CREDIT CARD FRAUD DETECTION

# Introduction

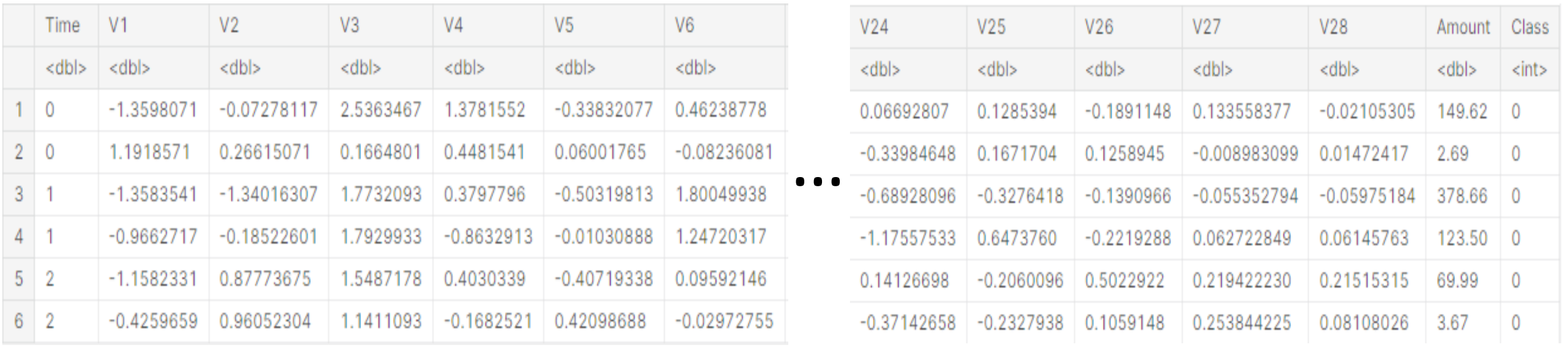
- The project we chose is called Credit Cart Fraud Detection because in these days, credit card fraud is a significant issue in the financial sector. A yearly loss of millions of money results from fraudulent card transactions.
- The aim of this project is to implement strategies that can successfully predict fraudulent transactions as given in our data, work in harmony with the sampling and modeling techniques most suitable for our data due to the imbalance of our data set and give the closest approach to the truth.

# Methodology

- While developing the project, we started with the models that we thought were simpler, and continued with the models that looked more complex and that we thought might yield better results.
- We started out with Decision Tree model. Then we tried Logistic Regression, Random Forest and then we also tried XG Boost, which is based on Gradient Boosted Trees.

