**Santander Customer Transaction Prediction**

**Arnav Chowdhury**

**12-08-2019**

**Santander Customer Transaction Prediction**

**INTRODUCTION**

Santander - A Spanish banking business giant has one mission "Help people and business prosper". Santander has been always looking for new and innovative ways to help its customers understand their financial health and stability and identify which products and services might help them achieve their monetary goals.

Santander has been working extensively with its data science team to find better and effective solution for its customer.

**Problem Statement:**

The goal of this project is to predict a binary classification problem: Will customer buy this product? Can a customer pay the loan? etc. so that the bank can target only specific customers and better utilize resources.

As the name of the project stands "Customer transaction prediction", our primary goal here is to predict will the transaction be complete or not(loan/banking products).

In this project we aim to clean the data, visualize different relationship between variables and also figure out new features that are better predictors of our 'target' variable.

**HYPOTHESIS**

Customer transaction prediction can be based on the following hypothesis:

1. Type of customer? (Employed/Businessman/Home-maker/Retired)
2. Average household income in a year?
3. Previous loan history?
4. Type of loan?
5. Gender, Age, etc.

**DATA**

On exploring both the datasets we find:

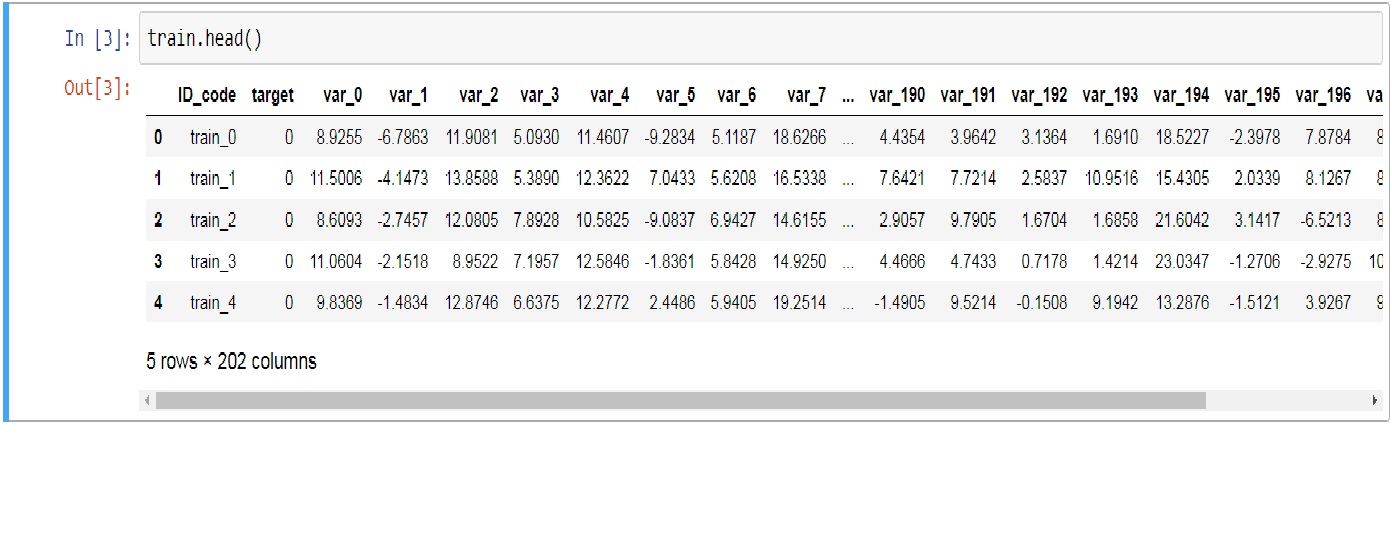
train dataset - contains 2,00,000 rows and 202 columns.

test dataset - contains 2,00,000 rows and 201 columns.

The 'train' dataset contains 'ID\_Code' and 'target' column and var\_0 to var\_199(200 columns) as the rest.

The 'test' dataset contains exactly the same column names but misses on 'target' column as the whole agenda is to predict the target column in the 'test' dataset.

**Figure 1.1 Santander customer transaction prediction sample data.**



As you can see in the data sample provided above, apart from the 'target' and 'ID\_code' column rest of the columns are just a bunch of random numbers. All the numbers have 4 digits after decimal point. Rest is like black dots. It doesn't convey any kind of information.

It would not be wise to consider any specific column variable as predictors of 'target' column which is only binary classified column. So in our problem here we will consider all the variables(var\_0 to var\_199) as predictors of 'target' column.

**METHODOLOGY**

**Pre-Processing**

Data pre-processing is the technique which is used to transform raw data into useful and efficient format. This is also called as Exploratory Data Analysis. Most Analysis require the data to be uniformly distributed. So, let's clean and visualize the data with the help of different graphs and plots before we can apply any of the Machine learning model on it.

**Missing Value Analysis**

On exploring both the 'train' and 'test' dataset, we were not able to find any missing value in the entire dataset. As there were 'no' missing values present in the dataset we proceed with the further observations and sections.

**Outlier Analysis**

Outliers were detected in the analysis using box plots. Outliers are observations which are inconsistent with rest of the dataset. Sometimes outliers also contain useful information about the dataset.

