CREDIT CARD FRAUD DETECTION

Thesis submitted in partial fulfillment of the

requirements for

Postgraduate diploma in data science

By

Jothi Prakash Anandan

(Reg no: 17125760069)

Under the guidance of

Ramakrishna Ganesan

Faculty

Manipal Prolearn

Bangalore, Karnataka

CREDIT CARD FRAUD DETECTION

Thesis submitted in partial fulfillment of the

requirements for

Postgraduate diploma in data science

By

Jothi Prakash Anandan

(Reg no: 17125760069)

Examiner 1	Examiner 2
Signature:	Signature:
Name:	Name:

CERTIFICATE

This is to certify that the project titled

CREDIT CARD FRAUD DETECTION

Is a bonafide record of the work done by

Jothi Prakash Anandan

Reg no: 17125760069

In partial fulfillment of the requirements for the award of **Postgraduate Diploma in data science** under Manipal University, Manipal and the same has not been submitted elsewhere for any kind of certification/recognition.

Ramakrishna Ganesan

Faculty

Manipal Prolearn

Manipal University

ACKNOWLEDGMENT

I would like to express my sincere gratitude to the Director of Manipal global academy of data science, **Dr. Ramesh Babu** for having given me the opportunity to work on the project titled **"Credit card fraud detection"**.

I would like to thank my project guide, **Mr. Ramakrishna Ganesan** for guiding me throughout this project term, providing valuable feedback and advice needed for successful completion of this project.

This project helped me research and learn important aspects of how credit defaults happen and also helped me sharpen my data analysis skills by working on this real business case.

2. TABLE OF CONTENTS

1. Acknowledgment
2. Table of contents
3. Abstract
4. Introduction
    a. Motivation
    b. Project scope
    c. Project focus
5. Project description
    a. Business and domain understanding
    b. Datasets understanding
    c. Data limitations
    d. Business objectives
6. Tools and technologies
    a. RStudio
    b. t-SNE
    c. Random forest algorithm
7. Exploratory data analysis
    a. Data exploration
    b. Data cleaning and transformation
8. Modeling
    a. Selection of model/technique
    b. Challenges faced
    c. Evaluation
    d. Validation metrics
    e. Variable importance and model interpretation
9. Conclusion
    a. Summary of project outcome
    b. Future works
10. References

3. ABSTRACT

In this project, machine learning, network analysis and UI building tools and techniques were used to analyze credit defaulting from credit card transactions in September 2013 by European cardholders. The dataset has been collected and analyzed during a research collaboration of Worldline and the Machine Learning Group (http://mlg.ulb.ac.be) of ULB (Université Libre de Bruxelles) on big data mining and fraud detection.

This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, they cannot provide the original features and more background information about the data. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

Further, various machine learning and visualization techniques were leveraged on the data to find defaults.

4. INTRODUCTION

a. Motivation

The health of the credit card industry is best measured not by the number of people with cards, but rather the number who pay their bills. Bad payment habits begin by nicking you with more fees and lower credit scores, and, in advanced cases, can lead to the loss of a vehicle or home, garnishment, and bankruptcy.

Credit card defaults, after a lengthy decline, are starting to tick up slightly.

Delinquencies in bank cards rose in the third quarter of 2016, but remain near historical lows, according to the American Bankers Association's Consumer Credit Delinquency Bulletin.

Bank card delinquencies increased 26 basis points to 2.74 percent of all accounts in the third quarter, but remain well below their 15-year average of 3.68 percent. The ABA report defines a delinquency as a late payment that is 30 days or more overdue.4

The trend looks similar when examining accounts that have been overdue for three months. TransUnion's Industry Insights Report found that the credit card delinquency rate reached 1.79 percent in Q4 2016, an increase of 12.6 percent from 1.59 percent in Q4 2015. The credit card delinquency rate remains more than a full point below its peak in Q4 2009 (2.97 percent).

b. Project scope

The scope of the project is somewhat restricted by the missing sensitive data. The PCA data might be useful for prediction but the high dimensional credit history data is really necessary in order to produce quality visualizations which are missing in the selected dataset.