# Data Set

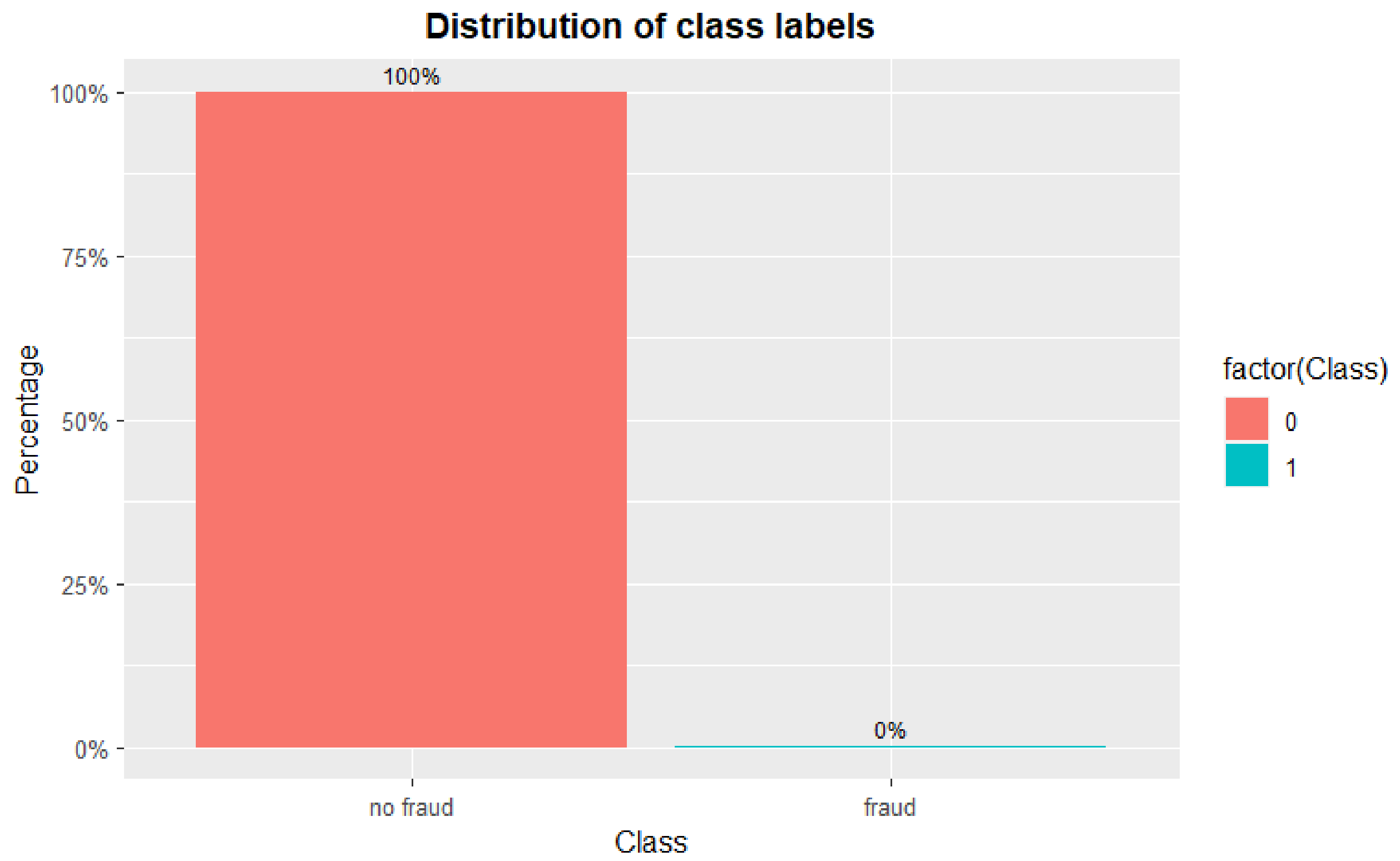
- We got our dataset from Kaggle and imported it. Due to the privacy and security of the data, we cannot provide detailed information on what the features are, but what we do know is that due to the PCA transformation, the features only contain numeric input variables. Our features are Time, Amount, Class and features from V1 to V28.
- -Time: Expresses the time in seconds between two transactions.
- -Amount: refers to the transaction amount.
- -Class: Response variable and takes the value 1 for Fraud and 0 for Non-Fraud.