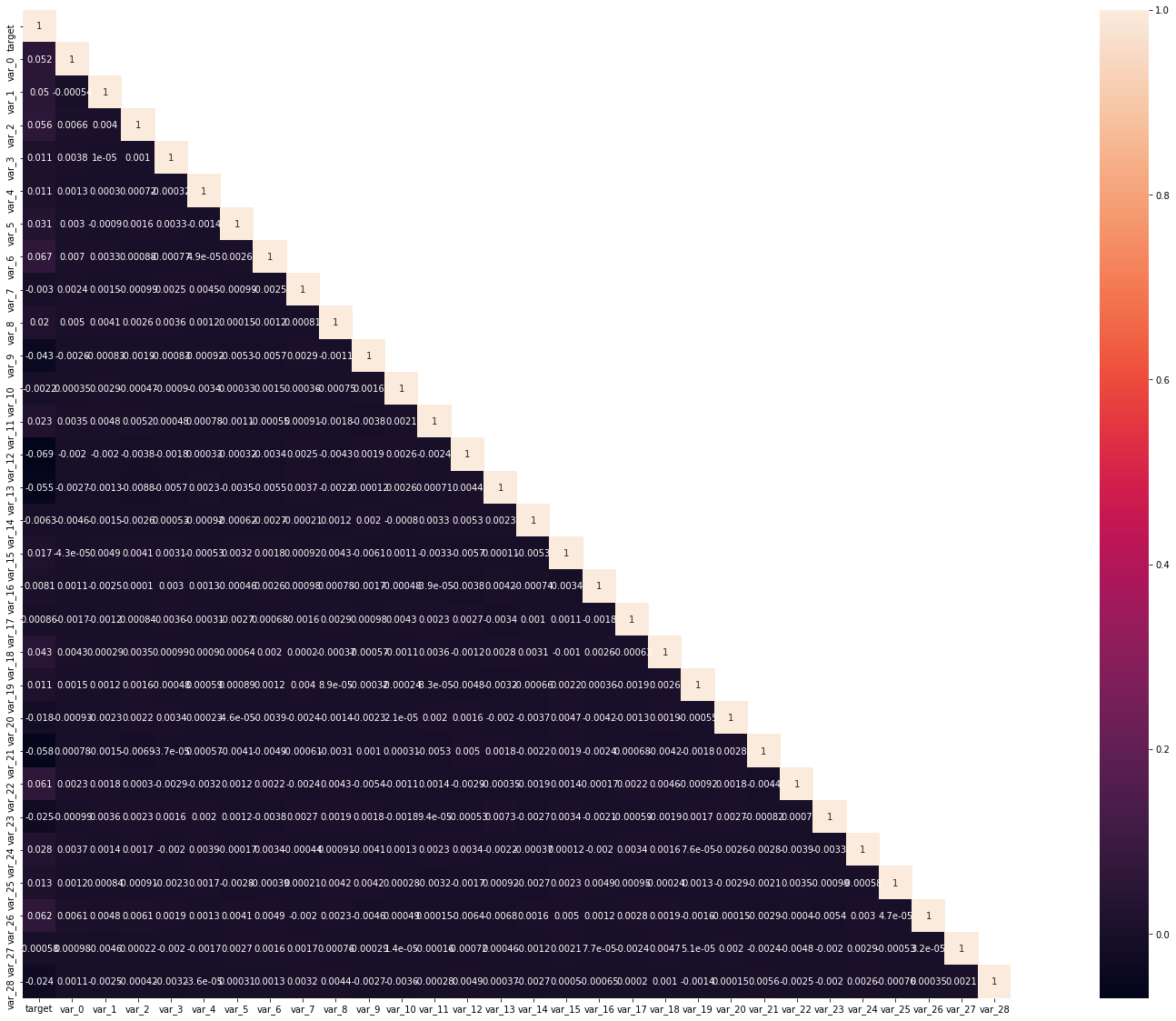
In our outlier analysis we saw there are 197333 observations which were classified as outliers in the 'train' dataset. This is 99% of the dataset which may contain very useful information about the dataset

As removing 99% of the dataset is not a wise-able option, we exclude the outlier analysis and carry on with further pre-processing techniques.

**Feature Selection**

Feature Selection is a technique used to select a subset of relevant features for use in model construction. It is also called as dimensionality reduction as the dimensions of the dataset are reduced for relevant features.

Features are selected based on their scores in various statistical tests for their correlation with the outcome variable. Correlation plot is used to find out if there is any multicollinearity between variables. The highly collinear variables are dropped and then the model is executed.



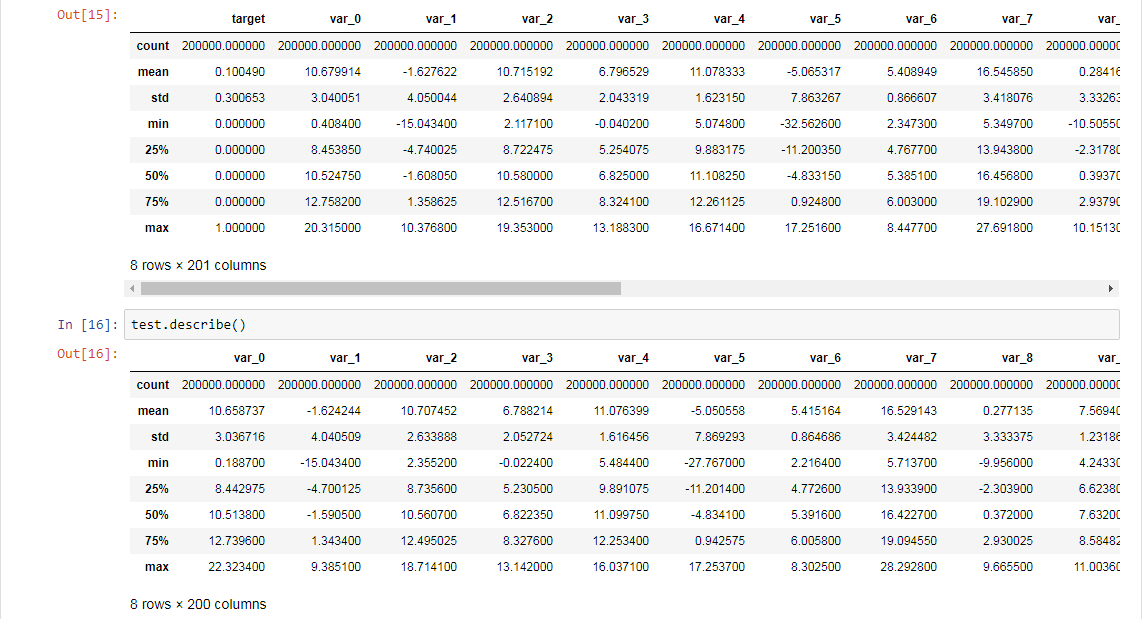
**Figure 2.1 Correlation plot of first 30 variables**

As we can see from the above plot that there is no collinearity problem between independent continuous variables. Also on checking for collinearity problems in the whole dataset we donot find any collinearity problems.

ANOVA (Analysis of Variance)

ANOVA test is carried out to check if groups of independent variable makes any effect on the target and helps in generating hypothesis.

As our dataset just contains only random numbers with no description of the Independent variables we skip the test. However, on visualizing the central tendency of each variable, we gain conservative insights into the data.



**Figure 2.2 Measures of central tendency**

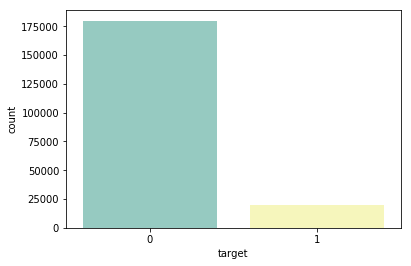
As we can see in both the datasets above,

* Standard Deviation is relatively large for both 'train' and 'test' dataset.
* Max, Min and STD are quite close in both the datasets.
* Min is distributed over a large number of range.

**Visualizations**

Let's visualize different aspects of the dataset to get more statistical insights into our data.

