**CLASSIFICATION PROJECT – CREDIT CARD FRAUD DETECTION**

**The main objective of this report** is to classify fraud credit card transactions using classification algorithms.

The report was prepared in 2 sections. “Main Section” explains the dataset, a brief description of data exploration, models, results, key findings and suggestions. “EDA and FE” section provides the extensive work done for EDA and FE. I separated this section from the main section in order to summarize the main sections before going into details about the data.

1. **THE MAIN SECTION**

**The data set** has 31 features. 28 of them are called “V1-28” and these are the PCA outcomes of the original dataset (which we did not have access due to privacy policies). Other 3 features were classes, time and amount. The data contains info for 2 days (48 hours)

**Data Exploration**: The values of V1-28 are small numbers with low variety, so they were not min-max scaled for support vector machine classification. The amount was log-transformed and its values were also small numbers with low variety. The non-fraudulent transactions were 284315 and the fraudulent ones were 426. After outlier elimination (which was done separately for each class), the new number of those classes were 254234 and 426, respectively. Due to having low-computational power, I first randomdownsampled the majority class to make it 4260, then using SMOTE, I upsampled the minority class amd the new dataset consisted of same amount of fraudulent and non-fraudulent data: 4260. The plots for Data explanation is shown in Section 2 in detail.

**Models**: I used 4 different clustering methods (Support Vector Machine, Random Forest, AdaBoost, XGBoost) and combined their results using stacking (with Logistic Regression). Below are the hyperparameters used for each model. In each model, GridSearch was used for parameter optimization

1. **Support Vector Machine (SVM):**

svm\_param\_grid = {'C': [0.1, 1, 10], 'kernel': ['linear', 'rbf'], 'gamma': ['scale', 'auto']}

1. **Random Forest (RF):**

rf\_param\_grid = {'n\_estimators': [20,40,70,100,130], 'max\_depth' : [5,10,15,20],'max\_features':["auto", "sqrt", "log2"]}

1. **AdaBoost:**

adaboost\_param\_grid = {'n\_estimators': [50, 100, 150], 'learning\_rate': [0.1, 0.5, 1]}

1. **XGBoost:**

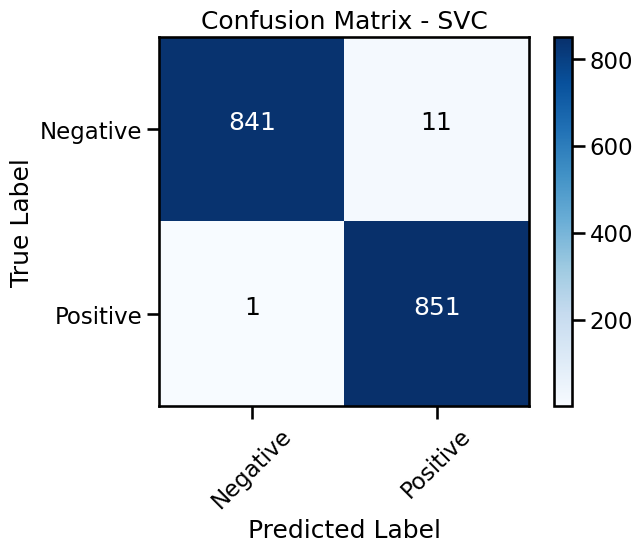
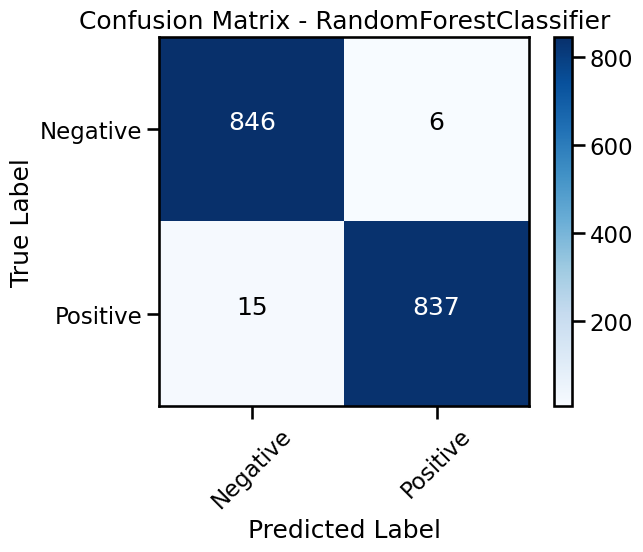
gradientboost\_param\_grid = { 'n\_estimators': [50, 100, 200,500], 'learning\_rate': [0.1, 0.5, 1],

