TEAM 8 PROJECT REPORT

TITLE: Credit Card Fraud Detection.

TEAM MEMBERS: Evan Chan, Farhan Rasheed Chughtai

1. Problem Statement/Motivation:

The primary driving force behind this project is to address a problem that virtually all banks worldwide are currently facing: how to identify and reduce fraudulent activity, particularly that involving credit cards. In addition to posing a threat to the security of banks and other financial institutions, this issue must be resolved to prevent significant revenue losses for them. Using the dataset for quick detection, we can train a model with supervised machine-learning techniques to identify fraudulent credit card transactions. This will address the problem of implementing a highly accurate program to detect fraudulent activity, which will in turn allow less revenue loss through efficiency and automation in financial systems. This will also lead to the mitigation of fraudulent activities.

2. Dataset/EDA:

The dataset we used for our project consists of transactions made by credit cards in September 2013 by European credit card holders. The transactions occurred in the span of two days, and we have 492 fraud cases out of 284,807 transactions. The dataset is highly unbalanced, with the positive class (frauds) accounting for only 0.172 percent of all the transactions. The dataset mainly contains numeric input variables, which are the result of a PCA transformation. Due to the confidentiality of the information, the feature names are hidden and are referred to as V1, V2, etc. The only known columns in the dataset are the time column and the amount column.

Next, we tried to visualize the data using the time column to see how the transactions looked over the course of a day, and we saw that most of the fraudulent transactions were taking place after midnight, from 1 a.m. to 7 a.m., with the rest happening close to noon, as seen from the diagram below (Figure 1). Furthermore, we did PCA and tried to visualize the data in two-dimensional space to better visualize the dataset and see if fraudulent transactions are separable or not from the non-fraudulent dataset, and we saw that a linear line could be used to separate them. The PCA diagram shown below (Figure 2) shows us just that, and in our model training and testing, you will see that linear models indeed performed well.

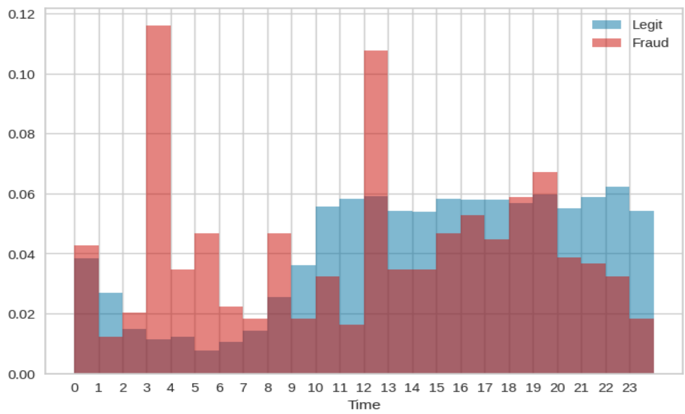

Figure 1 Visualization Over 24 hours

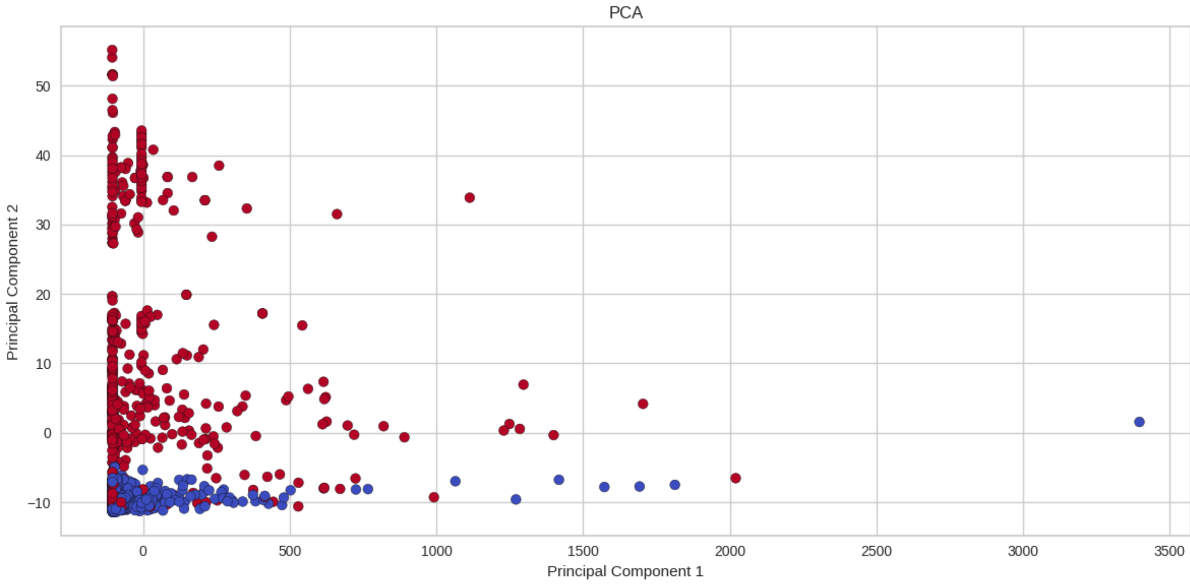

Figure 2 PCA

3. Methodology:

We used the following models in this project: Logistic Regression, Random Forest, Decision Trees, Bernoulli, LDA, CatBoost, and XG-Boost. Furthermore, we used SMOTE and undersampling to fix the balancing issue of the dataset and compare the results of all the models without any balancing and after balancing. For the first five models, we first validated our results using stratified K-folds of five folds, then took the top two performing models and got the results on the final test set. For xgboost and catboost, we computed their performance based on the test set using the same balancing techniques. For our metrics, we primarily focused on the precision and recall score of the fraud class. As this is an unbalanced dataset, we will not focus on the AUC or accuracy of the model. For more details, see the figure in the appendix.

4. Low Risk Goals and Results:

We started with the non boosting Models and got the results for the unbalanced , Smote and undersampled dataset you can see the validation set results below of all the three categories.

	Model	Class	Precision	Recall	F1-Score	Support
0	Logistic Regression	Fraud	0.875159	0.642097	0.739638	78.8
1	Logistic Regression	NotFraud	0.99938	0.999837	0.999609	45490.2
2	Random Forest	Fraud	0.937484	0.776599	0.848724	78.8
3	Random Forest	NotFraud	0.999613	0.999908	0.99976	45490.2
4	Linear Discriminant Analysis	Fraud	0.878622	0.761409	0.815061	78.8
5	Linear Discriminant Analysis	NotFraud	0.999587	0.999815	0.999701	45490.2
6	BernoulliNB	Fraud	0.841698	0.639598	0.72445	78.8
7	BernoulliNB	NotFraud	0.999376	0.999789	0.999582	45490.2
8	Decision Trees	Fraud	0.893215	0.748718	0.81386	78.8
9	Decision Trees	NotFraud	0.999565	0.999842	0.999703	45490.2

Figure 3 Unbalanced Result

	Model	Class	Precision	Recall	F1-Score	Support
0	Logistic Regression	Fraud	0.057175	0.903635	0.107515	78.8
1	Logistic Regression	NotFraud	0.999829	0.9741	0.986796	45490.2
2	Random Forest	Fraud	0.877401	0.789354	0.829707	78.8
3	Random Forest	NotFraud	0.999635	0.999807	0.999721	45490.2
4	Linear Discriminant Analysis	Fraud	0.093738	0.81493	0.168068	78.8
5	Linear Discriminant Analysis	NotFraud	0.999675	0.986252	0.992918	45490.2
6	BernouliNB	Fraud	0.157516	0.812269	0.262948	78.8
7	BernouliNB	NotFraud	0.999672	0.992315	0.99598	45490.2
8	Decision Trees	Fraud	0.350694	0.763973	0.479581	78.8
9	Decision Trees	NotFraud	0.99959	0.997525	0.998556	45490.2