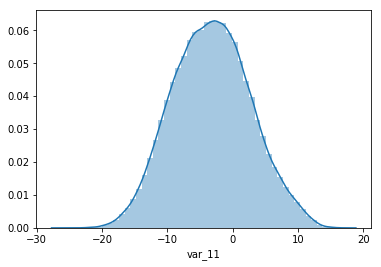
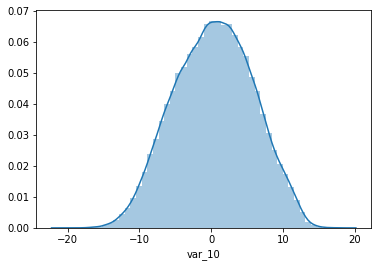
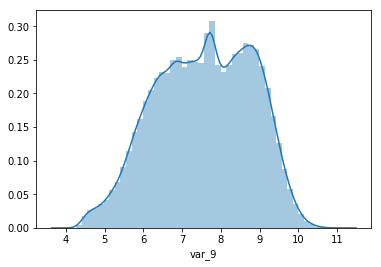
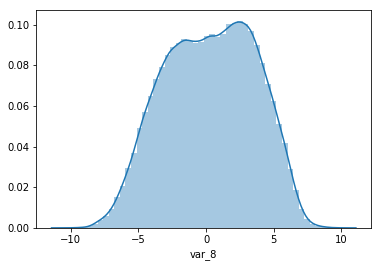
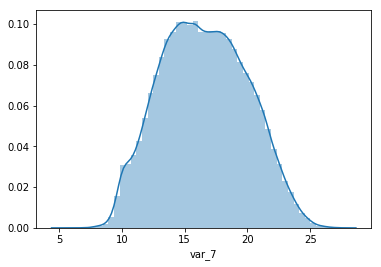
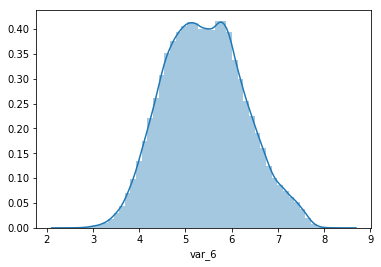
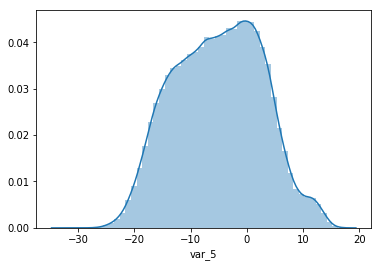
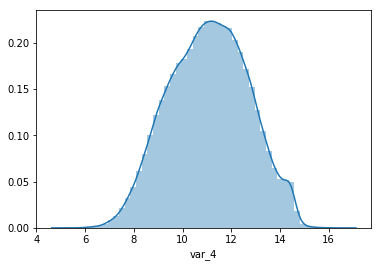
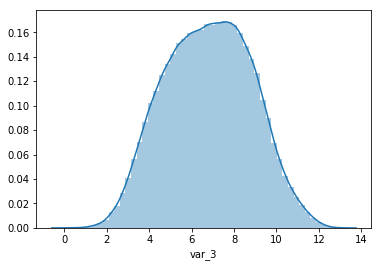
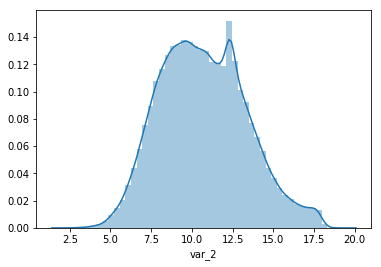
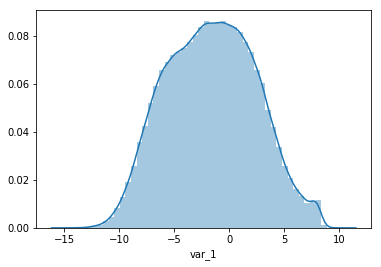
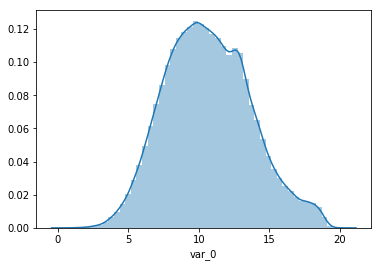
Firstly we visualize the distribution of 'target' variable



**Figure 2.3 Distribution of 'target' variable**

As we can see from the figure above the distribution of the target class is highly imbalanced. This will be a problem for us during modelling. To handle target class imbalance we will follow sampling techniques in the later part.

Let's now visualize some of the Independent variables from our dataset.



**Figure 2.4 Distribution of continuous variable**

We can see from the above that the Independent classes are fairly uniform in distribution. For the purpose of the report we have plotted only the first 12 variables in the dataset. The rest can be glared at in the code file attached.

Let's look at the distribution of Independent variable with respect to 'target' column of train dataset.

**Figure 2.5 Distribution of Independent variables with 'target'**

We can observe here that there are considerate number of features with significant different distribution for both the target variable '0' and '1'. Also most of the variables seems to be almost uniformly distributed.

Now let's look at the distribution of all the features in both 'train' and 'test' dataset.



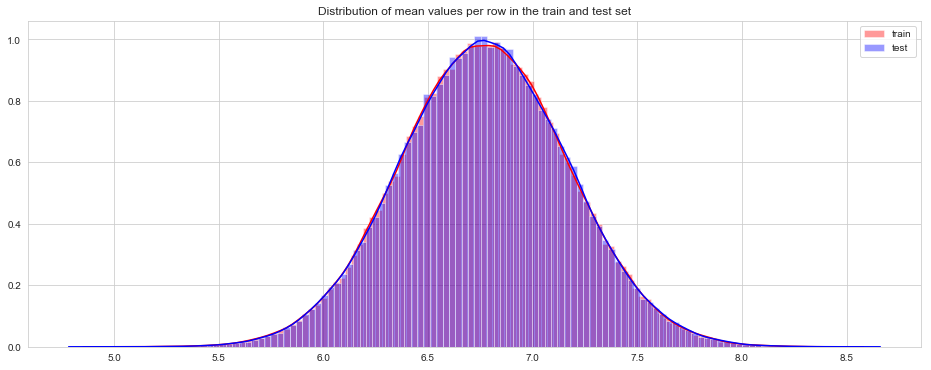


**Figure 2.6 Distribution of all features in train and test**

As we can see in the figure 'train' and 'test' seems to be well balanced with respect to distribution of numeric variables.

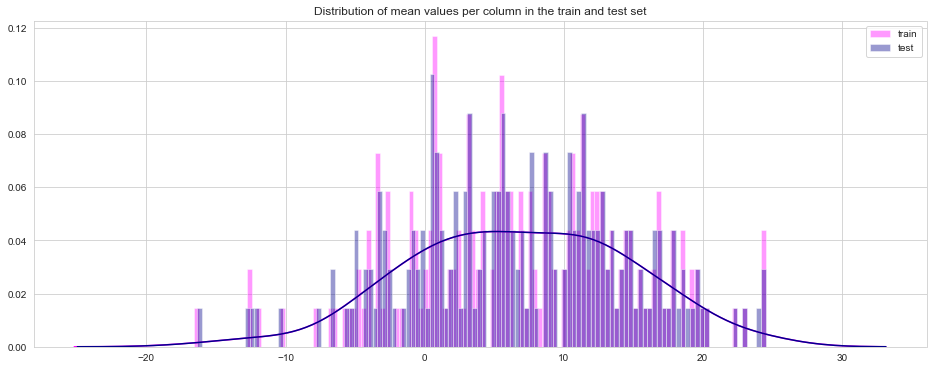
**Distribution of Mean and Standard deviation**

Distribution of Mean value per row in the datasets.



