RISHABH JAIN

[rishujain273@gmail.com](mailto:rishujain273@gmail.com)

+91-8860486224

Case Study – Credit Card Fraud Detection

1. Data Introduction

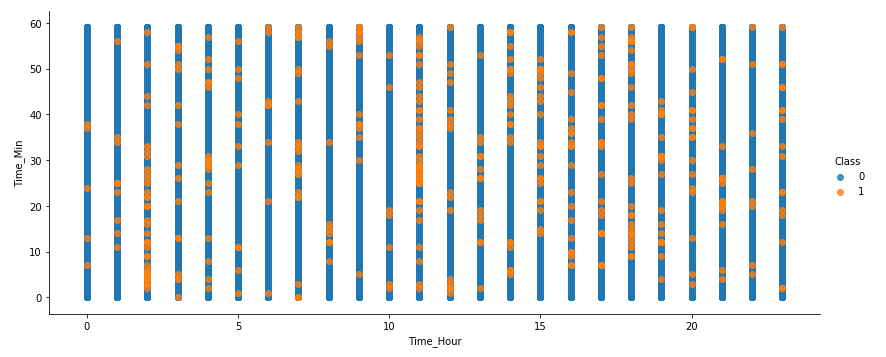
The Credit Card data contains 31 variables which includes Time in seconds, Amount and various variables which are result of PCA dimensionality reduction.

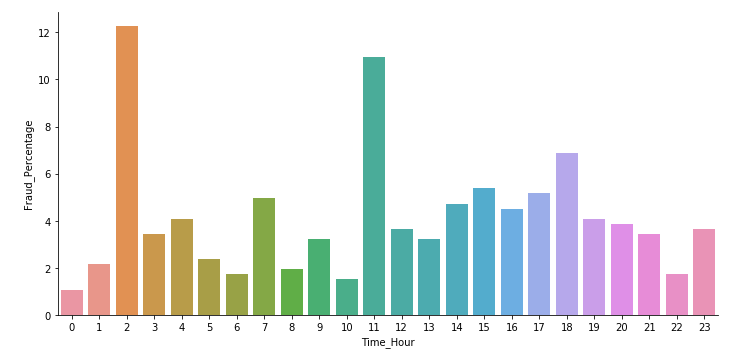
All the information out of given Variables which are result of PCA dimensionality reduction have no variable information. So, analysis will start by visualizing the distribution of variable in the dataset.

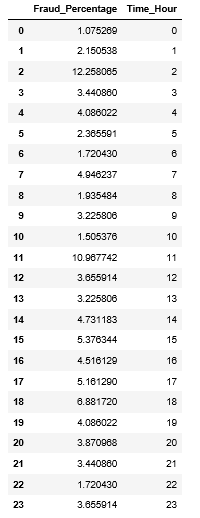
1. Exploratory Data Analysis
   1. Time

Time is given in seconds which is not giving much of value so minutes and hours will be extracted from the data to see any trend. To\_timedelta() function is used to extract minutes and hour data from the existing variables.

It is observed that there is no particular bad minute-hour combo where the chances of happening fraud is high. But, one point is observed that at some specific hours the frauds are happening more than the others.



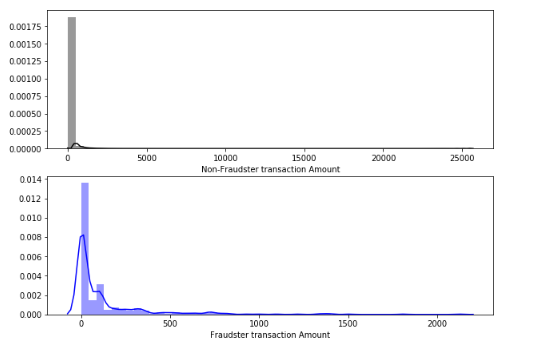




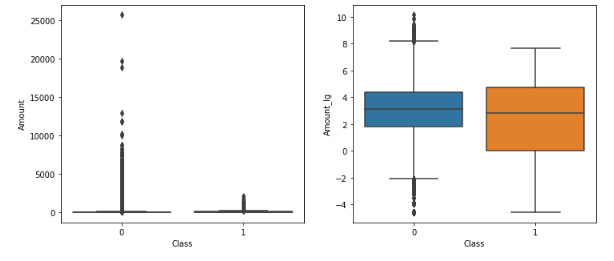
So at hour 2 and 11, major percentage of fraud is happening i.e., 12.2% and 10.9% of the frauds are happening at this time. It depicts that hourly data is necessary in building the model.

* 1. Amount

The variable amount is highly skewed for both the classes as given in the below figures shown below:



So, log transformation is used by using np.log() function to normalize the Amount data. Below is the before and after log transformation boxplot view of Amount where Amount\_lg represents the log transformed Amount.