Figure 4 Smote Results

	Model	Class	Precision	Recall	F1-Score	Support
0	Logistic Regression	Fraud	0.029879	0.921422	0.057871	78.8
1	Logistic Regression	NotFraud	0.999856	0.947945	0.973205	45490.2
2	Random Forest	Fraud	0.053596	0.903668	0.101069	78.8
3	Random Forest	NotFraud	0.999828	0.971814	0.985617	45490.2
4	Linear Discriminant Analysis	Fraud	0.06897	0.842843	0.127487	78.8
5	Linear Discriminant Analysis	NotFraud	0.999722	0.980211	0.98987	45490.2
6	BernouliNB	Fraud	0.132204	0.814833	0.226857	78.8
7	BernouliNB	NotFraud	0.999676	0.990556	0.995095	45490.2
8	Decision Trees	Fraud	0.015941	0.929049	0.031324	78.8
9	Decision Trees	NotFraud	0.999863	0.897477	0.945824	45490.2

Figure 5 Under Sample Results

Looking at the initial results of our models, we could clearly see that SMOTE and under sampling increase the recall score, but the precision score really suffers, so balancing the dataset was a failure, so we decided to stick with not balancing the dataset and check the performance of our two best-performing models on the final test set, and you can see the results below:

	Model	Class	Precision	Recall	F1-Score	Support
0	Random Forest	Fraud	0.940476	0.806122	0.868132	98
1	Random Forest	NotFraud	0.999666	0.999912	0.999789	56864
2	Linear Discriminant Analysis	Fraud	0.822917	0.806122	0.814433	98
3	Linear Discriminant Analysis	NotFraud	0.999666	0.999701	0.999683	56864

Figure 6 Top two Performing Model Results on Test Set

Further results like Precision recall curves and Confusion matrix can be seen in the Appendix.

5. Medium Risk Goals and Results:

For our Medium Level goals, we decided to purse the boosting algorithms and again used the same methodology. Below Figures show us the performance of the models using SMOTE, underdamping and no balancing. For More results, please refer to the appendix.

	Model	Class	Precision	Recall	F1-Score	Support
0	X-gBoost	Fraud	0.135385	0.897959	0.235294	98
1	X-gBoost	NotFraud	0.999822	0.990117	0.994946	56864
2	CatBoost	Fraud	0.615942	0.867347	0.720339	98
3	CatBoost	NotFraud	0.999771	0.999068	0.999419	56864

Figure 7 Smote Results on Test Set

	Model	Class	Precision	Recall	F1-Score	Support
0	X-gBoost	Fraud	0.035629	0.908163	0.068567	98
1	X-gBoost	NotFraud	0.999835	0.957636	0.97828	56864
2	CatBoost	Fraud	0.071016	0.877551	0.131398	98
3	CatBoost	NotFraud	0.999785	0.980216	0.989904	56864

Figure 8 Under Sampled Results

	Model	Class	Precision	Recall	F1-Score	Support
0	X-gBoost	Fraud	0.918605	0.806122	0.858696	98
1	X-gBoost	NotFraud	0.999666	0.999877	0.999771	56864
2	CatBoost	Fraud	0.964706	0.836735	0.896175	98
3	CatBoost	NotFraud	0.999719	0.999947	0.999833	56864

Figure 9 Unbalanced Dataset Results

From the above results, we can again see that SMOTE and under sampling performed poorly, and we got the best results on the unbalanced dataset. Finally, we were able to achieve a recall score of 0.96 and a precision score of 0.83 with the Cat Boost Model, which is quite good performance and will help us achieve the goal we set out to do in this project, to solve the problem statement.

6. High Risk Goals and Results:

For our high-risk goals, we wanted to suggest preventions for the companies to adopt to mitigate fraudulent transactions and identify key columns integral to the detection of these transactions. From our best-performing models, we saw that V4, V26, and V14 are three key columns that help identify fraudulent transactions and will help in mitigating the issue. So, our suggestion would be to keep a close eye on these features due to the confidentiality nature of the dataset. We don't know what these columns are, but these three columns are integral to stopping fraud in their systems and securing their systems from other fraudulent activities. Refer to the appendix for more details.

7. References:

- https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data the dataset used.
- Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson and Gianluca Bontempi. Calibrating Probability with Undersampling for Unbalanced Classification. In Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2015.
- https://catboost.ai/en/docs/features/feature-importances-calculation
- https://xgboost.readthedocs.io/en/stable/