The customer name, a date with an exact time of transfer, location, device used for money transfer are some of the crucial information necessary for clustering/segmenting the data and the number of defaults is very low and the unbalanced data might have some impact in the prediction and insights extracted.

To balance out the data and make the predictions and visuals in a justified manner t-SNE is used. To extract maximum insight variable importance is carried out in random forest algorithm. The Gini index will give us the idea of which dimensional data has the highest impact on the defaults.

c. Project focus

All the banks are having loses due to illegal or defaulted transactions that happen using their network and they are trying to predict/classify a transaction to be worthy or defaulting. There are heavy losses due to illegal transactions and some of them are mentioned below.

1. Estimates created by the Attorney-General's Department show that identity crime costs Australia upwards of $1.6 billion each year, with the majority of about $900 million being lost by individuals through credit card fraud, identity theft, and scams.
2. In 2015, the Minister for Justice and Minister Assisting the Prime Minister for Counter-Terrorism, Michael Keenan, released the report Identity Crime and Misuse in Australia 2013-14.
3. This report estimated that the total direct and indirect cost of identity crime was closer to $2 billion, which includes the direct and indirect losses experienced by government agencies and individuals, and the cost of identity crimes recorded by police.

To classify/predict the defaultable transaction is the target objective.

5. PROJECT DESCRIPTION

a. Business/Data understanding:

The dataset has been collected and analyzed during a research collaboration of Worldline and the Machine Learning Group (http://mlg.ulb.ac.be) of ULB (Université Libre de Bruxelles) on big data mining and fraud detection.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, they cannot provide the original features and more background information about the data. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

b. Data limitations:

This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

c. Dataset understanding

Column	Description	Type
Time	Number of seconds elapsed between each transaction (over two days)	Numeric
V1	-	Numeric
V2	-	Numeric
V3	-	Numeric
V4	-	Numeric
V5	-	Numeric
V6	-	Numeric
V7	-	Numeric
V8	-	Numeric
V9	-	Numeric
V10	-	Numeric
V11	-	Numeric
V12	-	Numeric
V13	-	Numeric
V14	-	Numeric
V15	-	Numeric
V16	-	Numeric
V17	-	Numeric
V18	-	Numeric
V19	-	Numeric
V20	-	Numeric
V21	-	Numeric
V22	-	Numeric
V23	-	Numeric
V24	-	Numeric
V25	-	Numeric
V26	-	Numeric
V27	-	Numeric
V28	-	Numeric
abc	-	Numeric
Amount	Amount of money for this transaction	Numeric
Class	Fraud or Not-Fraud	Boolean

d. Business objective

This project has been broken down into 3 subparts and objectives

1. Identify the important variables which led to the default.
2. Exploratory data analysis to be implemented as graphs and UI based decisions for efficient and user-friendly exploration and comparison of correlation.
3. Prediction of default using machine learning algorithms.