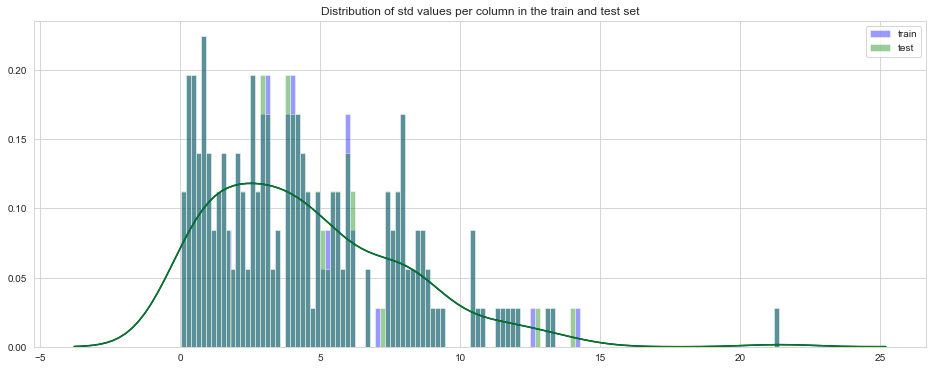
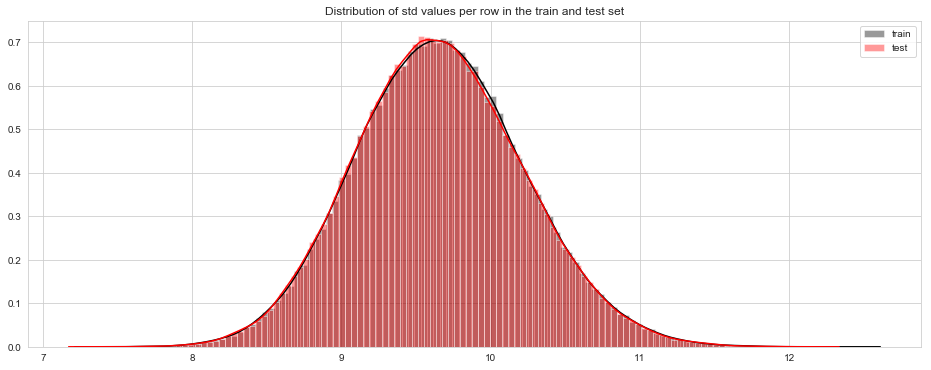
**Figure 2.7 Distribution of mean values per row**

Distribution of Mean value per column in the datasets.



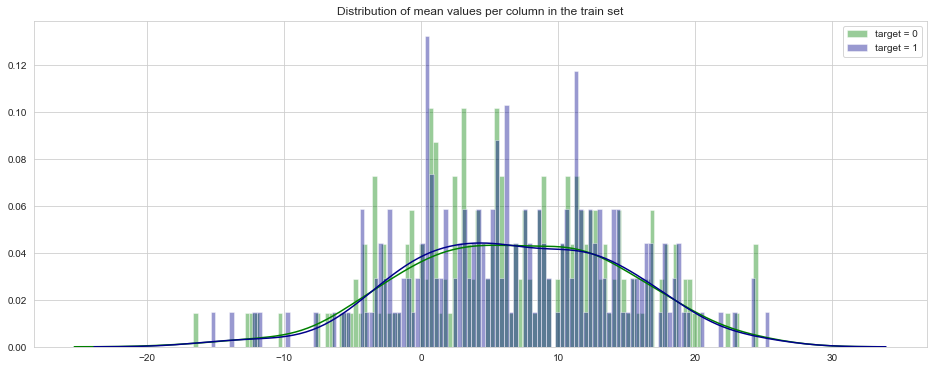
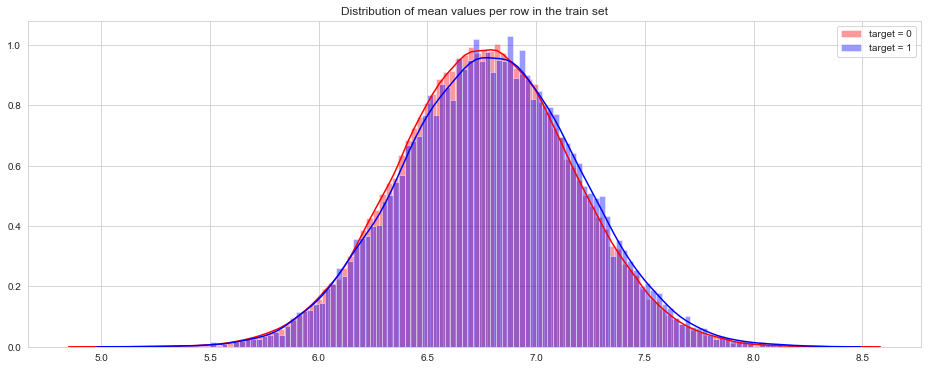
**Figure 2.8 Distribution of mean values per column**

Distribution of Standard deviation per row and column in the datasets



**Figure 2.9 Distribution of Standard Deviation per row and column**

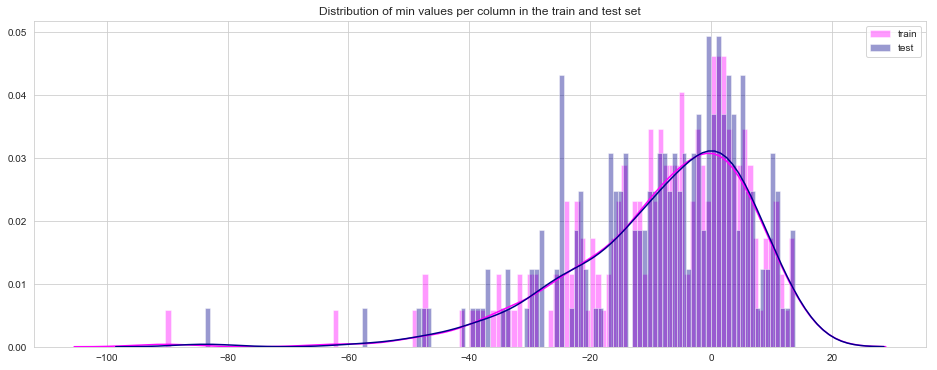
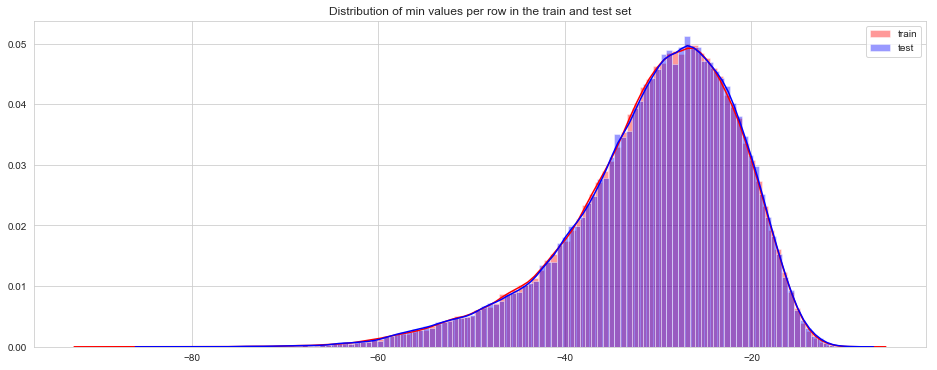
Distribution of Mean grouped by value of target per row and column