8. Appendix:

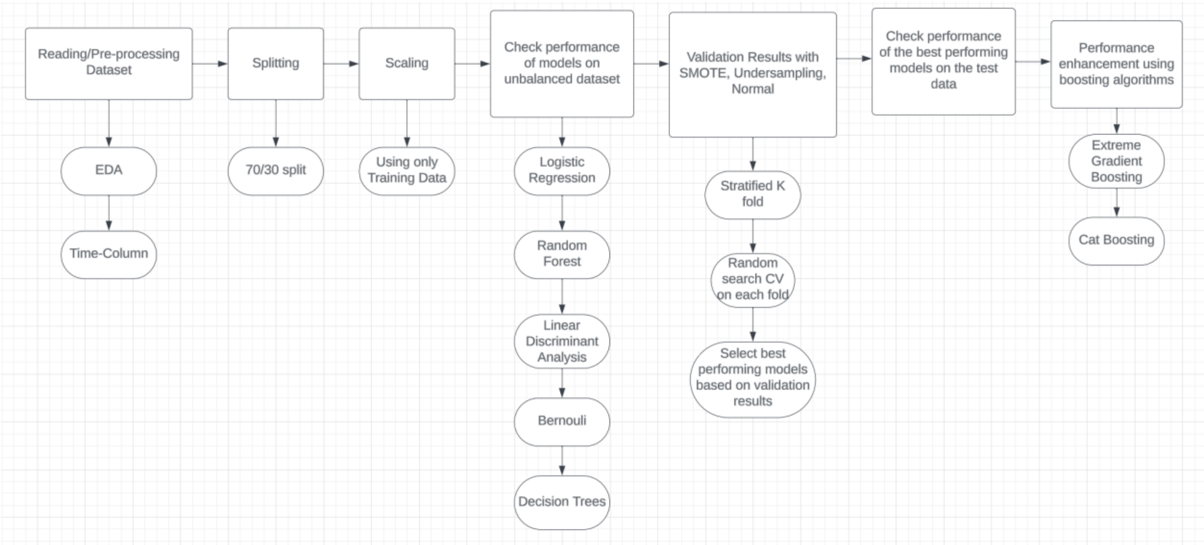

Figure 10 Methodology Diagram

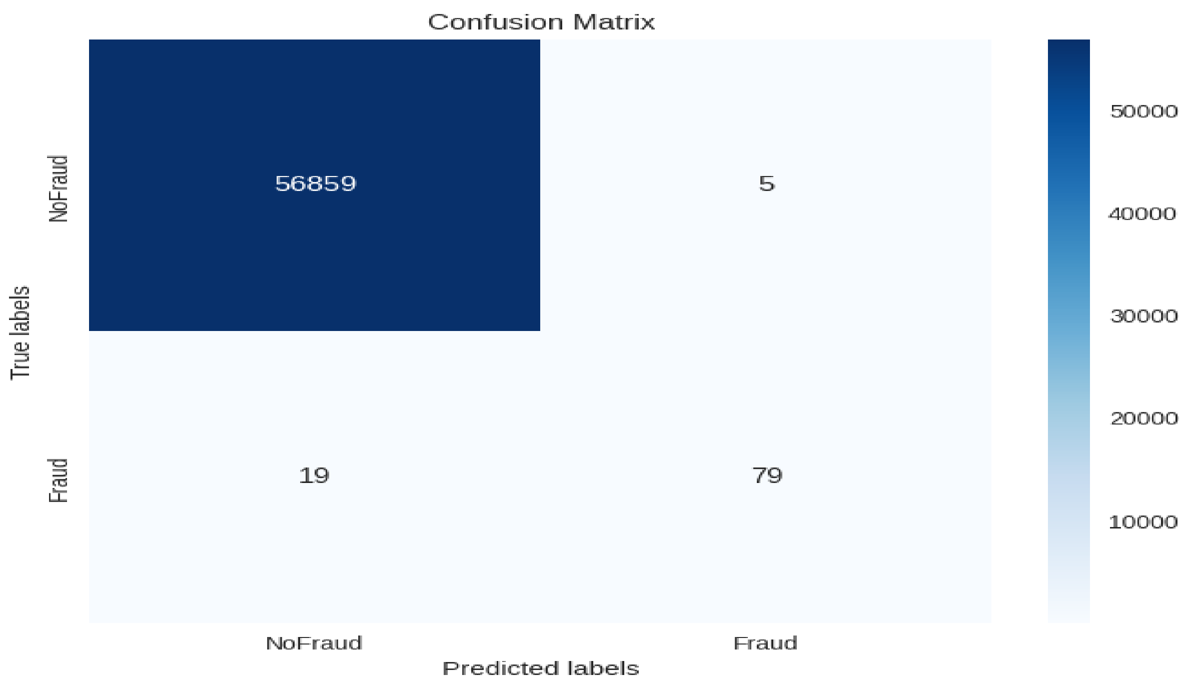

Figure 11 Confusion Matrix Random Forest on Final Test Set

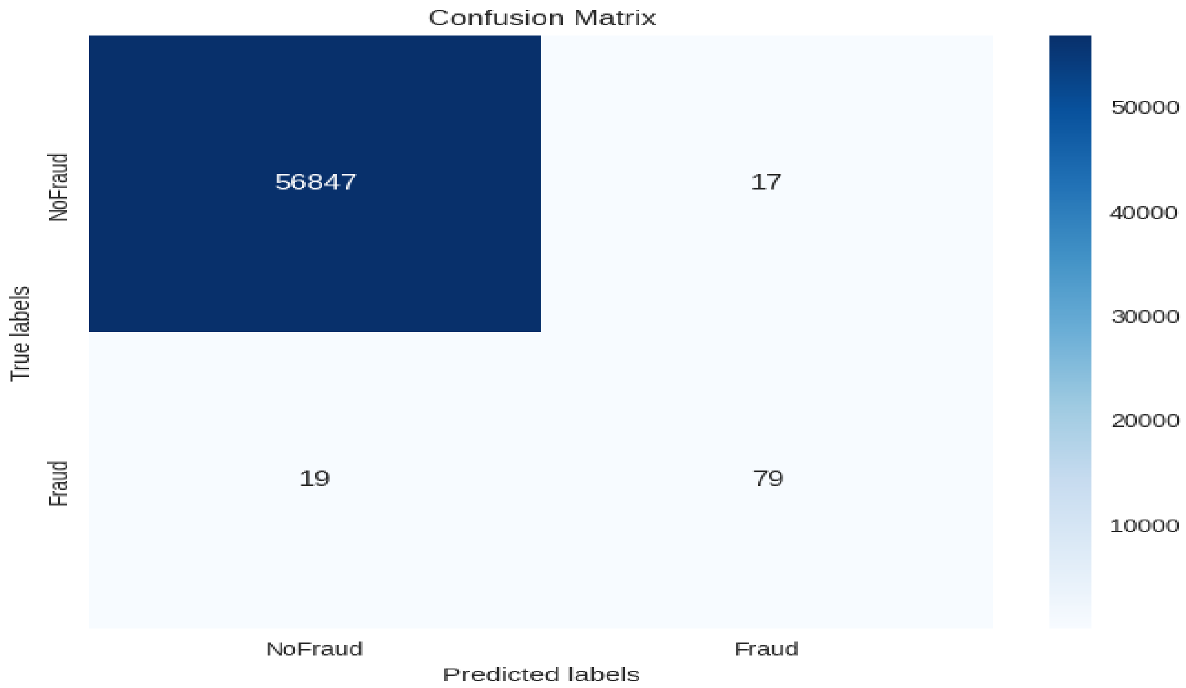

Figure 12 Confusion Matrix LDA on final Test Set

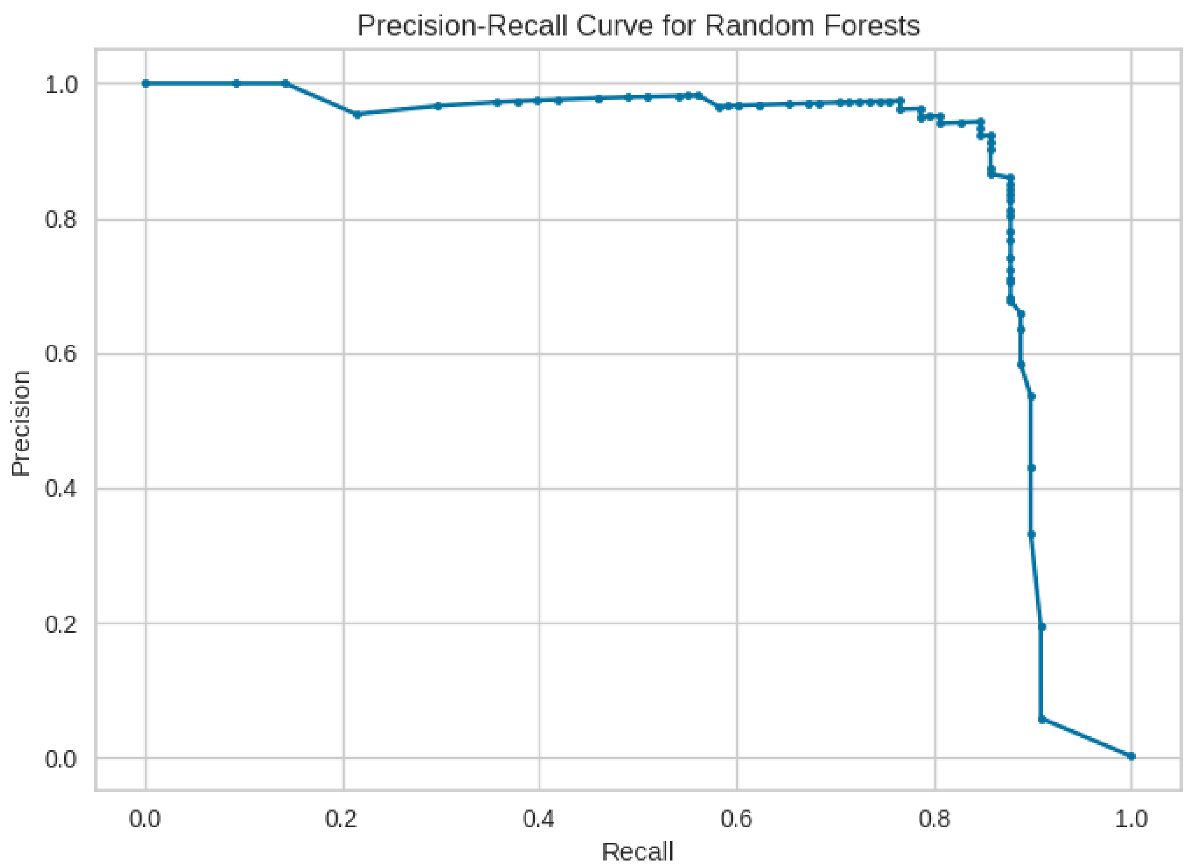

Figure 13 PR curve for Random Forest Final Test Data