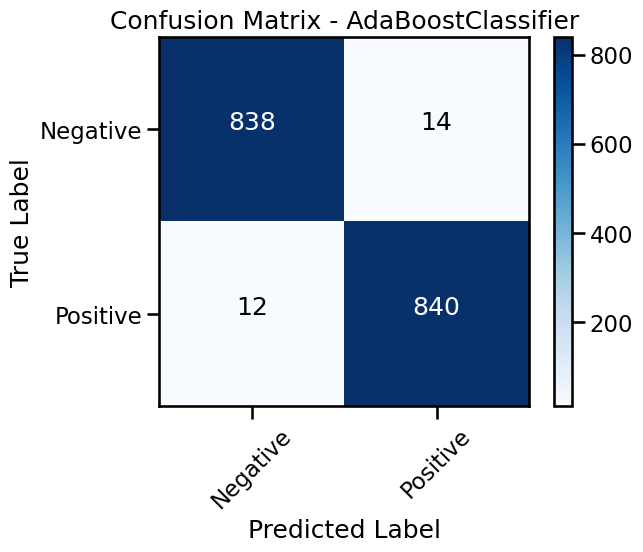
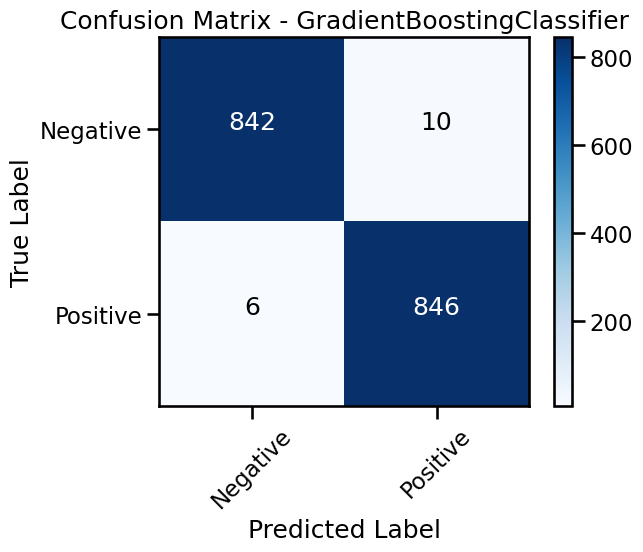
'max\_depth': [5, 10, 15,20]}

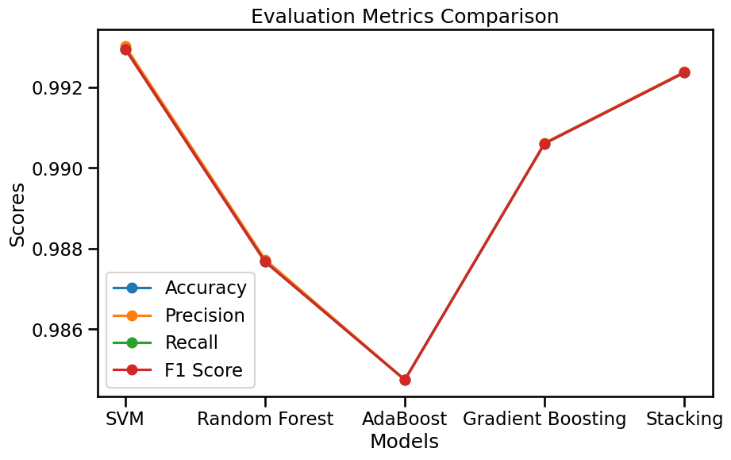
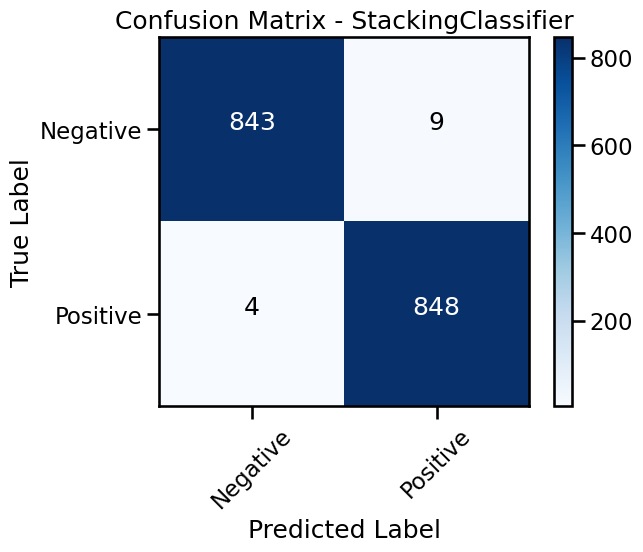
**Best Performing model:** The results of the models are listed in the table below:

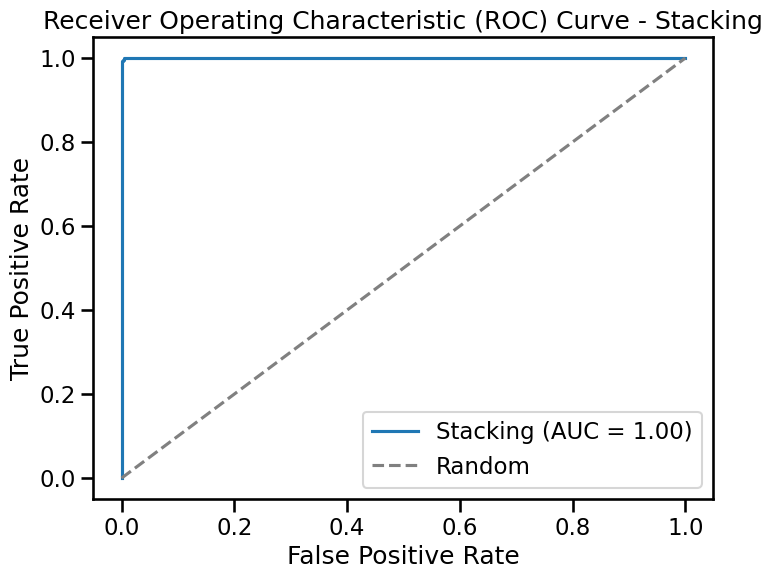
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Evaluation Metric** | **SVM** | **RF** | **AdaBoost** | **XGBoost** | **Stacking - Logistic Regression** |
| **Accuracy** | 0.992 | 0.987 | 0.984 | 0.990 | 0.992 |
| **Precision** | 0.993 | 0.987 | 0.984 | 0.990 | 0.992 |
| **Recall** | 0.992 | 0.987 | 0.984 | 0.990 | 0.992 |
| **F1** | 0.992 | 0.987 | 0.984 | 0.990 | 0.992 |

As the table shows, SVM performed the best, and combining the results from each model through Stacking did not provide better classification results. Below are the confusion matrices, plot of scores from each model and ROC curve obtained from stacking.





**Key findings:** Even though the dataset was reduced in size for better computational efficiency and the imbalance in the dataset was very high, the classification performance using any of the models considered was very high. Stacking did not increase the performance (I.e., its results were not better than the ones of SVM). This can be due to the much better performance of SVM compared to the other 3 methods that stacking did not help increase the performance. The reasons that SVM performed better results can be due to the followings:

Effective handling of complex relationships: SVMs can capture intricate relationships by utilizing kernel functions like RBF, enabling them to model complex patterns and capture subtle feature interactions.

Robustness against outliers: SVMs are less influenced by outliers as they focus on the points near the decision boundary, resulting in robustness when dealing with outlier-containing datasets.

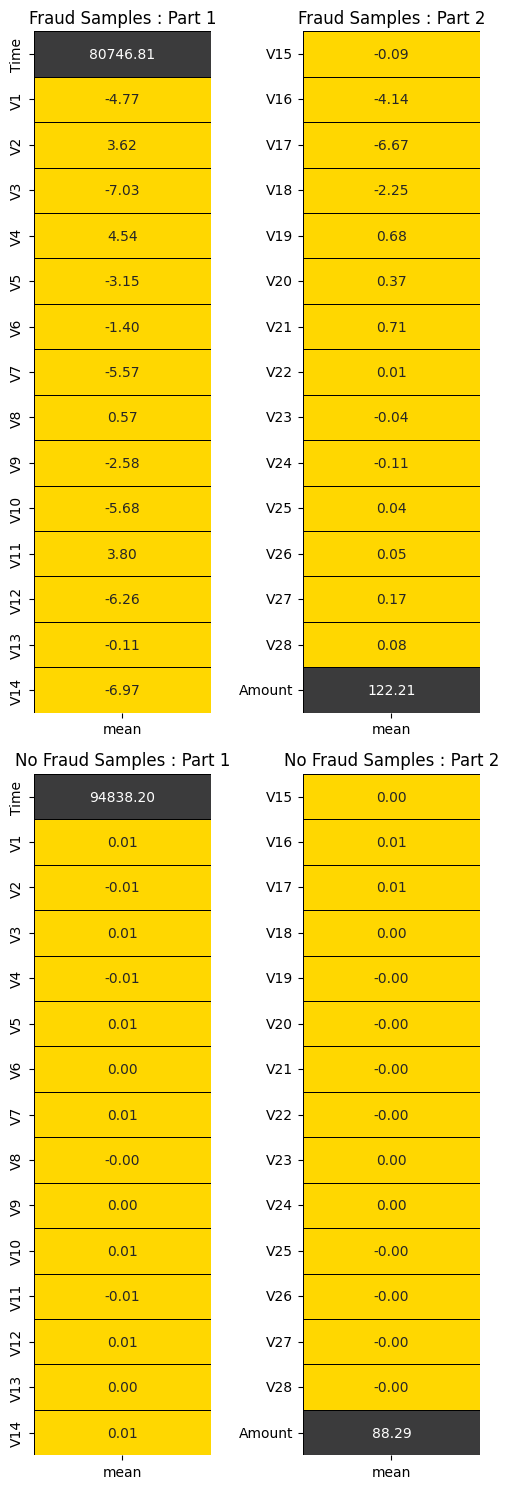
Effective in high-dimensional spaces: SVMs excel in high-dimensional data where other algorithms struggle, as they can find optimal hyperplanes to separate classes and overcome the curse of dimensionality.