**Figure 2.10 Dist. of Mean grouped by 'target'**

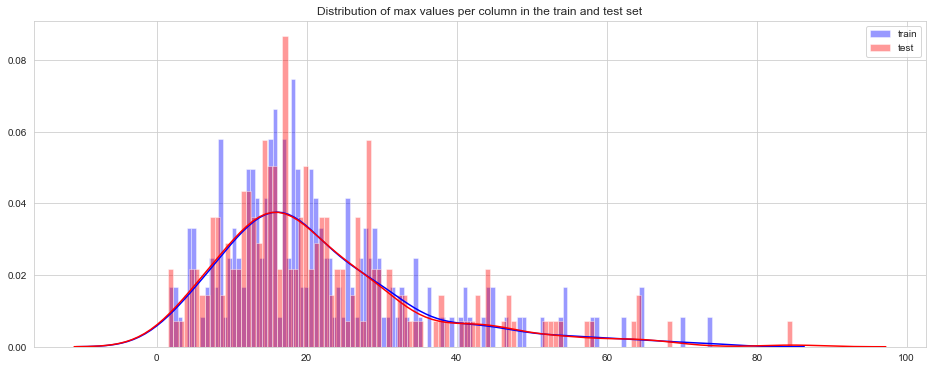
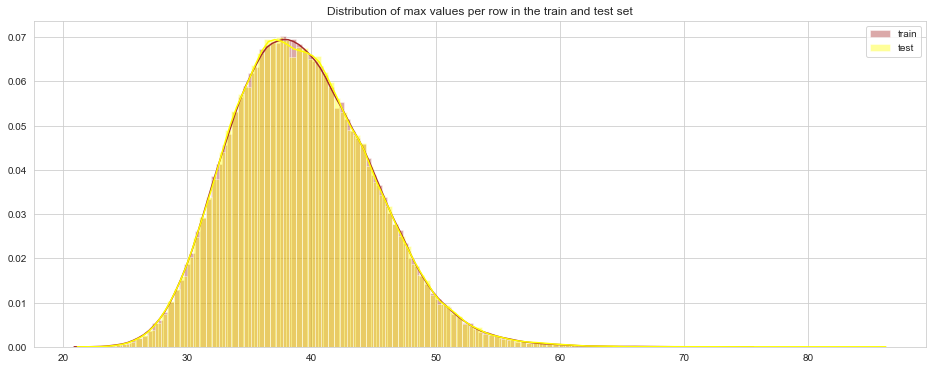
**Distribution of Min and Max values**

Distribution of Min value per row and column in train and test dataset



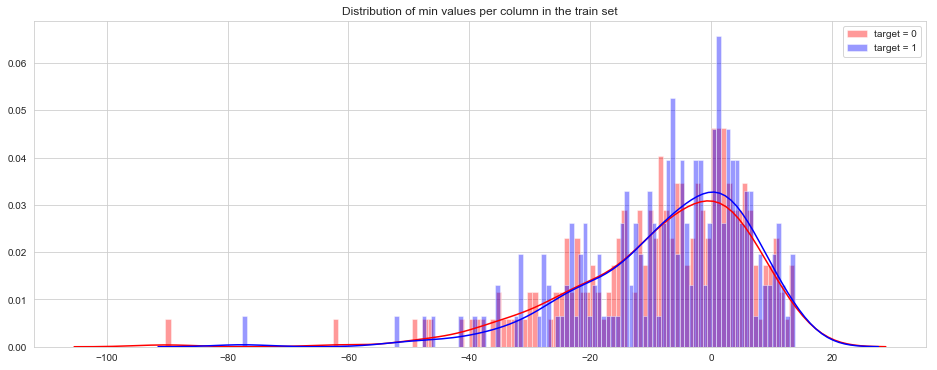
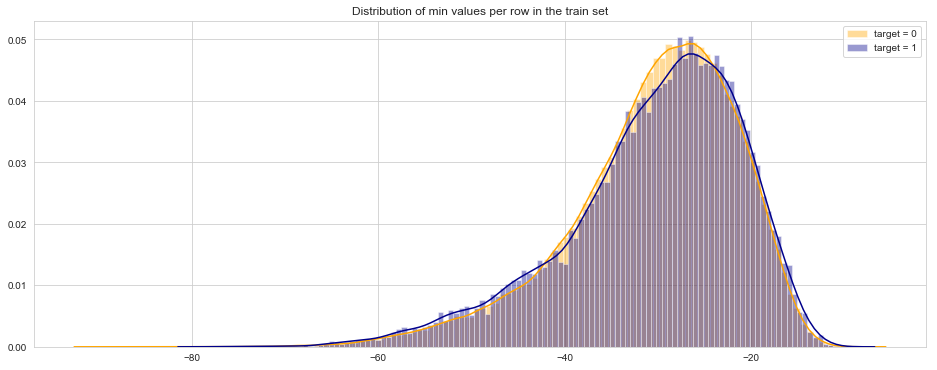
**Figure 2.11 Dist. of Min values per row and column**