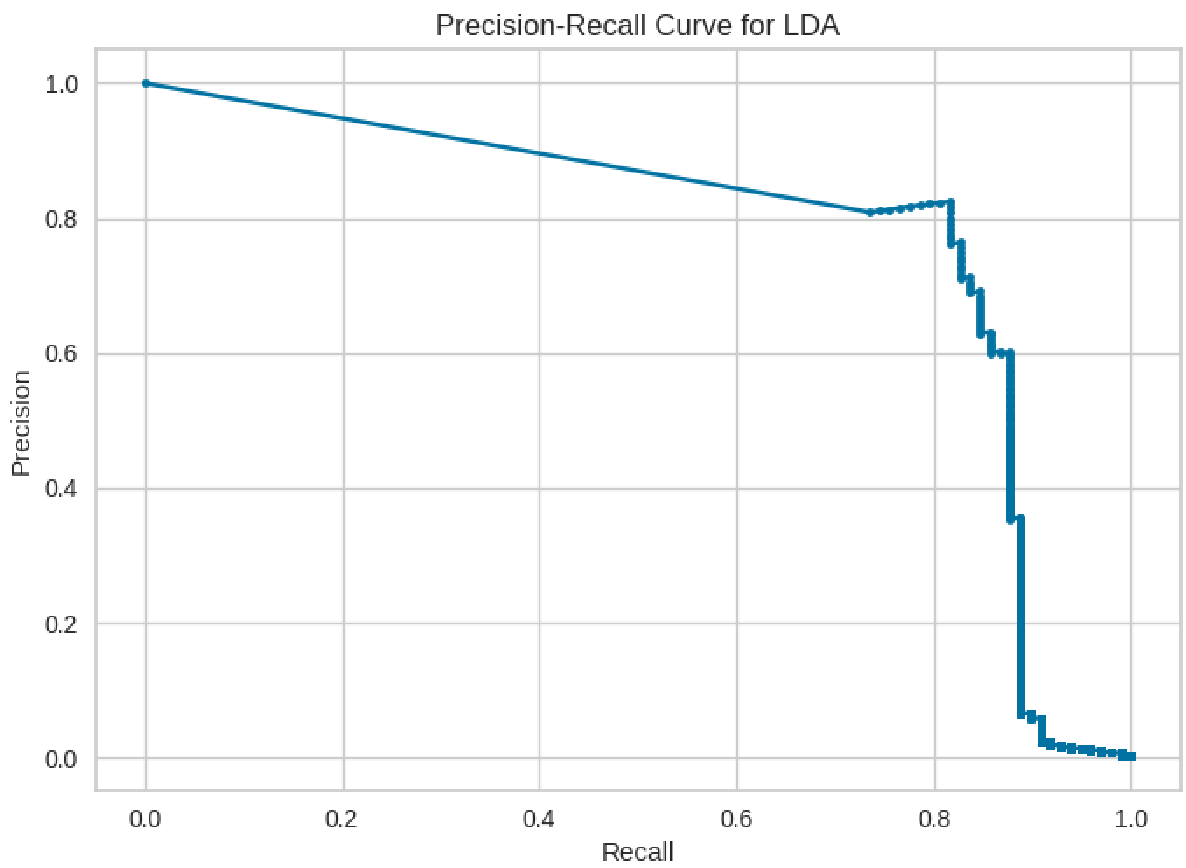

Figure 14 PR curve for LDA on Final Test Data

	Model	Class	Precision	Recall	F1-Score	Support
0	Random Forest	Fraud	0.860215	0.816327	0.837696	98
1	Random Forest	NotFraud	0.999683	0.999771	0.999727	56864
2	Decision Trees	Fraud	0.377551	0.755102	0.503401	98
3	Decision Trees	NotFraud	0.999577	0.997855	0.998715	56864