Versatile kernel functions: SVMs offer various kernel functions to capture different types of relationships between data points, adapting to diverse data distributions.

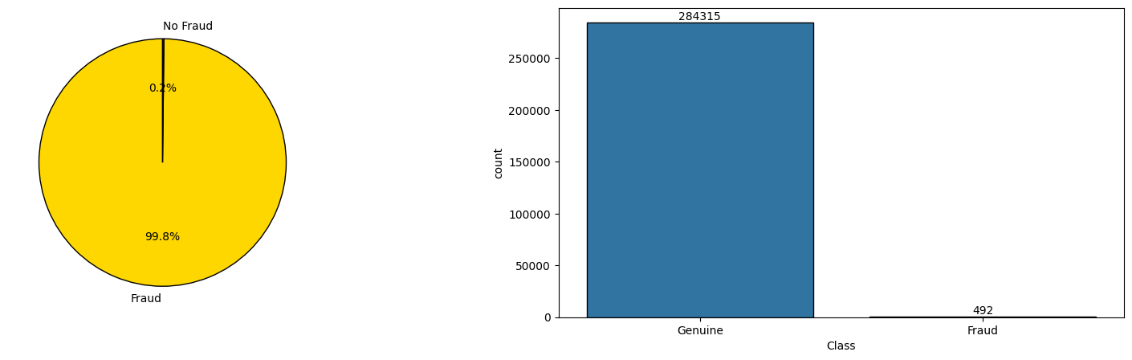
**Suggestions:** The good feature of a classification project is that results from different classifiers can be combined using stacking. Since I already used 4 classifiers stacked, adding another model like KNN is not expected to increase performance (in my experience). However, the param\_grids for each model can be finer, which might increase the performance of each model, and which in return can also increase the performance of stacking (maybe making the stacking results the best). The improvements for the project can also be made on the computational side. I decreased the data size a lot using undersampling due to the low computational power that I have. The results can be re-evaluated using the entire dataset (only upsampling of minority class to account). Moreover, Since the dataset had PCA of many outputs (whose names we do not know), there was no control on feature selection. If provided a dataset with all features with real variable names, feature selection might be performed better by analyzing the correlation of raw features.

1. **EDA AND FE**

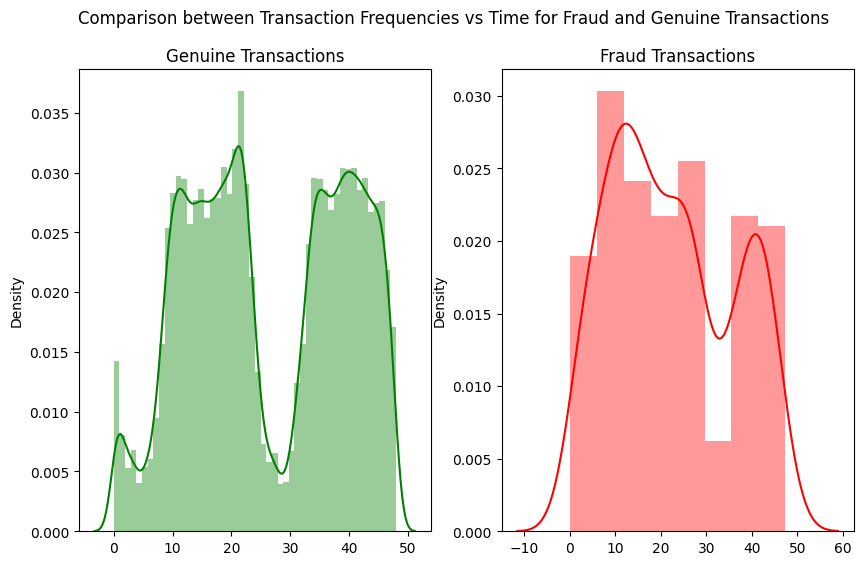
* The figure below shows how the mean values of fraudulent and non-fraudulent features are different.