Distribution of Max values per row and column in train and test dataset



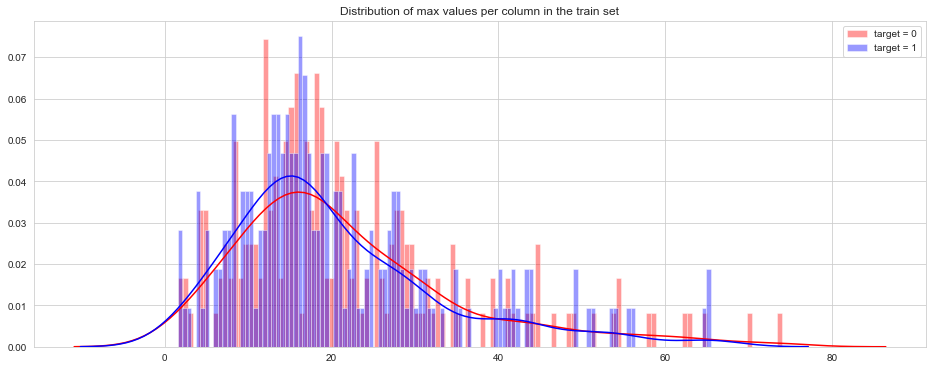
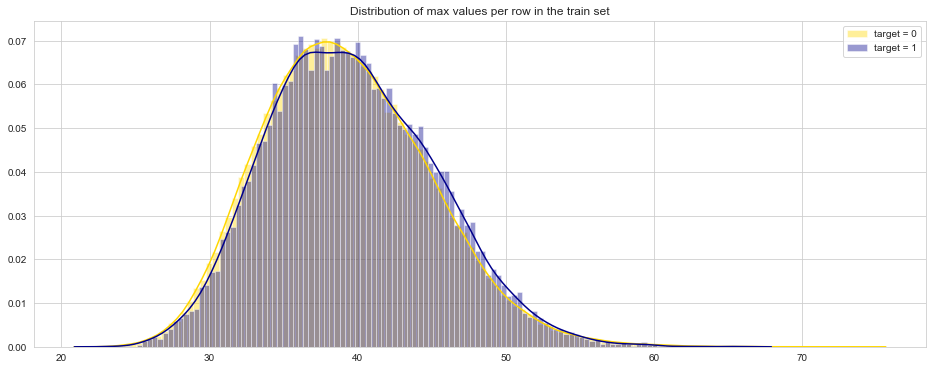
**Figure 2.12 Dist. of Max values per row and column**

Distribution of Min values grouped by value of target per row and column



**Figure 2.13 Dist. of Min grouped by 'target'**

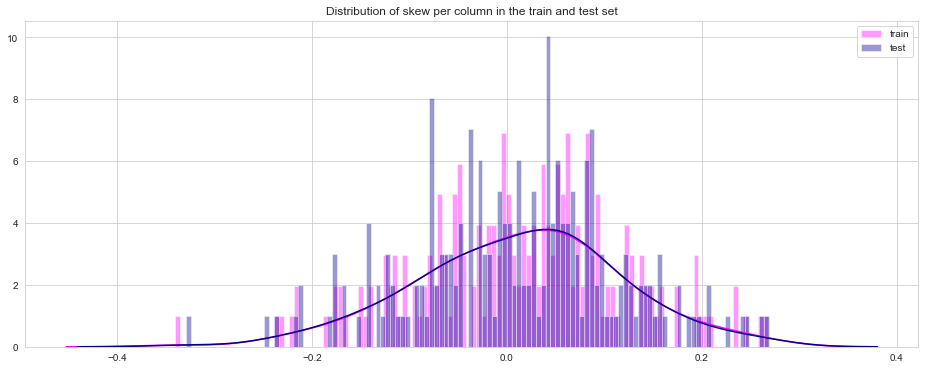
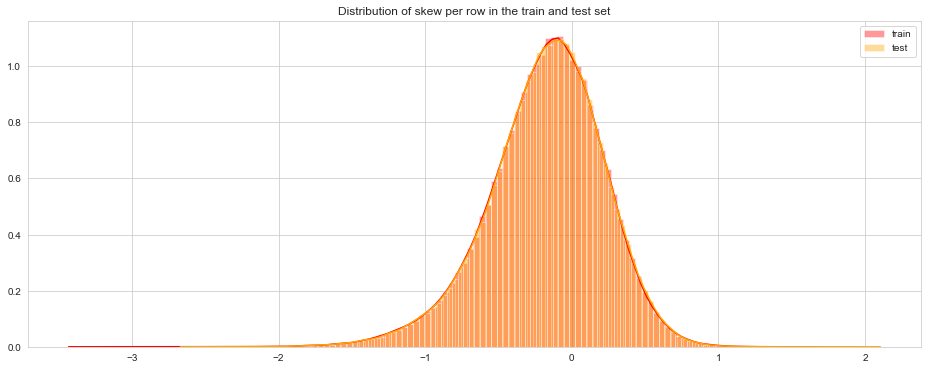
Distribution of Max values grouped by value of target per row and column



**Figure 2.14 Dist of Max grouped by 'target'**

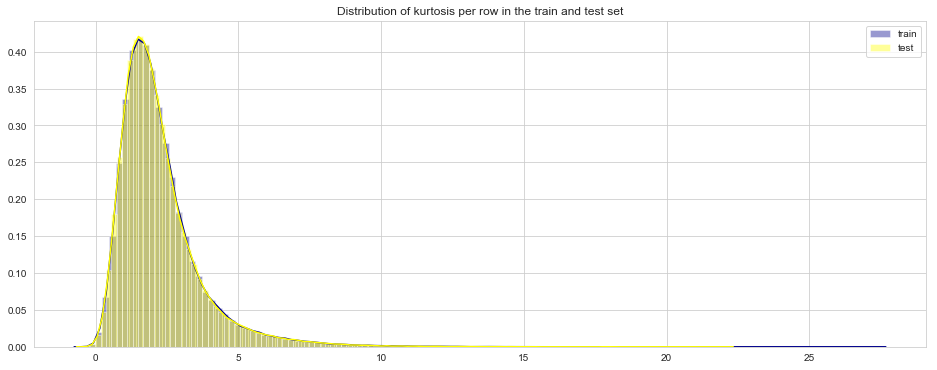
**Distribution of Skewness and Kurtosis**

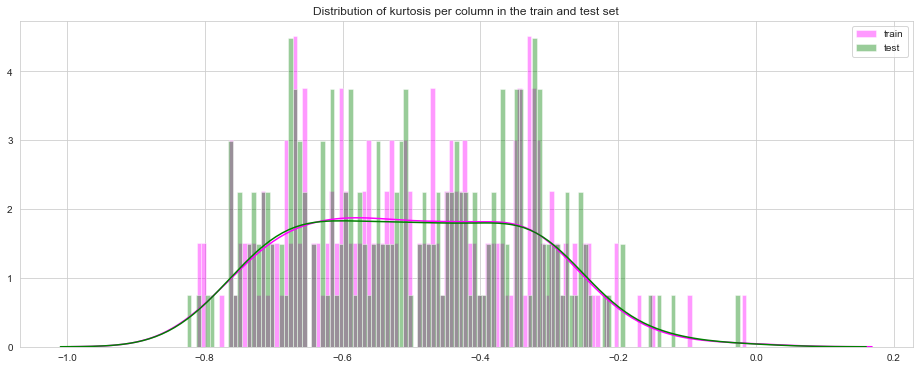
Distribution of Skewness per row and column in the train and test dataset