6. TOOLS AND TECHNOLOGIES

Exploratory data analysis & machine learning

1. RStudio
2. T-SNE
3. Random forest algorithm

RStudio

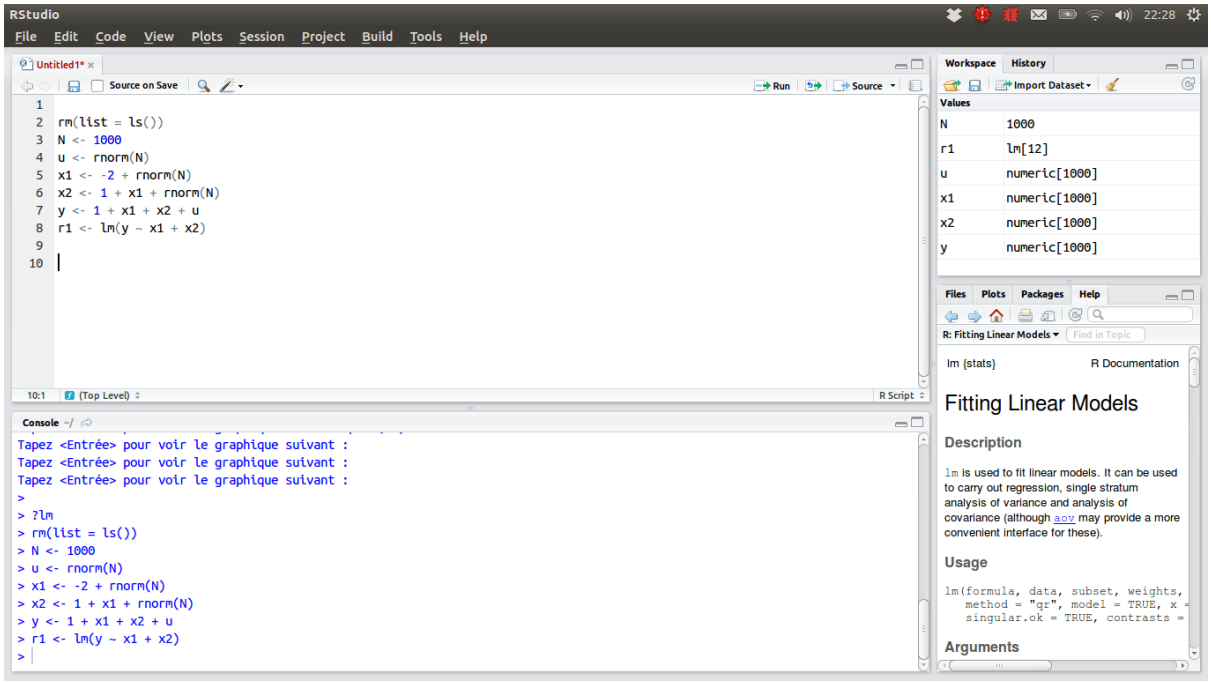
RStudio is a free and open-source integrated development environment (IDE) for R, a programming language for statistical computing and graphics. RStudio was founded by JJ Allaire, creator of the programming language ColdFusion. Hadley Wickham is the Chief Scientist at RStudio.

RStudio is available in two editions: RStudio Desktop, where the program is run locally as a regular desktop application; and RStudio Server, which allows accessing RStudio using a web browser while it is running on a remote Linux server. Prepackaged distributions of RStudio Desktop are available for Windows, macOS, and Linux.

RStudio is available in open source and commercial editions and runs on the desktop (Windows, macOS, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian, Ubuntu, Red Hat Linux, CentOS, OpenSUSE, and SLES).

RStudio is written in the C++ programming language and uses the Qt framework for its graphical user interface.

Work on RStudio started around December 2010, and the first public beta version (v0.92) was officially announced in February 2011. Version 1.0 was released on 1 November 2016. Version 1.1 was released on 9 October 2017.



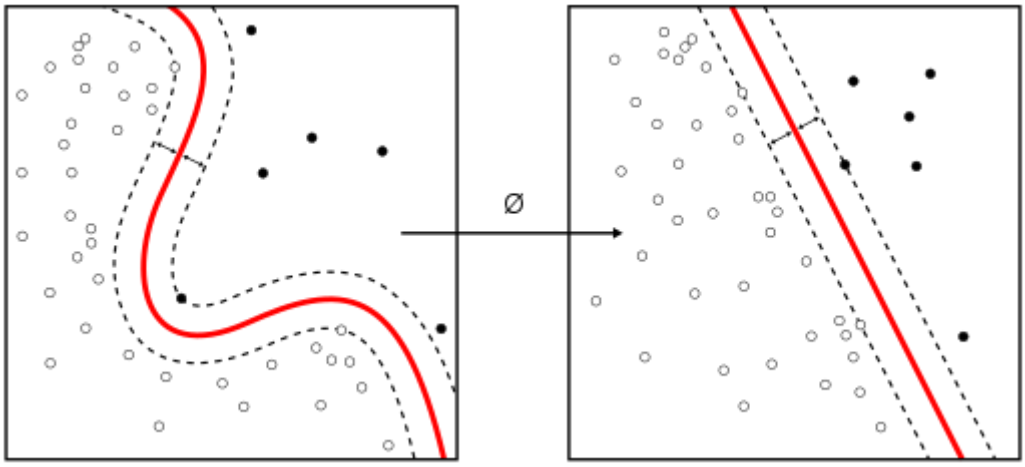
t-SNE

t-distributed stochastic neighbor embedding (t-SNE) is a machine learning algorithm for visualization developed by Laurens van der Maaten and Geoffrey Hinton. It is a nonlinear dimensionality reduction technique well-suited for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions. Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability.