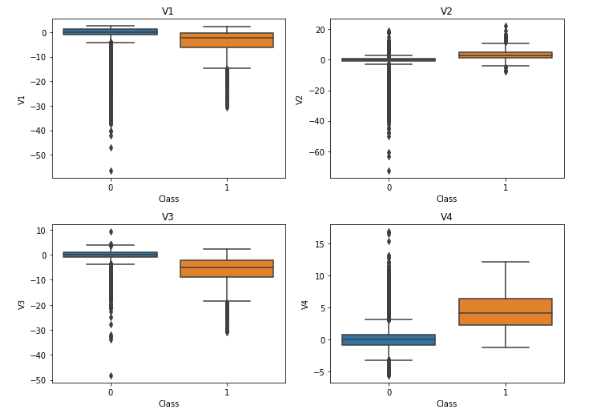


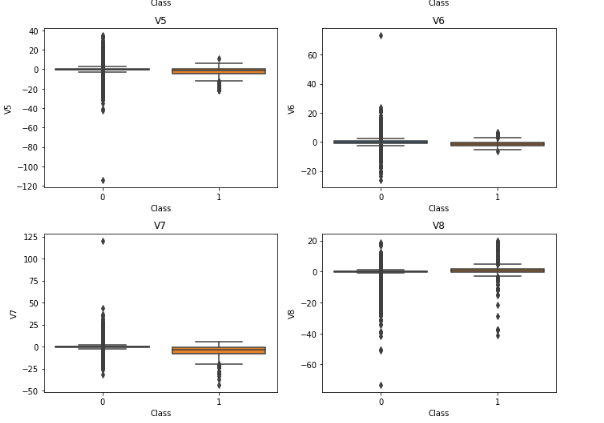
The data distribution now looks pretty much into shape and normalized also as it can be seen that the data has been scaled as the data lies in between range of [-10,10] which in this dataset most of the variables are.

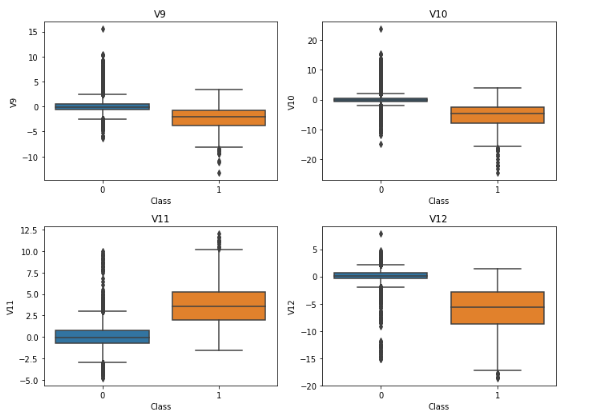
Other Variables are not normalized because most of them are already in normalized shape but there were outliers present in them due to which the distribution plot looks pretty much skewed. Therefore for them outlier treatment is performed.

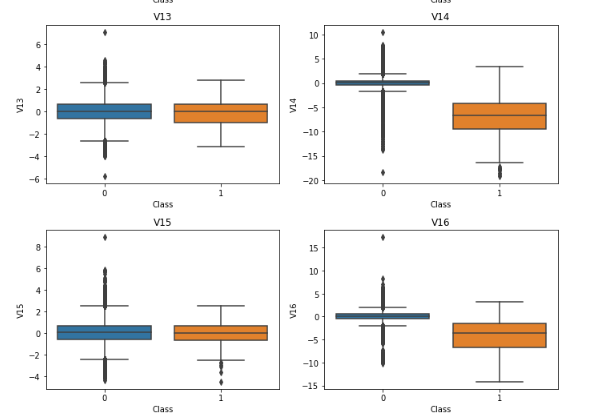
1. Outlier Analysis

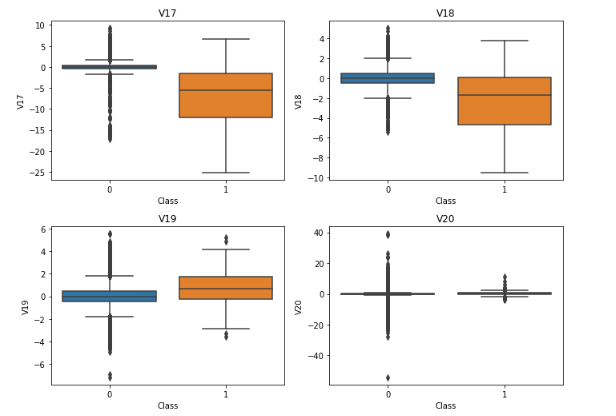
Box plot for all the variables from V1-V28 are given below:

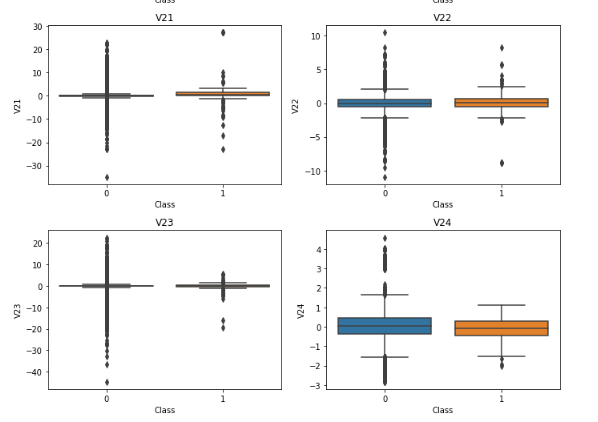


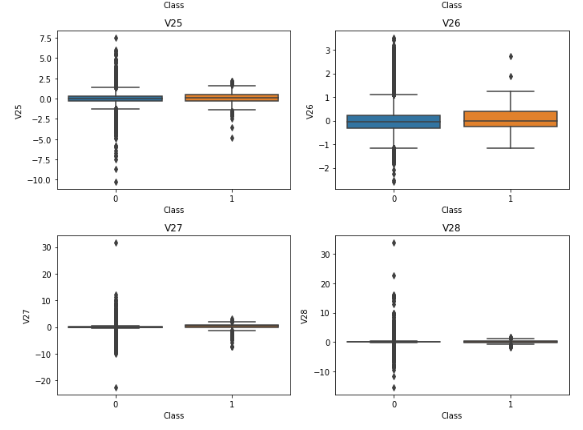




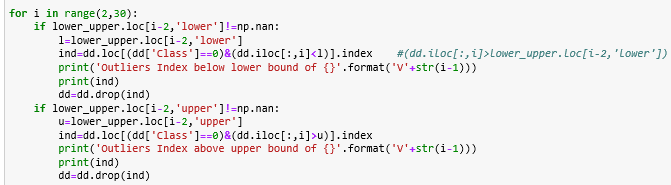








The threshold is set for each variable in csv file in which values are manually entered (so it can be changed at any point of time) and that data file is used to remove outliers by below code.



After implementation of above code, now the data looks less skewed than before.

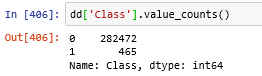
1. Correlation Analysis

For the continuous variables correlation analysis is done but no such major correlation is found at the continuous level.

Also, for the categorical variables i.e., Time\_Hour and Time\_Min, no correlation was found there also. So the data is having no correlation in between the variables that’s why no removal of variable.

1. Unbalanced Dataset

The dataset is highly imbalanced i.e., Fraudulent data is very less in comparison to Non-Fraudulent data because of which if the model predicts all data as Non-Fraud, it will achieve 99% of accuracy.



So to deal with this imbalanced data distribution SMOTE method is used to synthetically develop new data rows near Fraud data.

SMOTE is done during Cross Validation so as to avoid biased selection of Validation set.

1. Data Science Modelling

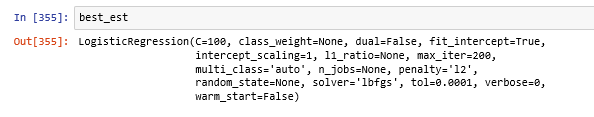
In this Case Study two data science model are used with hyper parameter optimization of those classifiers. Those classifiers are given below:

1. LogisticRegression
2. DecisionTree

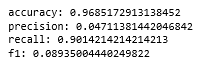
Both of them are used with RandomSearchCV() optimization of hyper parameters to find the best estimator. Below are the analyzed performance of the model and metrics to make a call for a model.

1. LogisticRegression

Best estimator having best configuration on the data set is :;



Performance metrics for the model :



Area under the Precision-Recall curve or AUC Score is calculated using roc\_auc\_score()

function and it comes out to be:

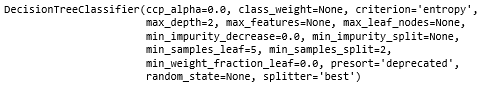


And average precision-recall score of the model comes out to be:

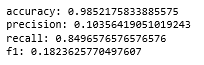


1. DecisionTree

Best estimator having best configuration on the data set is :;



Performance metrics for the model:



Area under the Precision-Recall curve or AUC Score is calculated using roc\_auc\_score() function and it comes out to be:



And average precision-recall score of the model comes out to be:



1. Conclusion

After looking into both of the models’ performance metrics it can be said that Logistic regression model with RandomSearchcv() optimization has performed well on the test set as

1. The AUC score is 0.93 which is more than for Decision Tree Classifier stating that the model had performed well for different threshold values.
2. Despite of greater AUC score the model has high Precision Recall score i.e., 0.81 which impacts the model greatly as it is the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight.
3. Although the model accuracy with Decision Tree Classifier (i.e., 98.5%) is greater than that of logistic regression, but for calculating the accuracy part the threshold probability set for binary classification would be 0.5, thus it’s not a good metrics to evaluate the model upon.