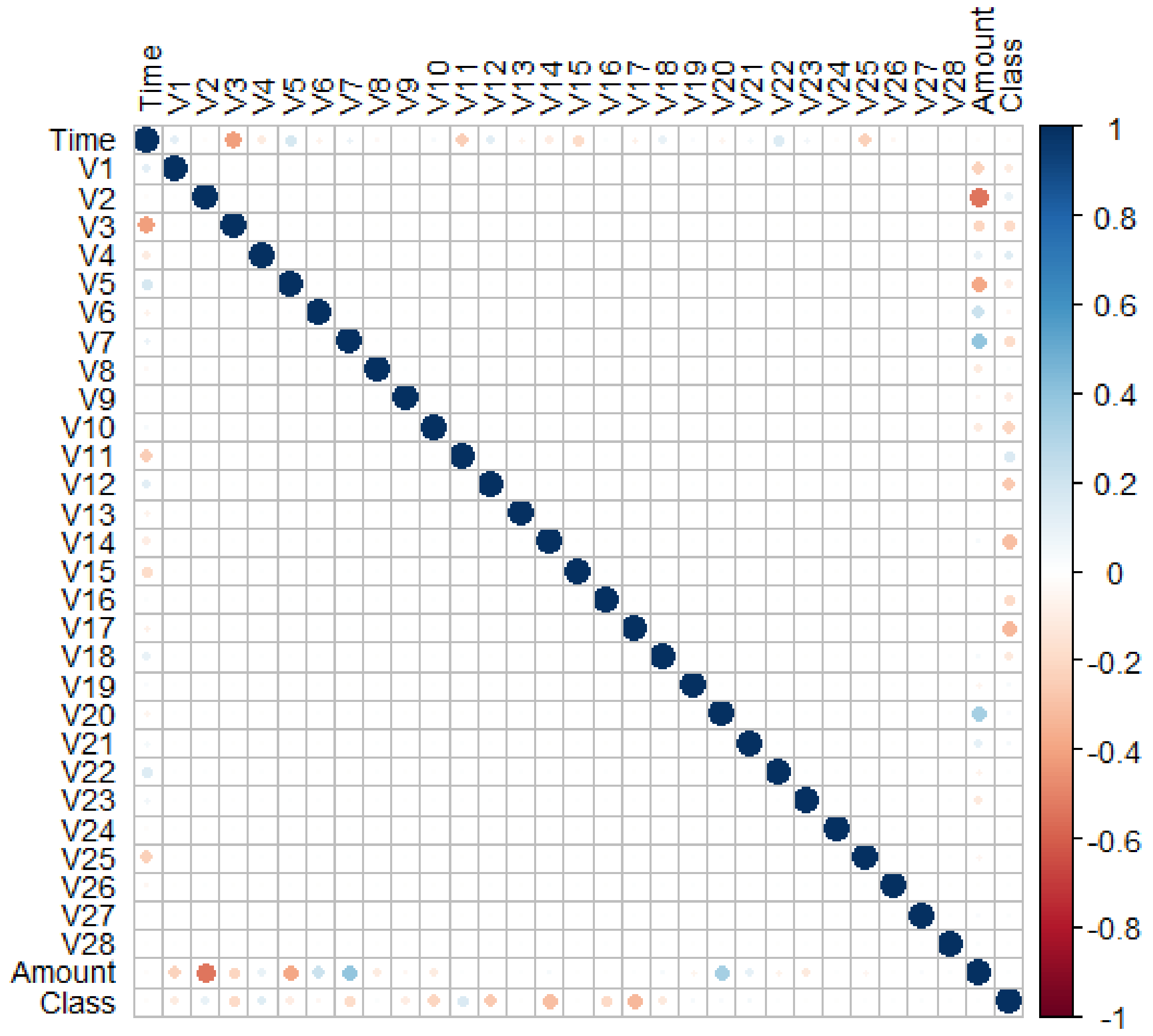
	Time	V1	V2	V3	V4	V5	V6
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	0	-1.3598071	-0.07278117	2.5363467	1.3781552	-0.33832077	0.46238778
2	0	1.1918571	0.26615071	0.1664801	0.4481541	0.06001765	-0.08236081
3	1	-1.3583541	-1.34016307	1.7732093	0.3797796	-0.50319813	1.80049938
4	1	-0.9662717	-0.18522601	1.7929933	-0.8632913	-0.01030888	1.24720317
5	2	-1.1582331	0.87773675	1.5487178	0.4030339	-0.40719338	0.09592146
6	2	-0.4259659	0.96052304	1.1411093	-0.1682521	0.42098688	-0.02972755

...

V24	V25	V26	V27	V28	Amount	Class
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<int>
0.06692807	0.1285394	-0.1891148	0.133558377	-0.02105305	149.62	0
-0.33984648	0.1671704	0.1258945	-0.008983099	0.01472417	2.69	0
-0.68928096	-0.3276418	-0.1390966	-0.055352794	-0.05975184	378.66	0
-1.17557533	0.6473760	-0.2219288	0.062722849	0.06145763	123.50	0
0.14126698	-0.2060096	0.5022922	0.219422230	0.21515315	69.99	0
-0.37142658	-0.2327938	0.1059148	0.253844225	0.08108026	3.67	0



By looking at the table we created above, we can clearly see how big the difference is between the two classes of data (no fraud 0 and fraud 1) and how unevenly the dataset is distributed between these two classes. Even if the data is not complete, we can understand that almost 100% of the data belongs to non-fraud transactions. An accuracy approach that sees non fraud, that is, class=0 as having an accuracy close to 100%, will not be a correct practice as it will create insensitivity to false positives here.



