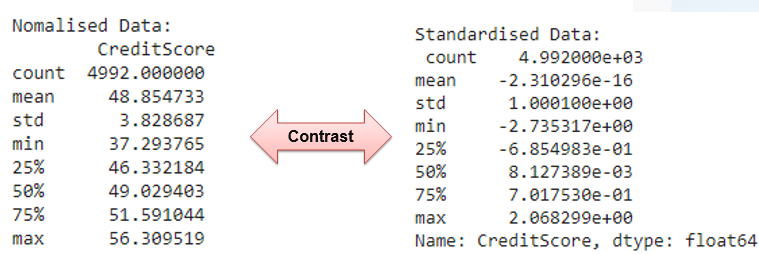
****

****

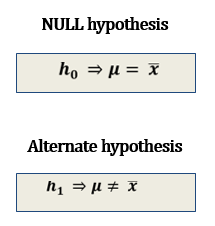
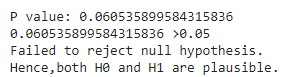
****

Normalization: Box-cox transformation is utilized to transform the non-normal distribution of our data to a normal distribution which allows us to run a lot more tests. The boxcox() function is imported from the scipy module. Box-cox does not guarantee normality because it never checks for the normality which is necessary to be foolproof that it has correctly transformed the non-normal distribution or not. It only checks for the smallest standard deviation. So, a normality check is performed with the help of norm.ppf().

Standardization: Standardization is about making sure that data is internally consistent, that is each data type has the same content and format. Standardised values are useful for tracking data that isn't easy to compare otherwise. StandardScaler() function is used which follows Standard Normal Distribution (SND). It takes mean = 0 and scales the data to unit variance. Then, the fit\_transform() function is used to scale the data and learn the scaling parameters. Finally, the descriptive statistics of both the normalised and standardised data are viewed using the describe() function.

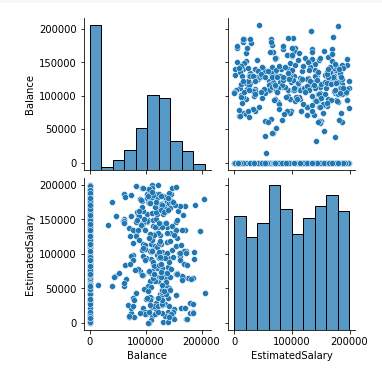


**HYPOTHESIS TESTING**

 In inferential statistics, the null hypothesis is a general statement that there is no difference between two measured phenomena or two samples that derive from the same general population. Testing the null hypothesis tells that there are (or there are not) grounds for believing that there is a relationship between the two phenomena. An alternate hypothesis is often provided to dispute the null hypothesis. It is generally assumed to be true until evidence states otherwise. The null hypothesis, h0, here proposes that mean estimated salary of the sample, can be generalised as the mean estimated salary of the population i.e., the mean estimated salary of all the customers of the bank, denoted by ‘μ’. Whereas the alternative hypothesis, h1, proposes that there is there is no relation between these parameters. But, the code on being run displayed the p-value being greater than α, which meant that we cannot reject the null hypothesis in this case. We can also infer that population parameter, μ, might or might not be equivalent to the sample parameter.

**CORRELATION**

After obtaining a lot of data, we started to check whether there is any sort of relation to be found among different columns of the dataset. Specifically, we looked to see if the ‘Balance’ in an account can tell something about the measure of ‘EstimatedSalary’ of the account holder. For this purpose, we used pairplots and unfortunately, there was none or neutral correlation between the two.



**RESULTS AND INFERENCE**

We concluded our findings with saying that the null hypothesis could not be rejected, which meant that both null and alternate hypothesis could be true. From checking for the correlation between the columns ‘Balance’ and ‘EstimatedSalary’ we can also safely say that the balance one has in their bank account is not necessarily a measure of how much they earn (or their estimated salary for that matter).