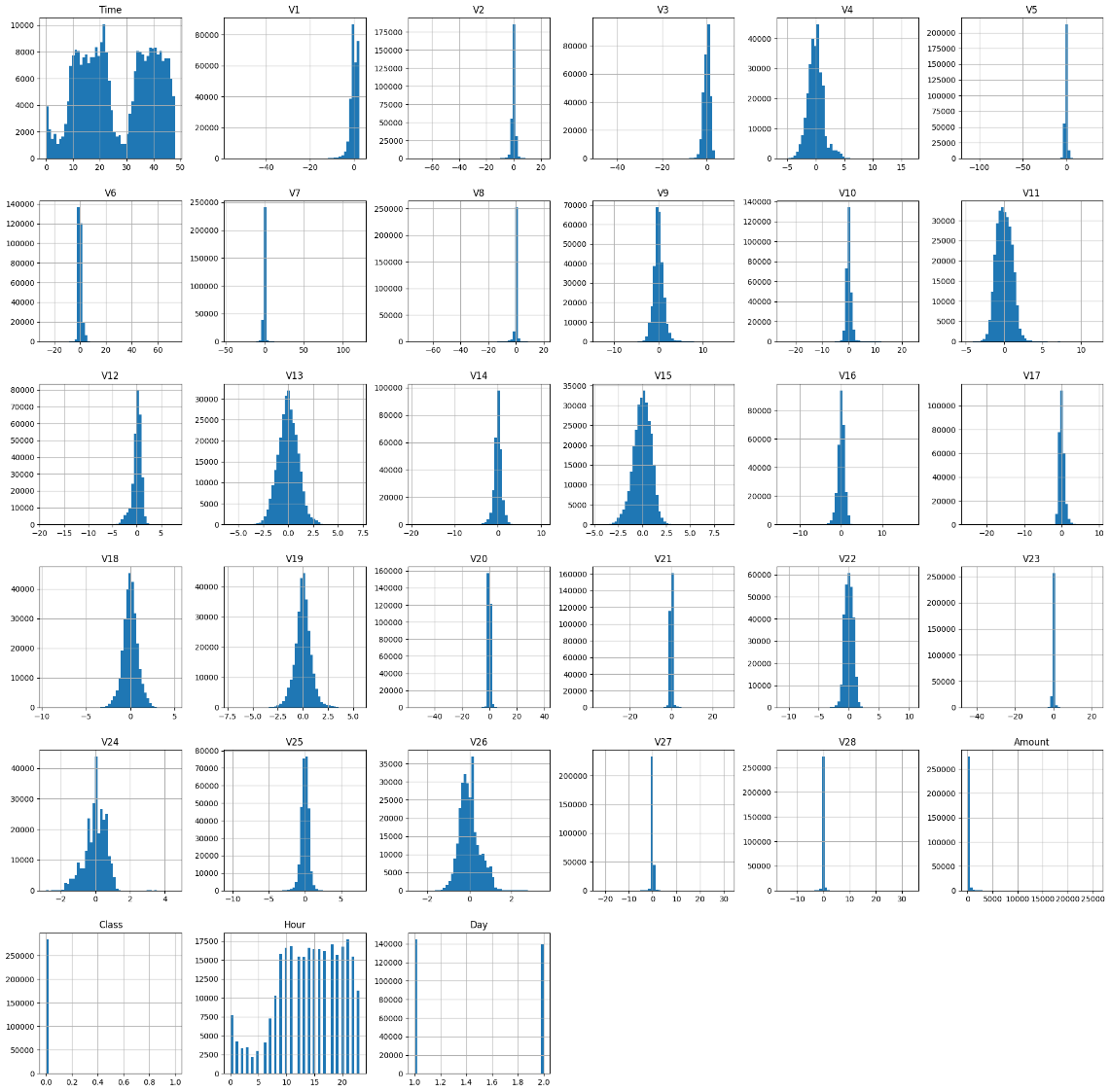
* The figure below visualizes the data imbalance