As we mentioned before, since our features are confidential, the relationship of these features with each other, that is, the knowledge of how they correlate with each other, has become important for us. So we looked at the correlation of V1-V28s, Time and Amount properties. We concluded that all these features are not very related to each other.

# Sampling Techniques

## Down-Sampling

This method helps to reduce the number of observations of the class that has the majority and to balanced the data set.

## Up-Sampling

This method helps to make a trade-off by replicates minority-class observations. It works with a logic similar to the down-sampling method.

## ROSE (random over-sampling examples)

Instead of replicating and adding the observations from the minority class, it overcome imbalances by generates artificial data. It is also a type of oversampling technique. It uses smoothed bootstrapping to draw artificial samples from the feature space neighbourhood around the minority class.

```
              Not-Fraud  Fraud
Original         227452    394
Up-sampling      227452 227452
Down-sampling       394    394
Rose-sampling    114081 113765
```

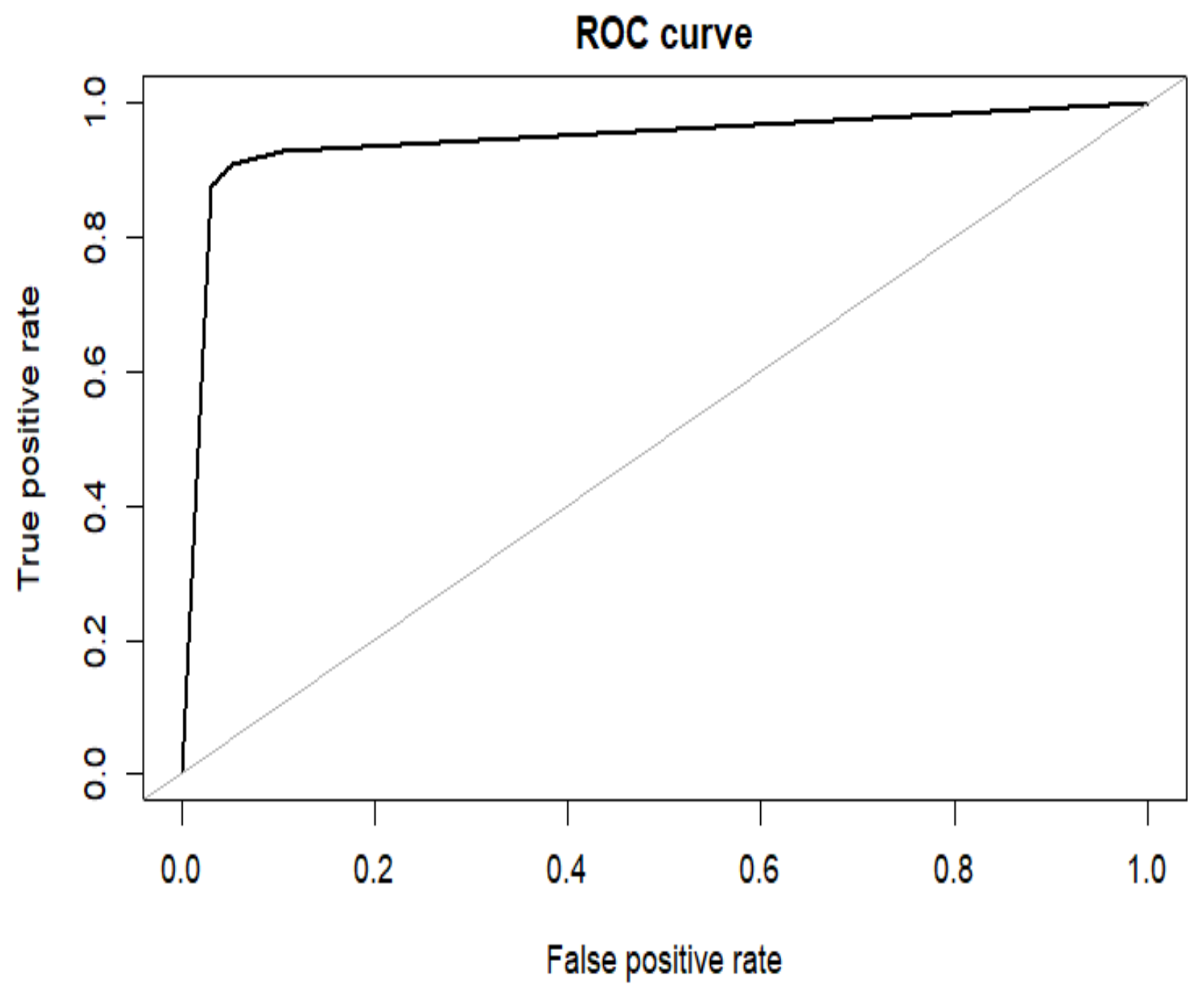
# Models

While evaluating the binary classification algorithm, we used the receiver operating characteristic (ROC) curve, so it would be easier to visually understand the performance of the classifier. As you can see from the graphs