Figure 15 Final Test Set Results using SMOTE

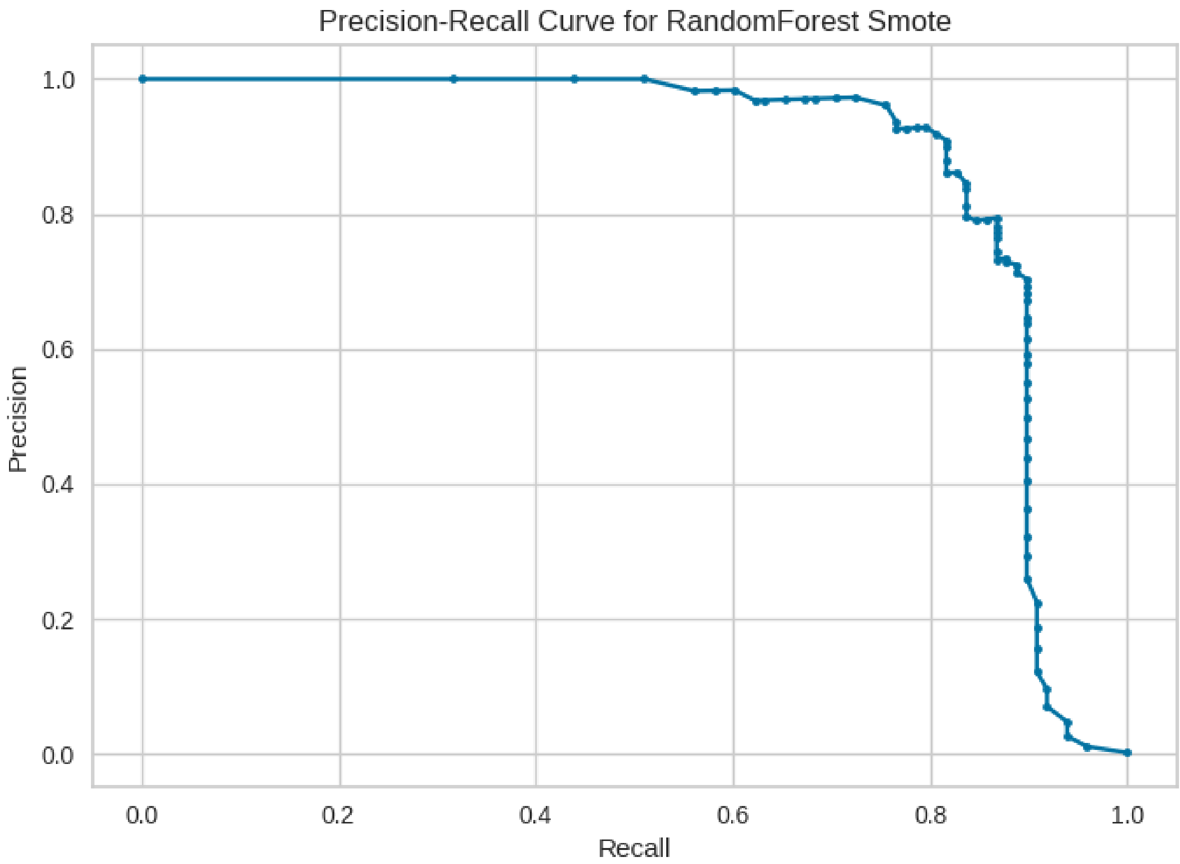

Figure 16 PR curve using SMOTE Random Forests

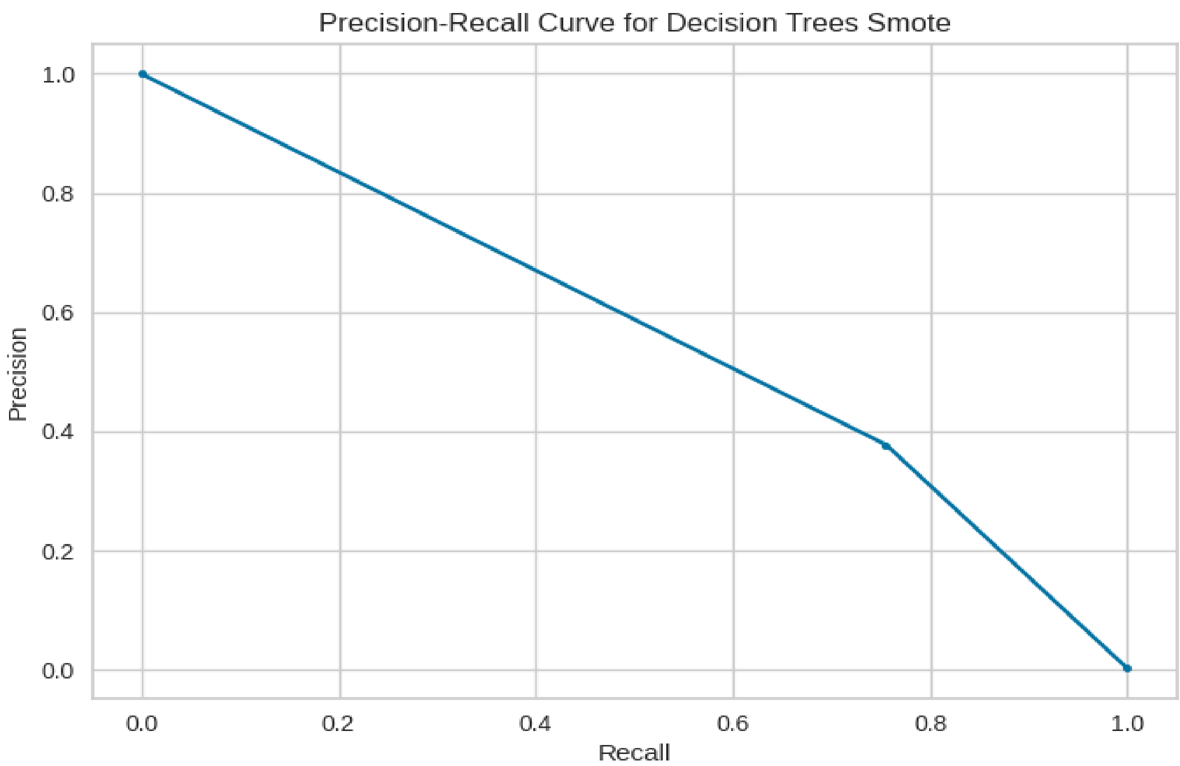

Figure 17 PR curve on Final test set Decision Trees

	Model	Class	Precision	Recall	F1-Score	Support
0	Linear Discriminant Analysis	Fraud	0.084016	0.836735	0.1527	98
1	Linear Discriminant Analysis	NotFraud	0.999714	0.984278	0.991936	56864
2	BernouliNB	Fraud	0.155009	0.836735	0.261563	98
3	BernouliNB	NotFraud	0.999716	0.992139	0.995913	56864