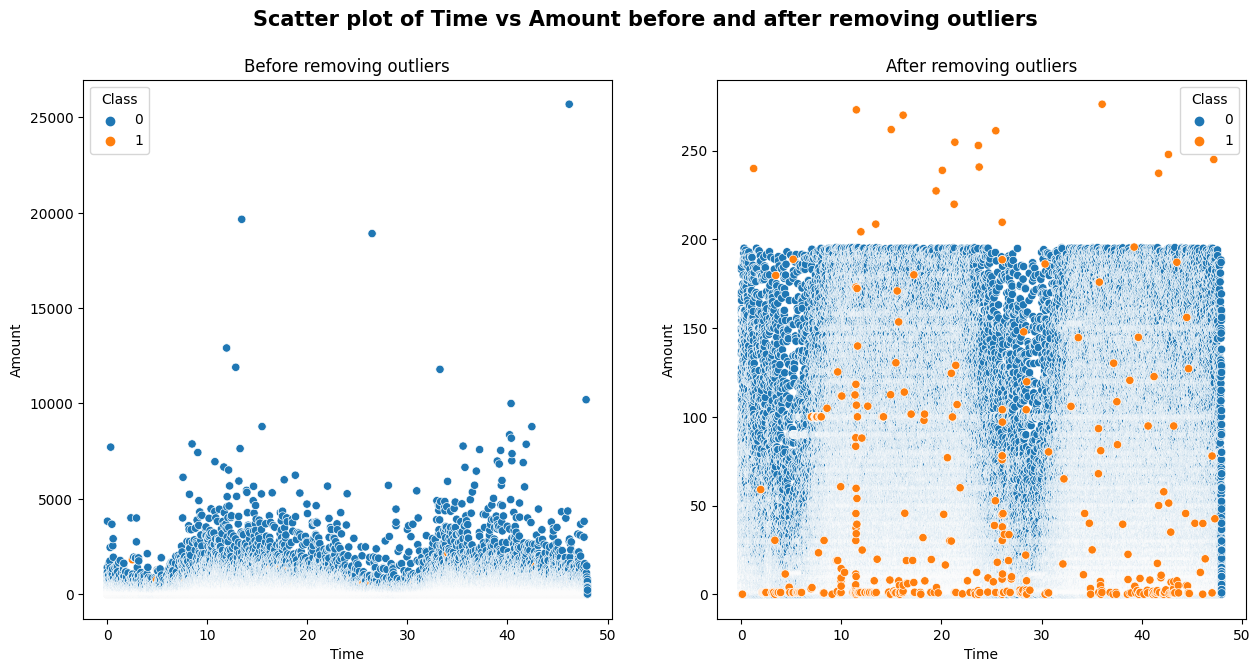
* The figure below shows the distribution of transactions with respect to 2 days (time feature was converted to hours using feature transformation). These show that around hours 0-5 and
* 25-30, which are the night times people are sleeping, fraudulent transactions still happen while non-fraudulent transactions barely happen in those hours.



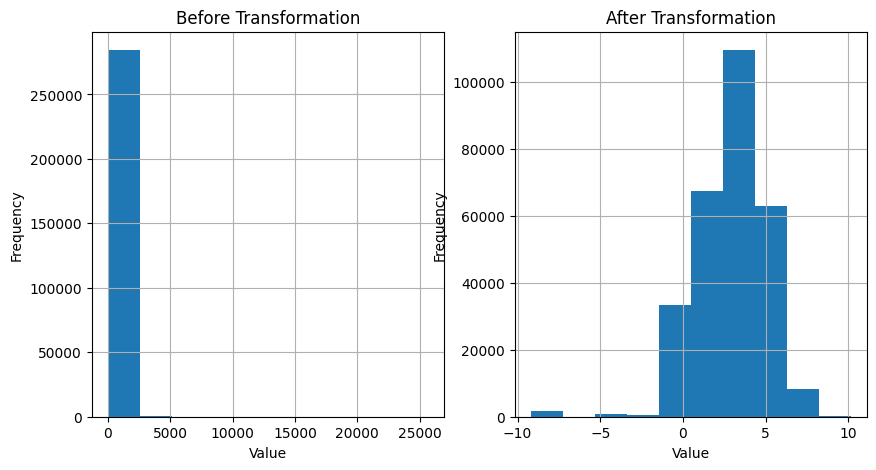
* The figure below shows the distribution of features, as shown, all features are normally distributed except the amount.