below, being closest to the True Positive line actually represents an almost perfect classifier for us. In other words, what we are looking for in this system is to find the ratio with the highest true positive and the lowest false positive ratio. In our result we tried a lot of example with using original data, down-sampling data, up-sampling data and rose sampling data and with this sampling methods we use four different models such as Decision Tree, Logistic Regression, Random Forest and XG Boost. We are looking at this models by using sampling methods in each of them and then, we made a decision about which of the model is better for our dataset.

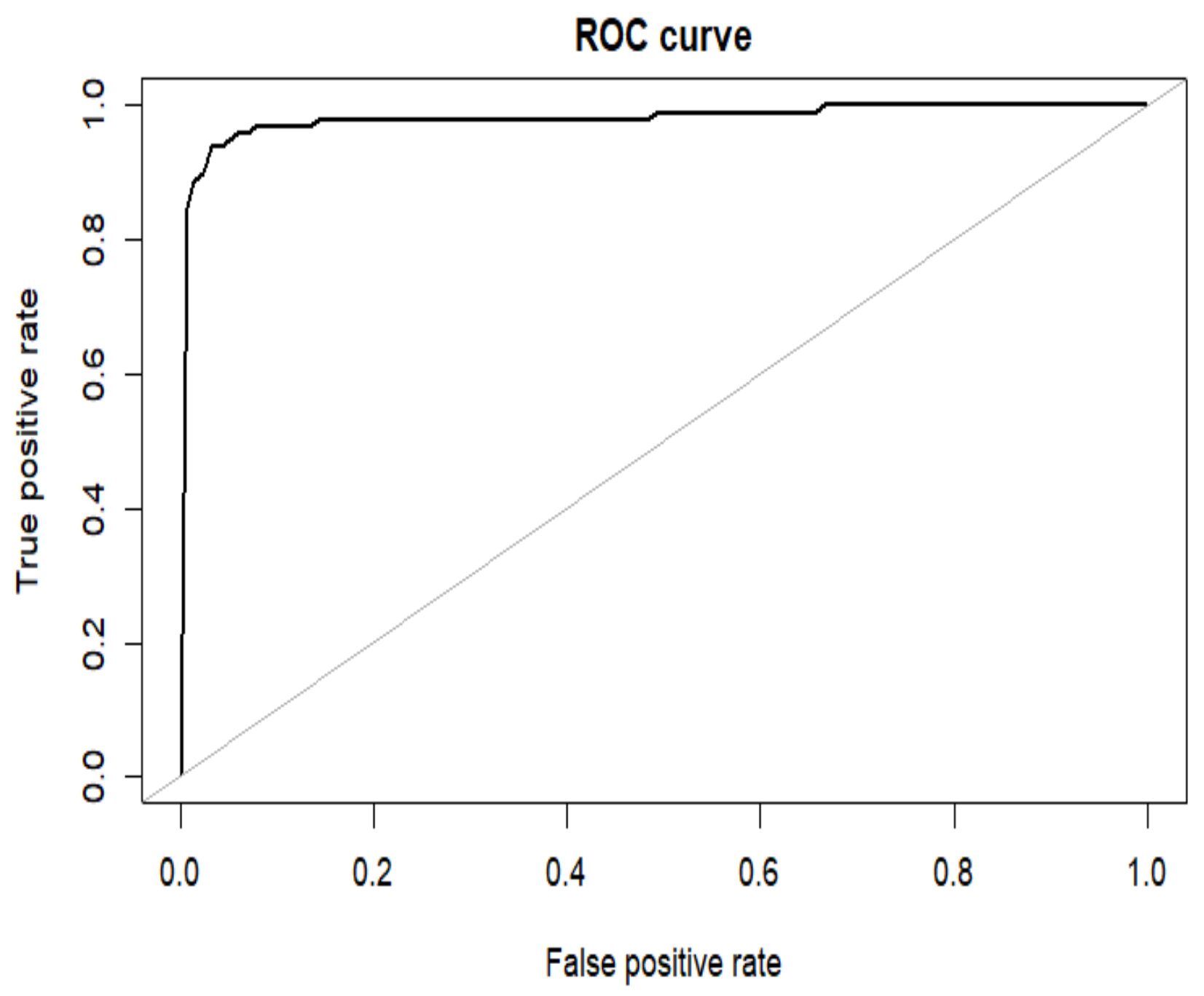
If we look at the roc curve generally, we can see that our roc curves close to the best one which means that close to true positive one. Then if we want to examine detailly or down to a single metric, AUC is useful for us in this issue. AUC stands for area under the (ROC) curve. Generally, the higher the AUC score (closest to 1.0), the better a classifier performs for the given task.

```
                    Original  Up-sampling  Down-sampling  Rose-sampling
Decision Tree          0.903        0.944          0.943          0.938
Logistic Regression    0.974        0.976          0.981          0.973
Random Forest          0.913        0.975          0.976          0.968
XG Boost               0.975        0.977          0.979          0.960
```