The t-SNE algorithm comprises two main stages. First, t-SNE constructs a probability distribution over pairs of high-dimensional objects in such a way that similar objects have a high probability of being picked, whilst dissimilar points have an extremely small probability of being picked. Second, t-SNE defines a similar probability distribution over the points in the low-dimensional map, and it minimizes the Kullback–Leibler divergence between the two distributions with respect to the locations of the points in the map. Note that whilst the original algorithm uses the Euclidean distance between objects as the base of its similarity metric, this should be changed as appropriate.

t-SNE has been used for visualization in a wide range of applications, including computer security research, music analysis, cancer research, bioinformatics, and biomedical signal processing. It is often used to visualize high-level representations learned by an artificial neural network.

While t-SNE plots often seem to display clusters, the use of t-SNE for clustering has been shown to be unreliable, as t-SNE does not preserve distances. Even data coming from a single Gaussian can appear to be "clustered" in a t-SNE visualization.

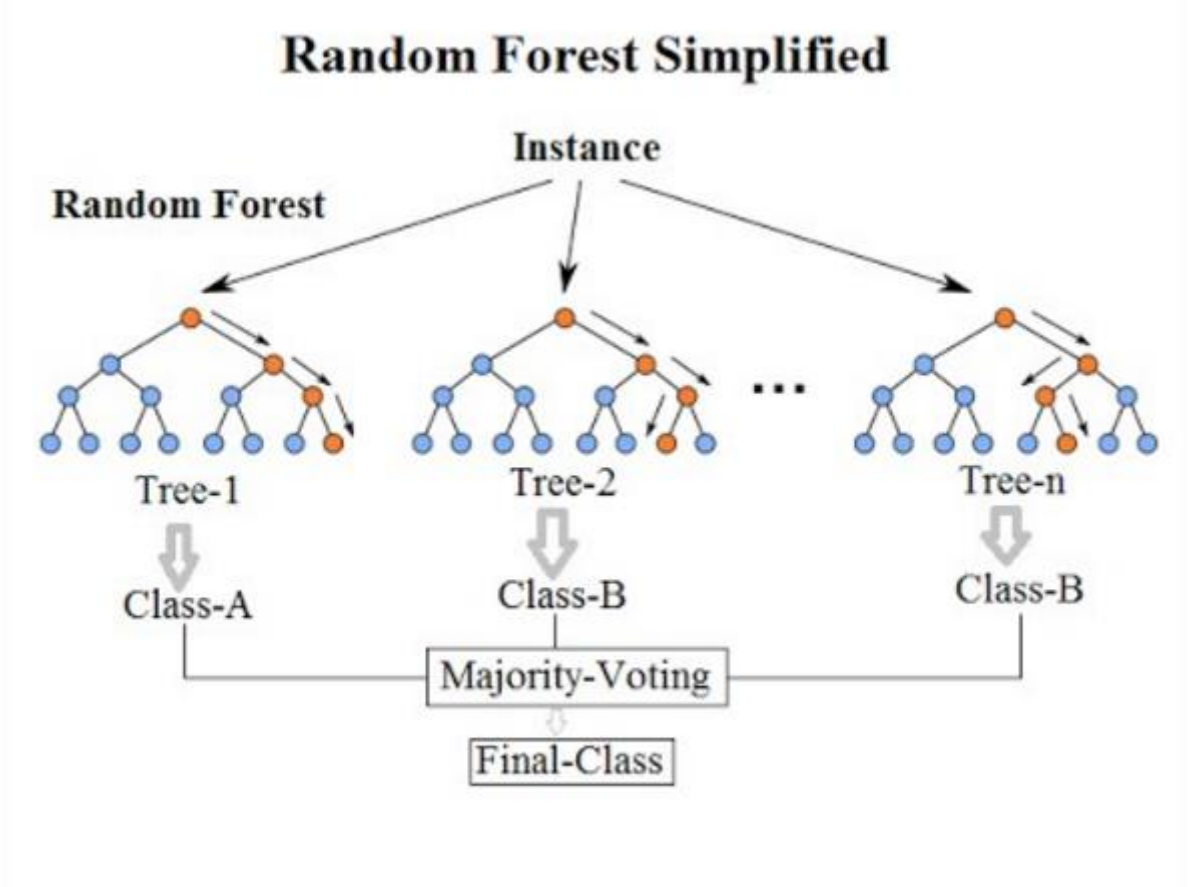


Random forest algorithm

Random forests or random decision forests are an ensemble learning method for classification, regression, and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or means prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to the classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, and "Random Forests" is their trademark. The extension combines Breiman's "bagging" idea and random selection of features introduced first by Ho and later independently by Amit and Geman in order to construct a collection of decision trees with controlled variance.