Figure 18 Under Sampling Final Results on Test Set

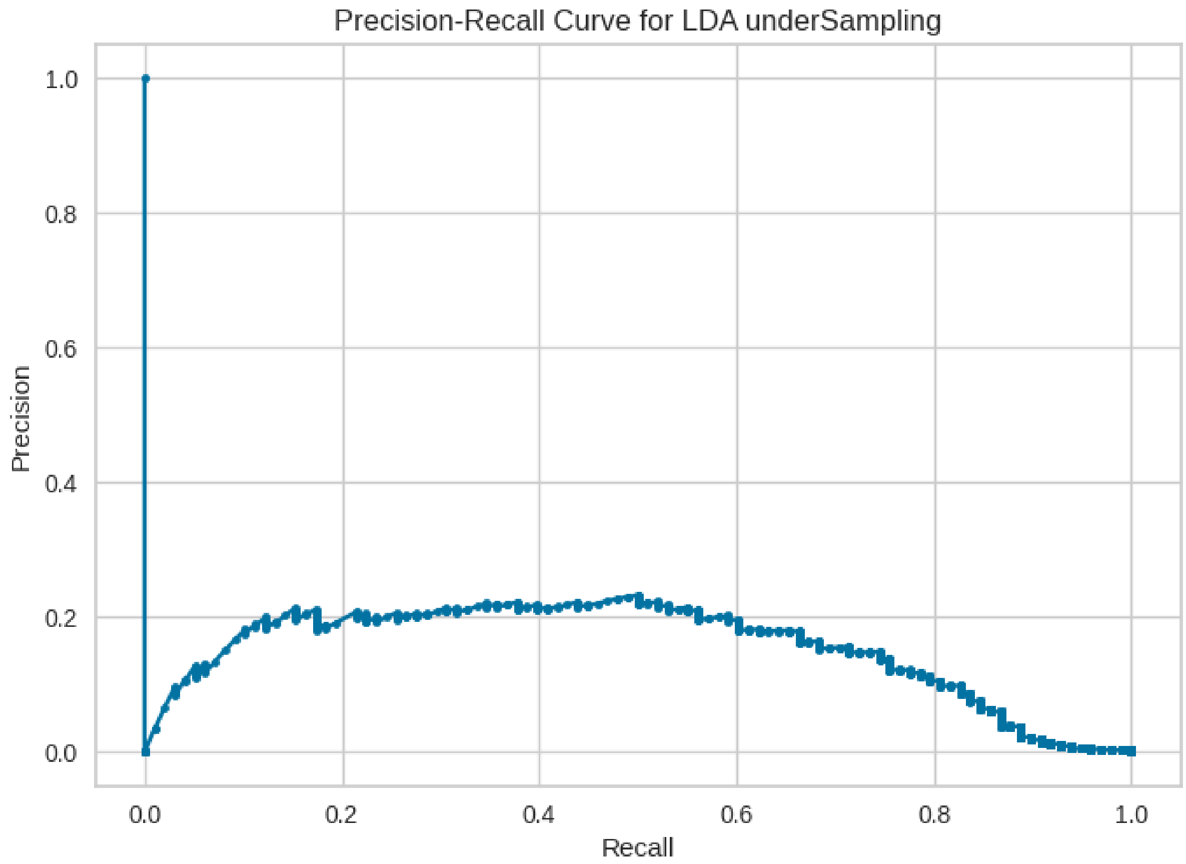

Figure 19 PR curve for LDA using Under Sampling on Test Set

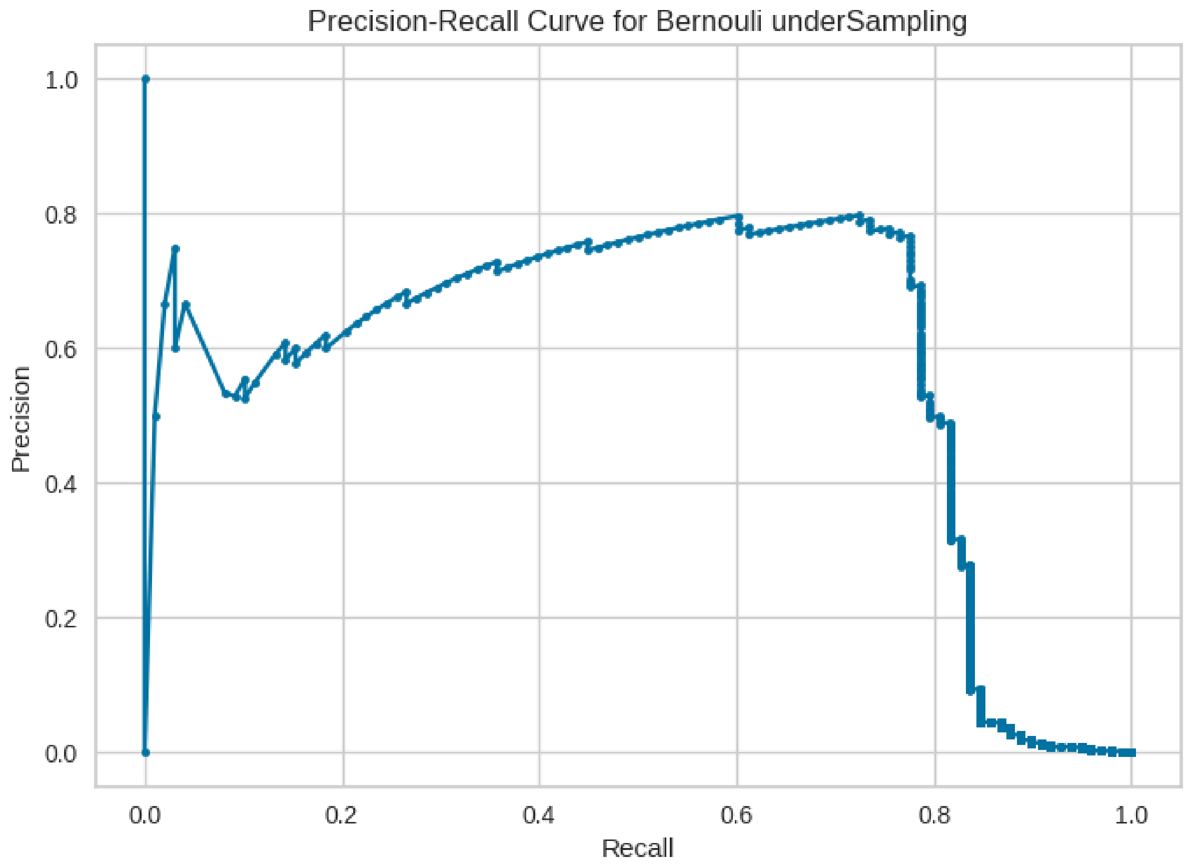

Figure 20 PR curve for Bernoulli Under Sampling on Final Test set