**Figure 2.15 Dist of Skewness per row and column**

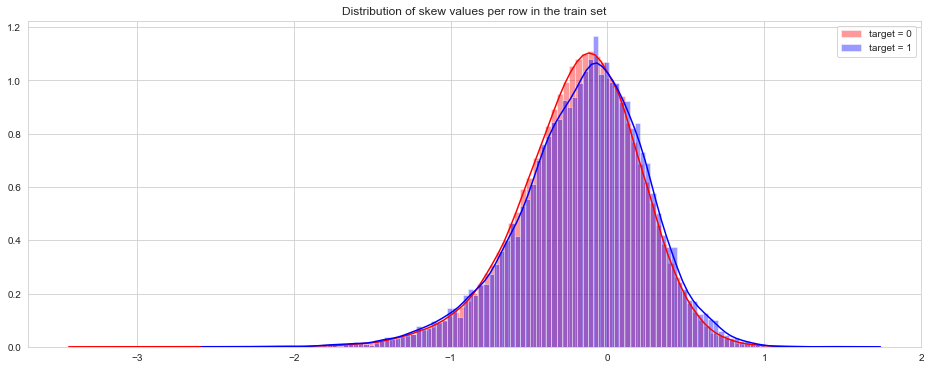
Distribution of Kurtosis per row and column of train and test dataset.

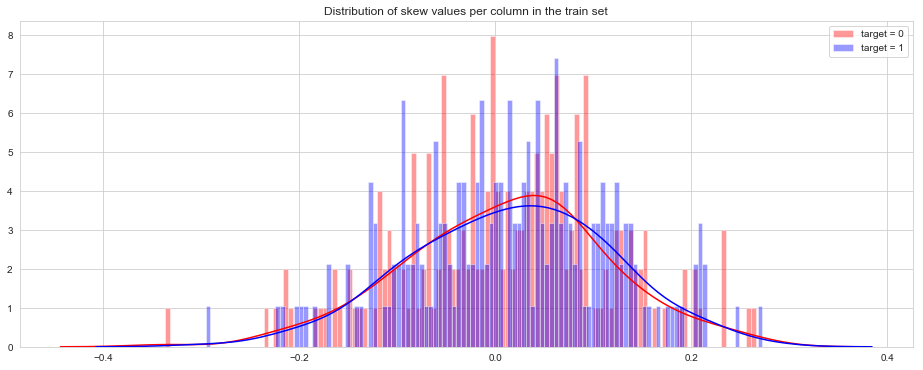




**Figure 2.16 Dist of Kurtosis per row and column**

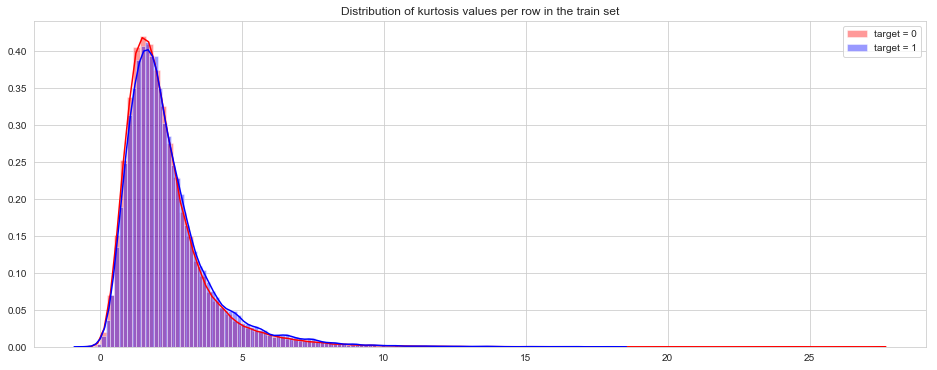
Distribution of Skew values grouped by value of target per row and column

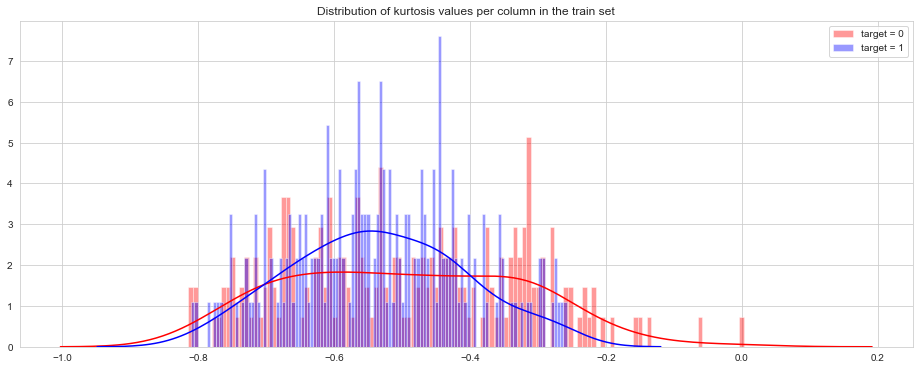




**Figure 2.17 Dist of Skew values group by 'target'**

Distribution of Kurtosis values grouped by value of target per row and column





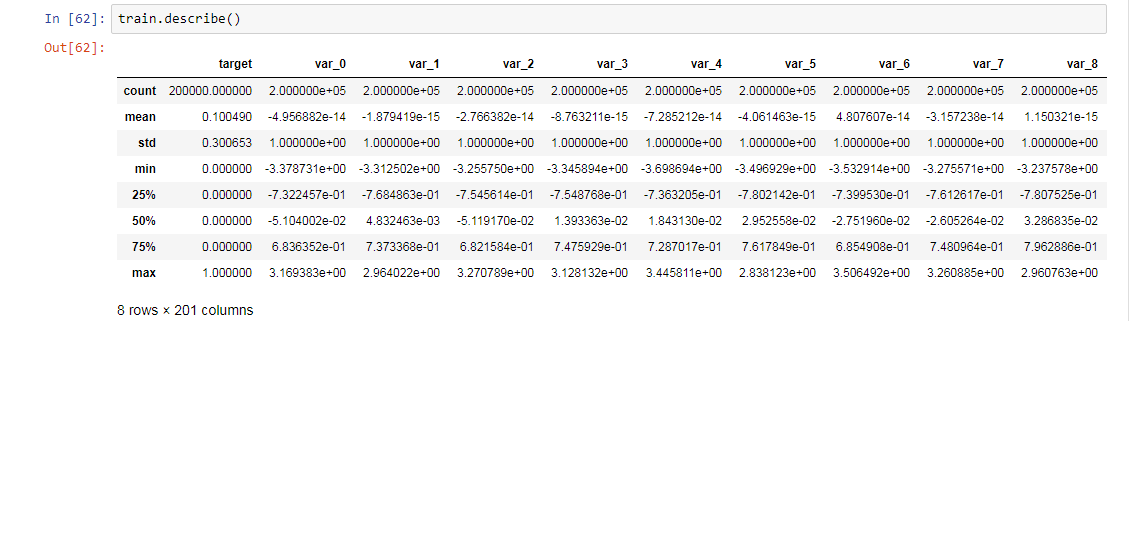
**Figure 2.18 Dist of Kurt values group by 'target'**