Random Forest Simplified
Instance
Random Forest
Tree-1
Tree-2
Tree-n
Class-A
Class-B
Class-B
Majority-Voting
Final-Class

7. EXPLORATORY DATA ANALYSIS

Exploratory data analysis(EDA) is an approach/philosophy for data analysis that employs a variety of techniques(mostly graphical) to

1. Maximize insight into the dataset
2. Uncover underlying data structure
3. Extract important variables
4. Detect outliers and anomalies
5. Test underlying assumptions
6. Develop parsimonious models
7. Determine optimal factor settings

a. Data exploration

a.1. Univariate data analysis

Univariate analysis is the simplest form of analyzing data. “Uni” means “one”, so in other words, your data has only one variable. It doesn’t deal with causes or relationships (unlike regression) and its major purpose is to describe; it takes data, summarizes that data and finds patterns in the data.

- Identifying variable type
- Identifying distribution
- Outlier identification and correction
- Data summary

a.2. Bivariate data analysis

Bivariate analysis is one of the simplest forms of quantitative (statistical) analysis. It involves the analysis of two variables (often denoted as X, Y), for the purpose of determining the empirical relationship between them.

Bivariate analysis can be helpful in testing simple hypotheses of association. Bivariate analysis can help determine to what extent it becomes easier to know and predict a value for one variable (possibly a dependent variable) if we know the value of the other variable (possibly the independent variable) (see also correlation and simple linear regression).