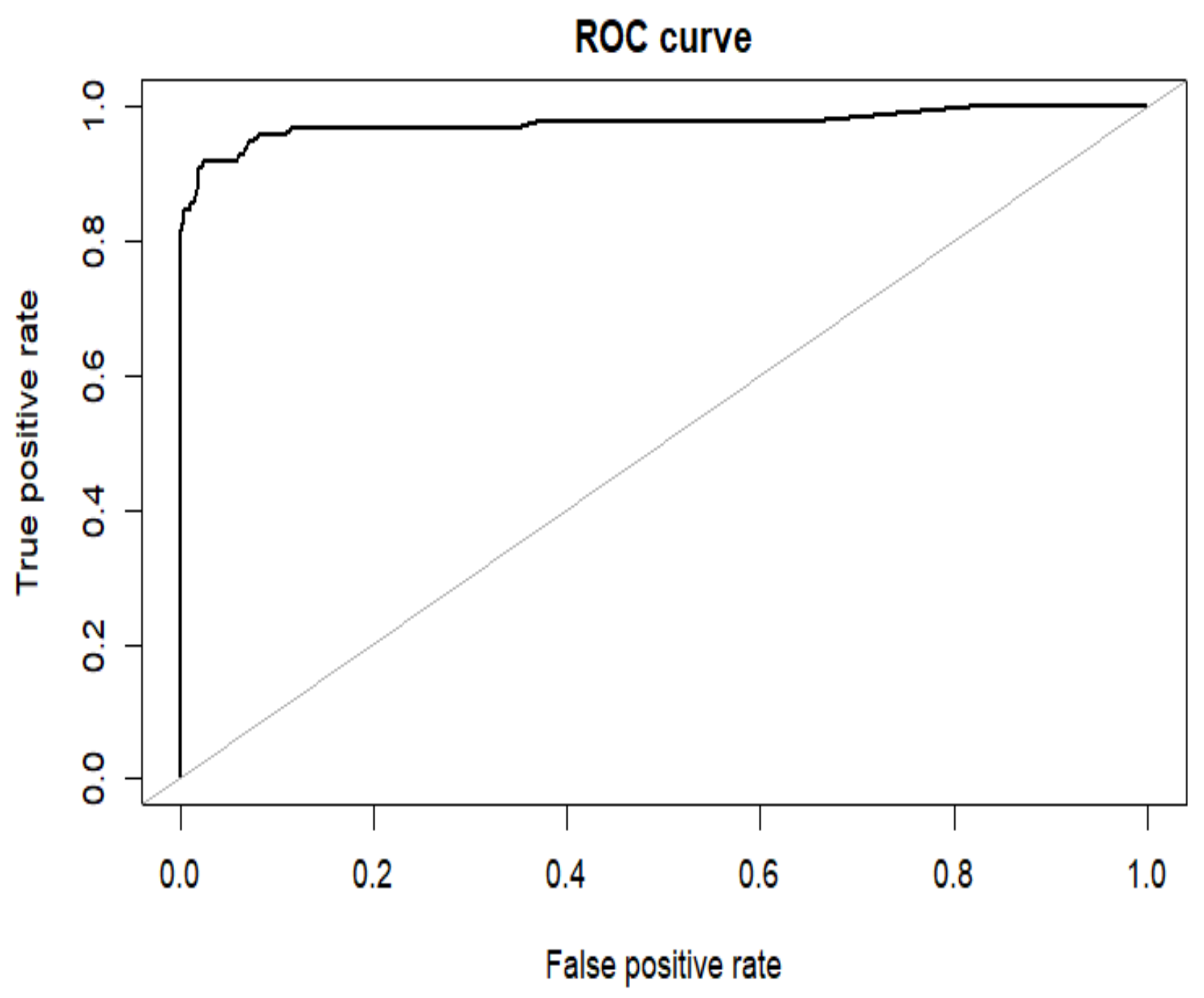
Decision Tree – Up-Sampling



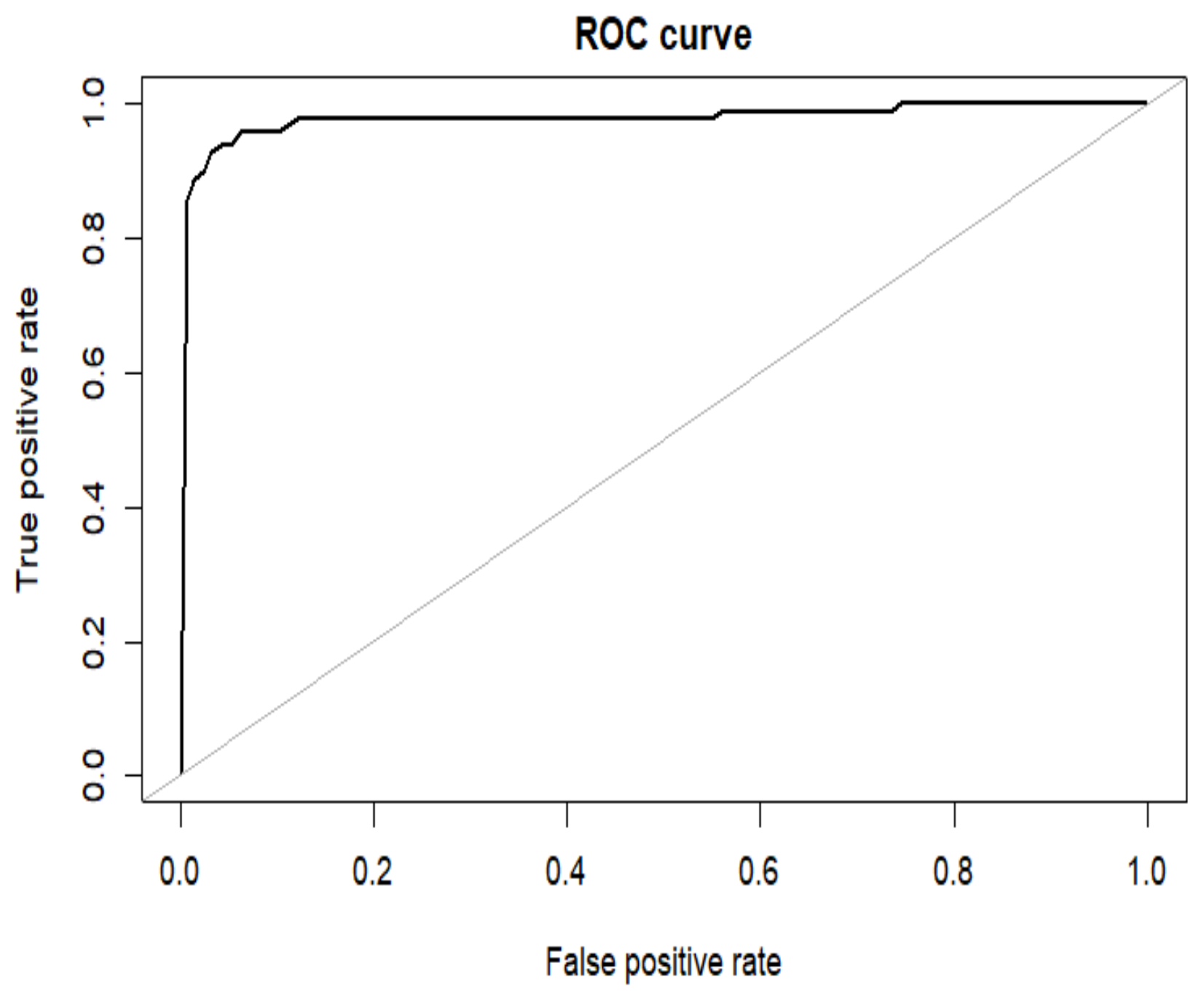
Logistic Regression – Down-Sampling



Random Forest – Down-Sampling



XG Boost – Down-Sampling



# Conclusion

1- Up-sampling method gives us better result in decision tree model. So, up-sampling is the most suitable sampling technique for the decision tree model in our dataset.

2- Down-sampling method gives us better result in logistic regression model. As a result of, down-sampling is the most suitable sampling technique for the logistic regression model in our dataset.

3- Down-sampling method gives us better result in random forest model. As a result of, down-sampling is the most suitable sampling technique for the random forest model in our dataset.

4- Down-sampling method gives us better result in XG Boost model. As a result of, down-sampling is the most suitable sampling technique for the XG Boost model in our dataset.