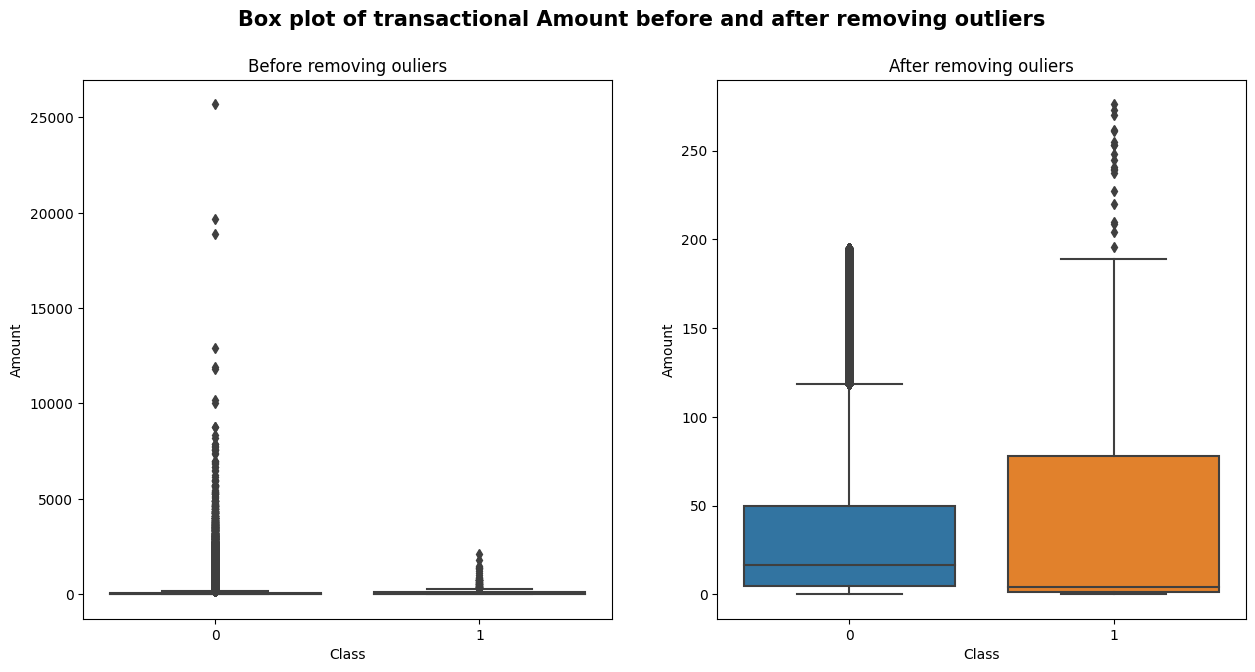
* The below figure shows the classes before and after outliers were removed using the amount feature since it was the feature with outliers.



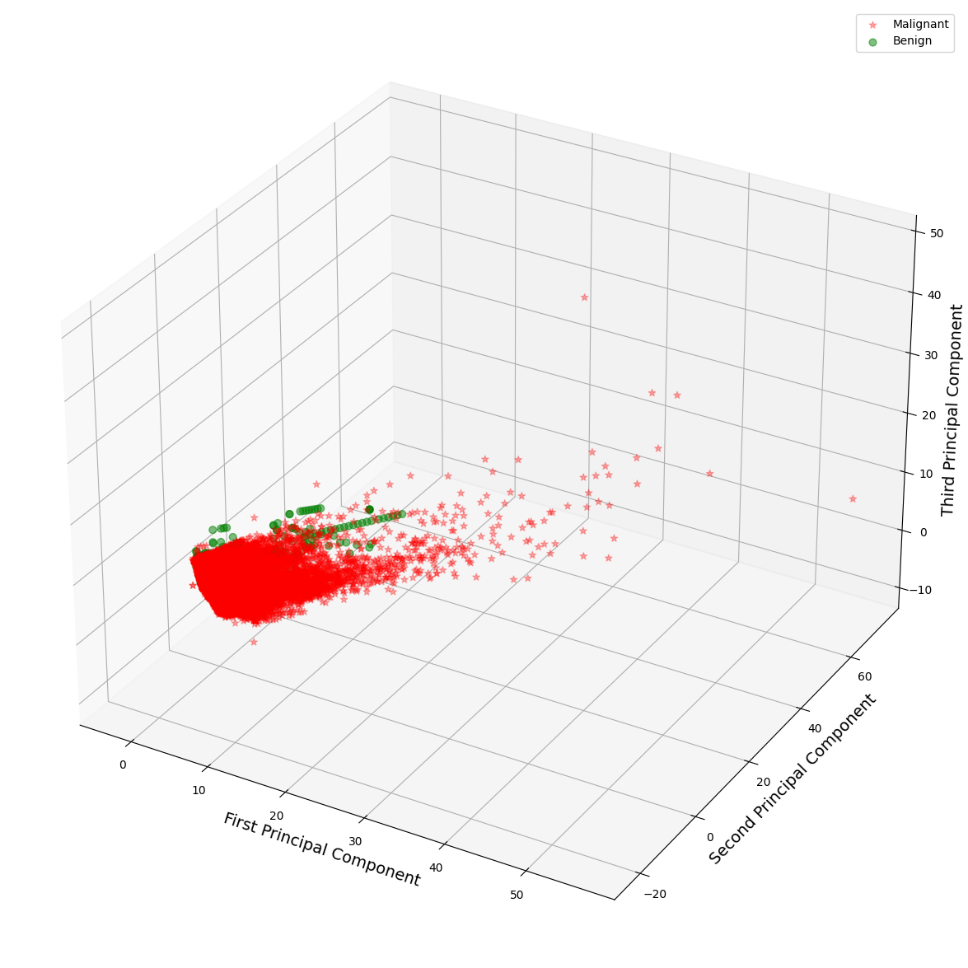
* The below figure shows the distribution of the amount before and after log-transform.