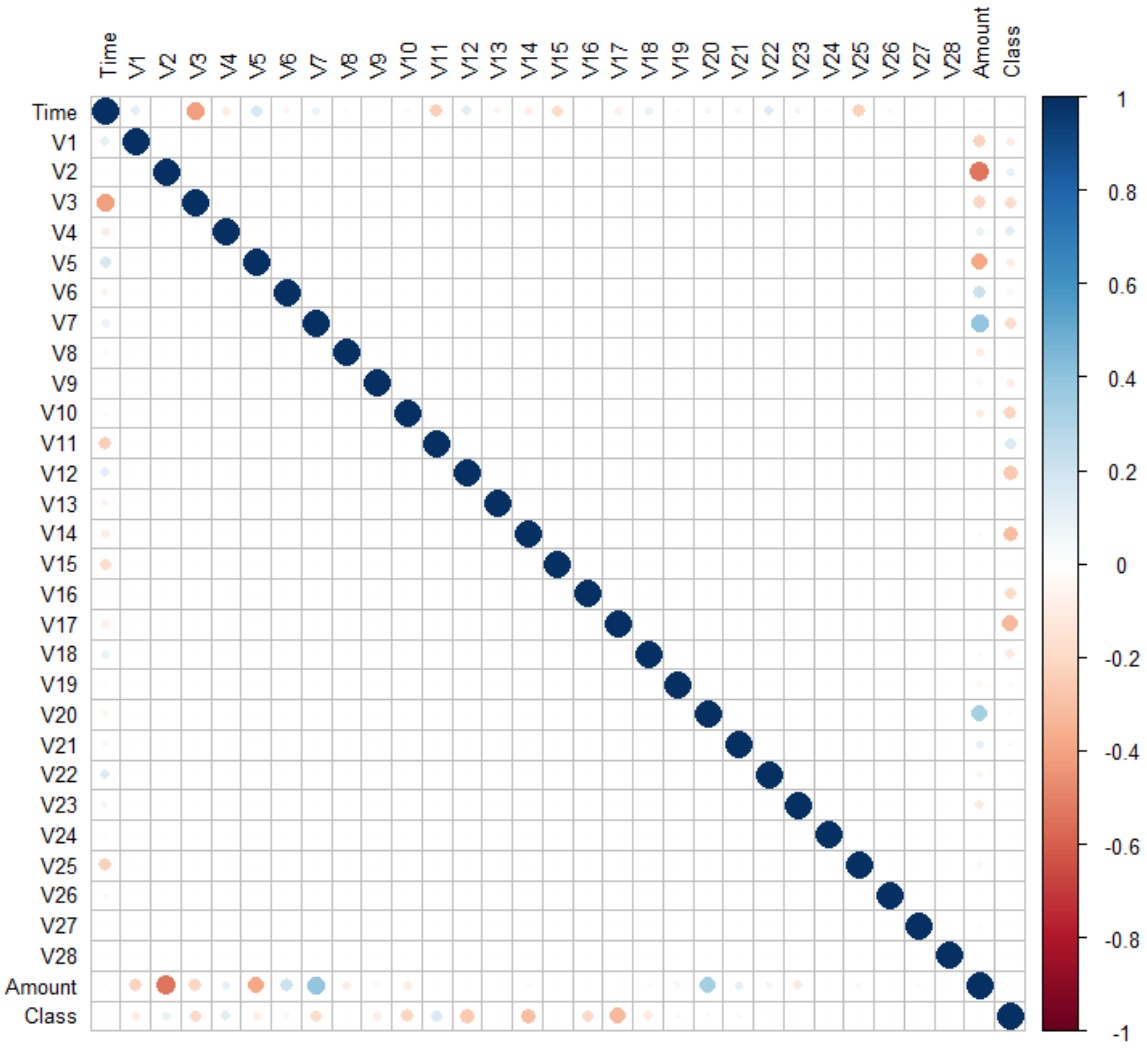
Bivariate analysis can be contrasted with the univariate analysis in which only one variable is analyzed. Like univariate analysis, bivariate analysis can be descriptive or inferential. It is the analysis of the relationship between the two variables. Bivariate analysis is a simple (two variable) special case of multivariate analysis (where multiple relations between multiple variables are examined simultaneously).