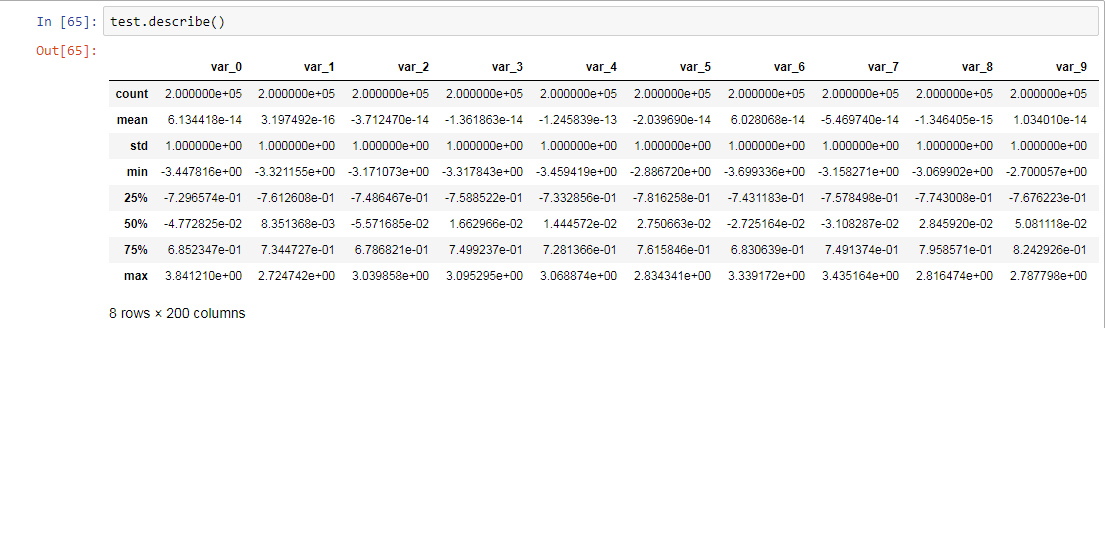
**Feature Scaling**

Feature Scaling is the process to normalize the range of Independent variable or features of data. In data processing it is also known as Normalization.

We followed the method of Standardization to scale the data.

On applying Standardization 'Standard Deviation' for all the features were bought within a range in both the train and test dataset.





**Figure 2.19 Standardized values after Feature Scaling**

**Feature Engineering**

Feature Engineering is the process of creating new features on the dataset that makes modelling the data better. It is about preparing the proper input dataset.

Based on the analysis we have done above, a set of 8 different features were created for the Input dataset for better modelling purpose.

**Sampling**

Sampling is a process of representing the whole population in predetermined number of observations. It is a process of selecting subsets based on the characteristics of the entire population.

As we have seen in the 'visualizations' part we have a 'target class imbalance' problem in the dataset. To overcome such difficulty and aid in better model development we followed 'Minority Under sampling' of the dataset to bring both the 'target' class in proportion . this will help us in following a unbiased approach to modelling.

**MODELLING**

Modelling is a process in data-science that uses data mining and probability to forecast outcomes. Each model is made up of number of predictors which are variables that are likely to influence future results.

**Decision Tree**

It is one of the predictive models based on a branching series of boolean tests. It is an algorithm which uses tree like graph to model decisions.

Each leaf node represents an attribute.

We have used C5.0 algorithm of the decision tree classifier.

Using decision tree we have got the following statistics:

|  |  |
| --- | --- |
| Accuracy | 62.027 % |
| False Negative Rate | 37.725 % |
| Precision | 61.641 % |
| Recall | 62.274 % |
| ROC(AUC) | .620 |

**Random Forest**

Random forest is an ensemble technique that consists of many decision trees. This method combines 'bagging' idea and random selection of features.

Using random forest we achieved the following statistics:

|  |  |
| --- | --- |
| Accuracy | 73.718 % |
| False Negative Rate | 25.062 % |
| Precision | 73.718 % |
| Recall | 74.937 % |
| ROC(AUC) | 0.737 |

**Logistic Regression**

Logistic regression is a statistical model which uses the weights or coefficients to develop a model and apply the same. It uses logistic function to predict the outcome.

Using logistic regression we achieved the following statistics:

|  |  |
| --- | --- |
| Accuracy | 80.186 % |
| False Negative Rate | 20.611 % |
| Precision | 80.965 % |
| Recall | 79.388 % |
| ROC(AUC) | 0.802 |

**KNN (K Nearest Neighbour)**

It is a simple algorithm which stores all available cases and classifies new cases based on a similarity measure. For classification it uses the 'majority' and 'minority' rule to classify data.

Using KNN we achieved the following statistics:

|  |  |
| --- | --- |
| Accuracy | 55.758 % |
| False Negative Rate | 75.769 % |
| Precision | 65.901 % |
| Recall | 24.230 % |
| ROC(AUC) | 0.558 |

As we can see KNN did not provide us a good statistics here.

**Naive Bayes**

It is a classification algorithm based on probabilistic classification. It works on Bayes theorem of probability to predict the class of unknown dataset.

All variables are considered independent given the target value and multicollinearity does not exists.

It is not a single algorithm but a family of algorithms where all of them share a common principle.

Using naive bayes we achieved the following statistics:

|  |  |
| --- | --- |
| Accuracy | 79.564 % |
| False Negative Rate | 21.430 % |
| Precision | 80.223 % |
| Recall | 78.569 % |
| ROC(AUC) | 0.796 |

**MODEL SELECTION AND CONCLUSION**

From the modelling section we have gained different evaluation metrics based on which we will choose the most appropriate model for our project.

The following is a comparative description of the model performance.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Desicion Tree** | **Random Forest** | **Logistic Regression** | **KNN** | **Naive Bayes** |
| Accuracy | 62.027 % | 73.718 % | **80.186 %** | 55.758 % | 79.564 % |
| False Negative Rate | 37.725 % | 25.062 % | **20.611 %** | 75.769 % | 21.430 % |
| Precision | 61.641 % | 73.718 % | **80.965 %** | 65.901 % | 80.223 % |
| Recall | 62.274 % | 74.937 % | **79.388 %** | 24.230 % | 78.569 % |
| ROC(AUC) | .620 | 0.737 | **0.802** | 0.558 | 0.796 |

We can see from the above the above that Logistic Regression outperforms others .

We can also see that Naive Bayes prediction is quite similar to Logistic regression however, naive bayes is a discriminitive model as it assumes that the dataset follows a particular pattern and uses a Gaussian or Bernoulli algorithm and even violates the assumption of conditional independence of the features. On the other hand Logistic Regression is a generative(informative) model.