The best sampling technique's AUC scores for the models are;

1- Decision Tree: 0.944

2- Logistic Regression: 0.981

3- Random Forest: 0.976

4- XG Boost: 0.979

To sum up, we searched for the answer to the question of how we can make the our data more balanced in our Credit Card Fraud Detection project, that is, in a project where fraud cases are quite low and the dataset is very imbalanced. In response to this question, we saw that some resampling methods could be tried, besides, we tried these methods not only on a single model, but also on 4 different models, and determined which sampling method would perform better on which model for our data set. Among the sampling methods, we saw that the down-sampling method showed the best performance on 3 different models. We have come to the conclusion that the model that shows the most appropriate approach for our data, standing out among the complex algorithms such as Decision Tree, Random Forest and XG Boost, is Logistic Regression with a value of 0.981 AUC.

# References

-- Ingle, A. (2020, December 21). Credit Card Fraud Detection with R + (sampling). Kaggle. <https://www.kaggle.com/code/atharvaingle/credit-card-fraud-detection-with-r-sampling/notebook>

-- Zientala, P. (n.d.). Credit card fraud detection using Machine Learning. <http://rstudio-pubs-static.s3.amazonaws.com/334864_28050f7860dd4927a596872f0cd52401.html>

-- Team, T. A. I. (2020, May 29). How, When, and Why Should You Normalize / Standardize / Rescale. . . Towards AI. <https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff>

-- Team, T. A. I. (2022, March 17). Standardize Data Frame Columns in R (2 Examples) \| scale Function. Statistics Globe. <https://statisticsglobe.com/standardize-data-frame-columns-in-r-scale-function>

-- Steen, D. (2021, December 15). Understanding the ROC Curve and AUC - Towards Data Science. Medium. <https://towardsdatascience.com/understanding-the-roc-curve-and-auc-dd4f9a192ecb>

MELTEM AKKOCA
BERKAY DURSUN