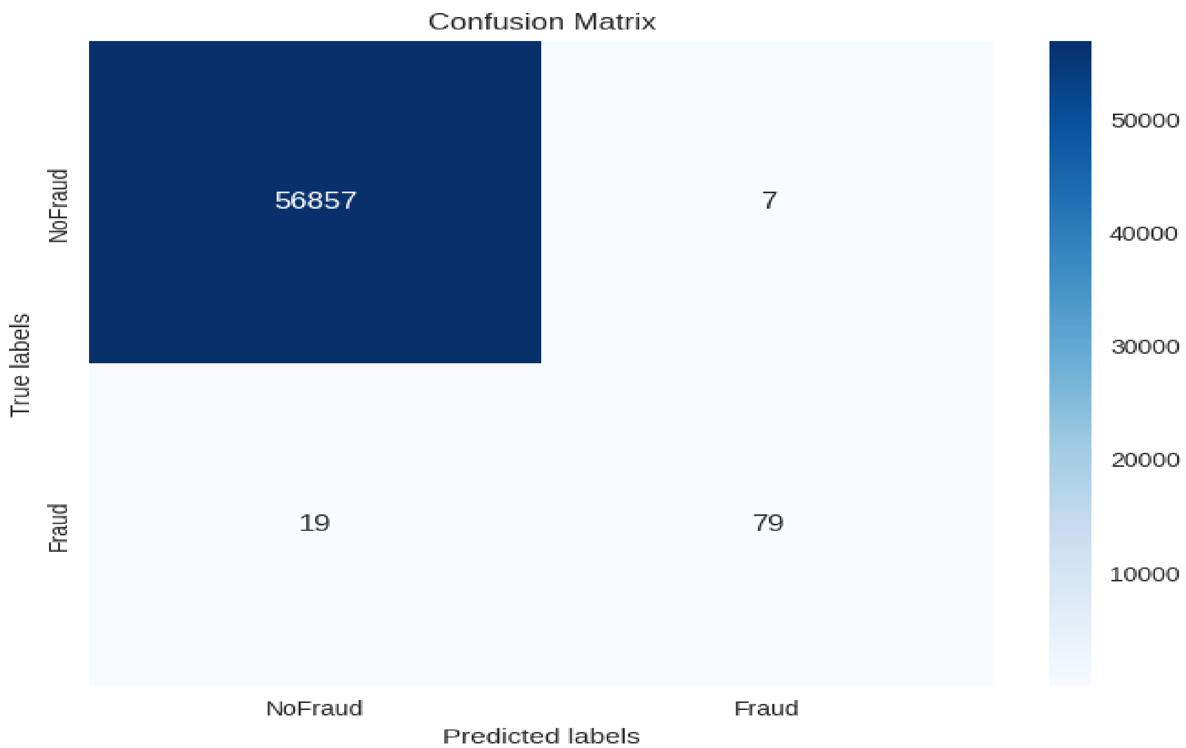

Figure 21 Confusion Matrix Xgboost Unbalanced Dataset on Final Test Set

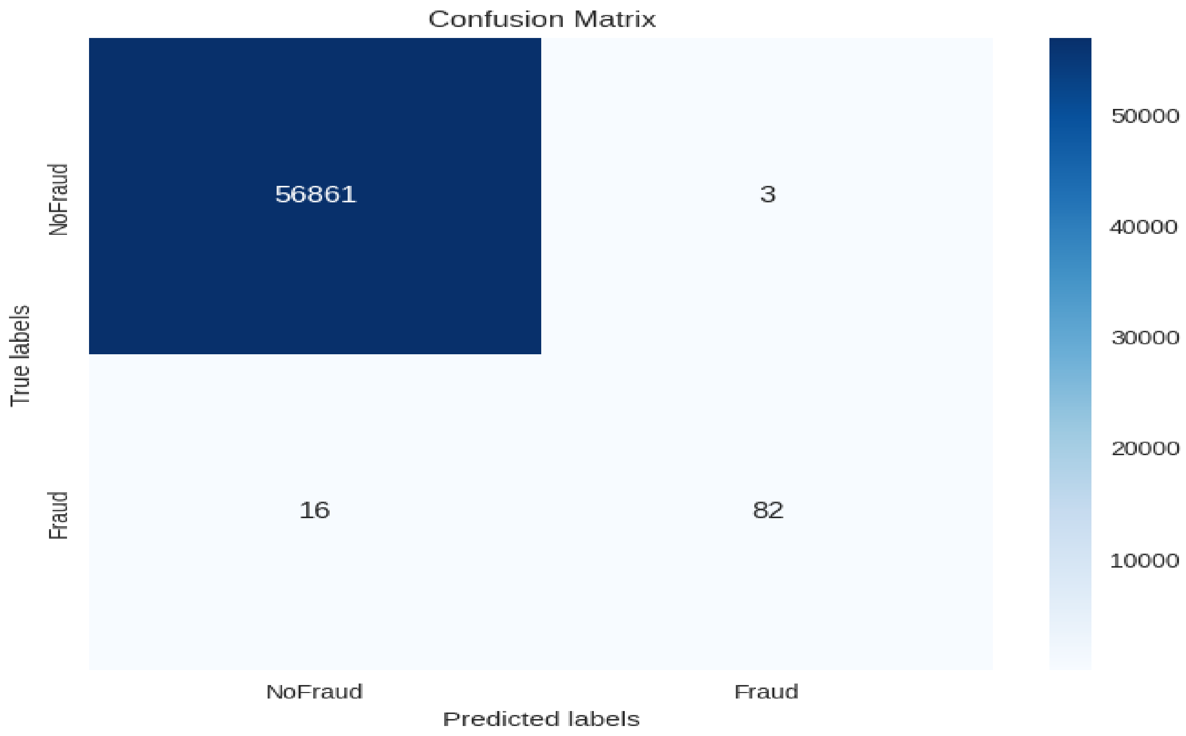

Figure 22 Confusion Matrix CatBoost on Final DataSet Unbalanced.