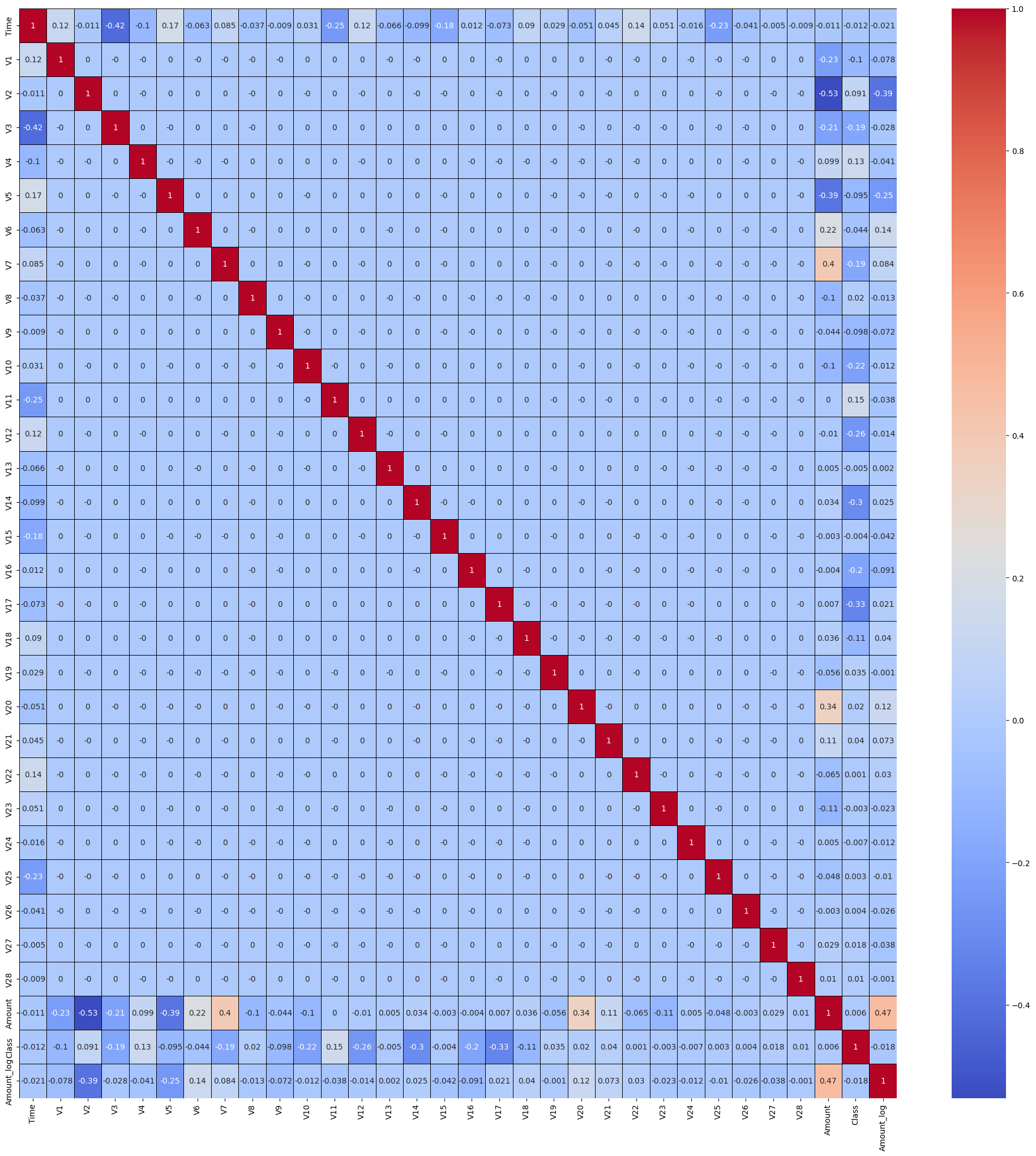
* The below figure shows the iqr of the amount feature before and after outliers were removed.



* The below figure shows the PCA for the visualization of classes.
* The below figure shows the classes before and after outliers were removed using the amount feature since it was the feature with outliers.



* The below figure shows the correlations, as you see, many variables are not correlated so no need to do feature removal.



* Below figure shows the most important features through correlation with classes.