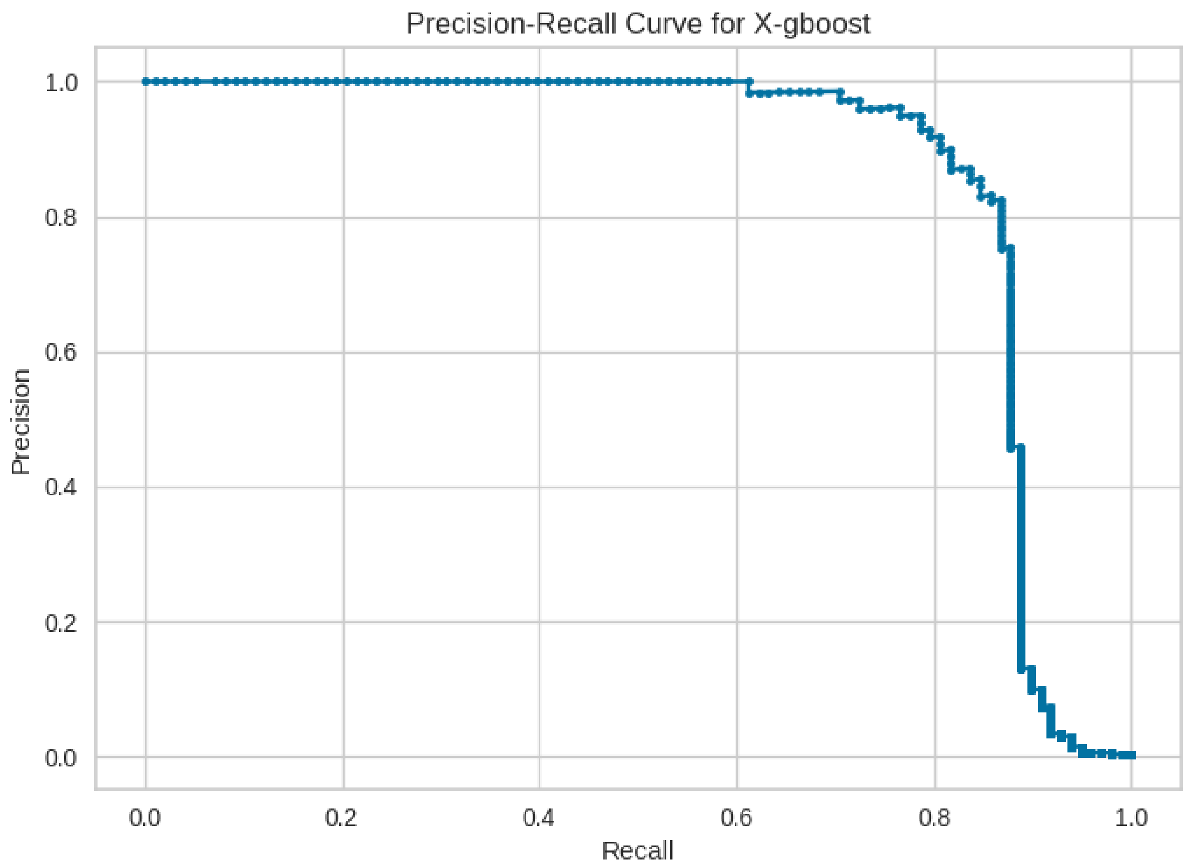

Figure 23 PR curve for XGboost Unbalanced Dataset on Test Set

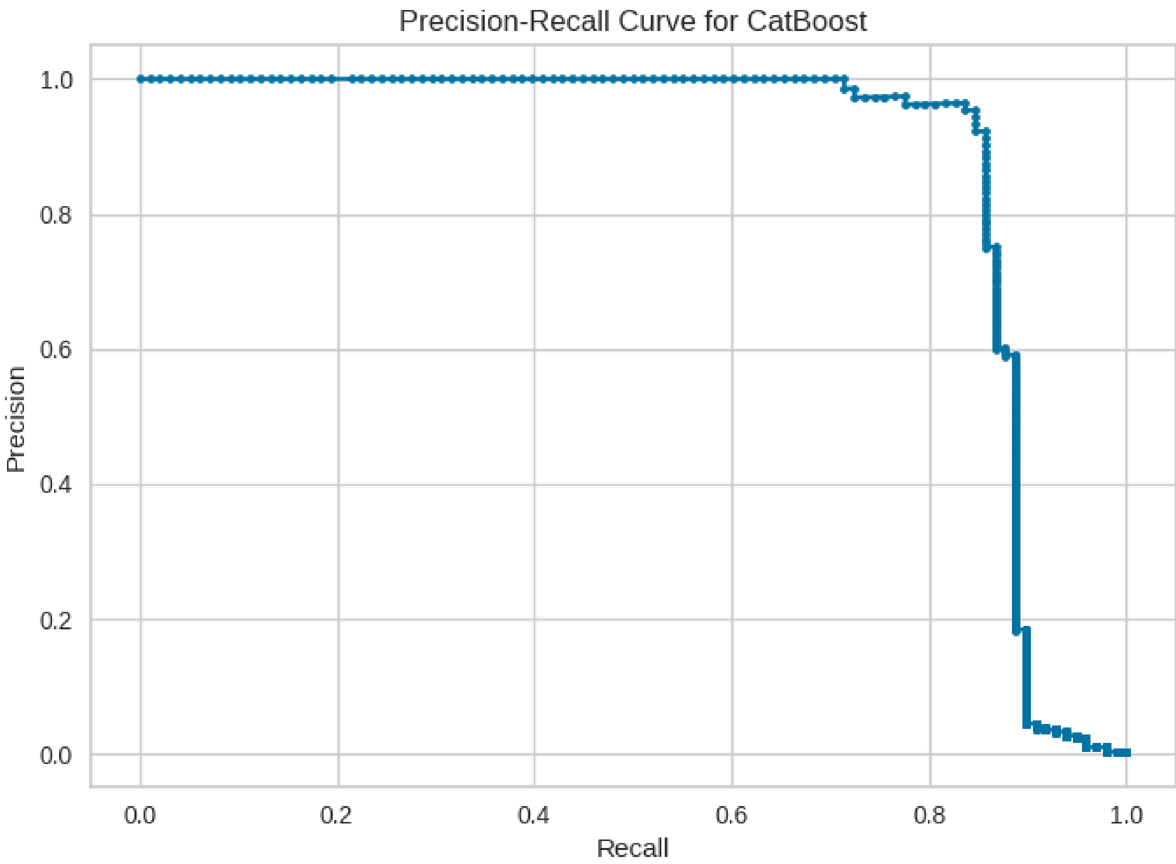

Figure 24 PR curve for CatBoost unbalanced Dataset on Final Test Set

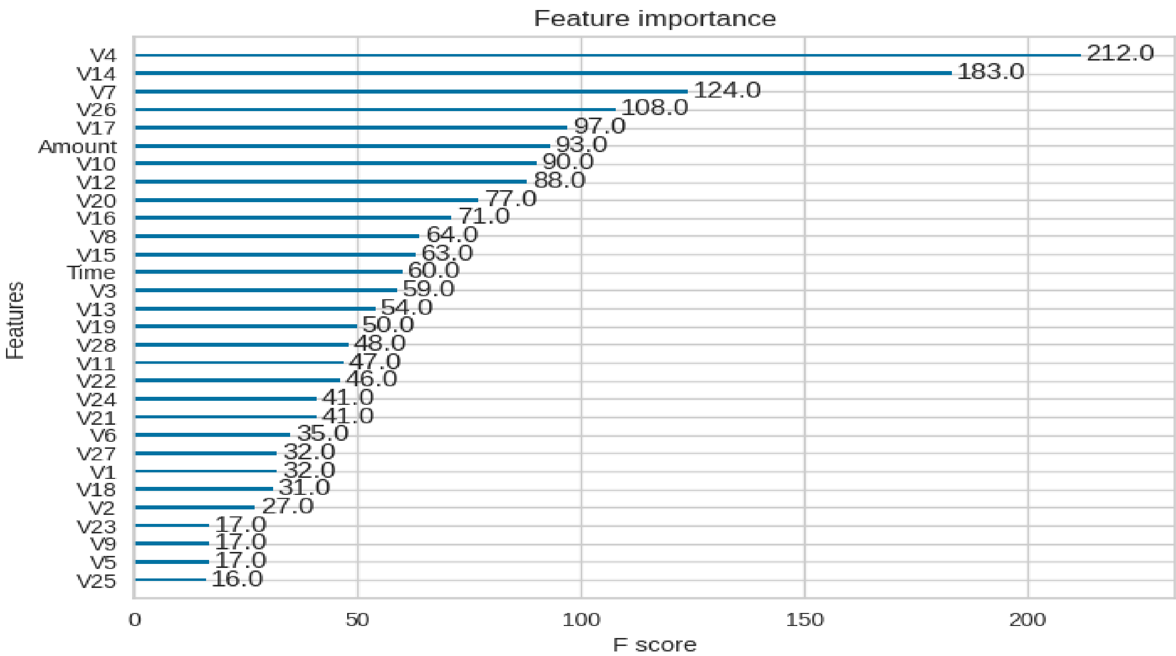

Figure 25 Feature Importance XgBoost

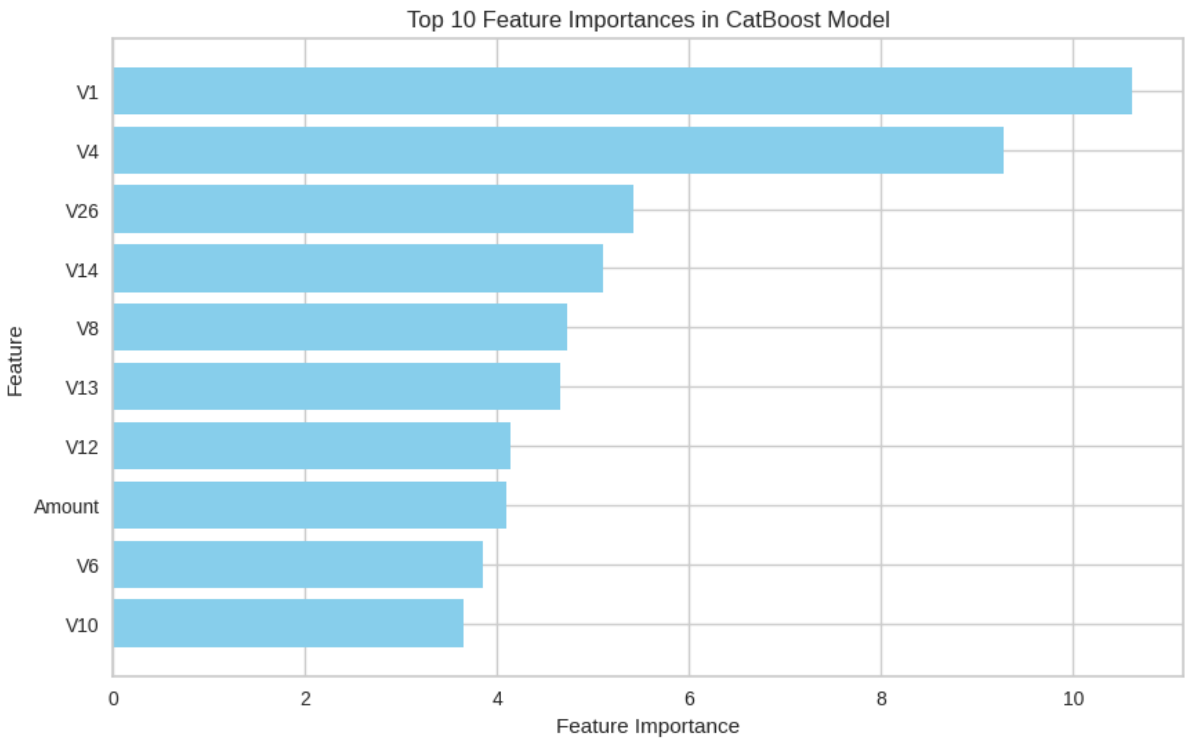

Figure 26 Feature Importance CatBoost