8. MODELING

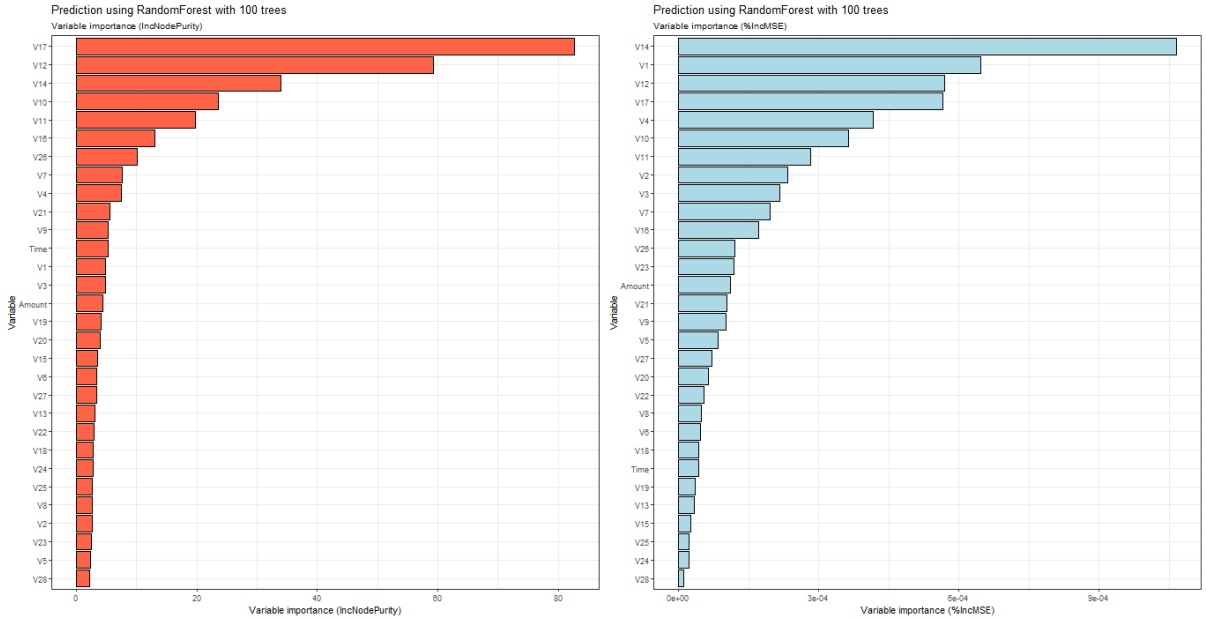
a. Selection of model/technique

As the original is masked with PCA components there is no much of data cleaning and transformation. When the dimensions are high PCA will be used to reduce the number of dimensions. In this case, to cover the sensitive credit data PCA is used.

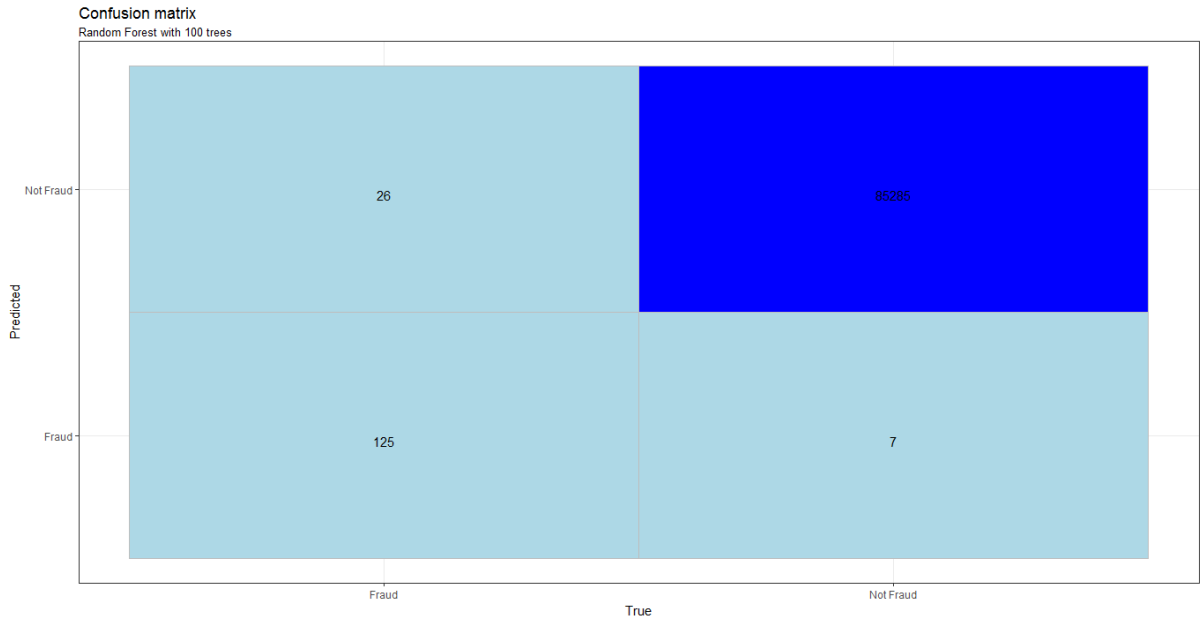
Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. If there are n observations with p variables, then the number of distinct principal components is min(n-1,p). This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component, in turn, has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.



We can observe that most of the data features are not correlated. This is because, before publishing, most of the features were presented to a Principal Component Analysis (PCA) algorithm. The features `V1` to `V28` is most probably the Principal Components resulted from propagating the real features through PCA. We do not know if the numbering of the features reflects the importance of the Principal Components. This information might be checked partially using the Variable Importance of Random Forest.



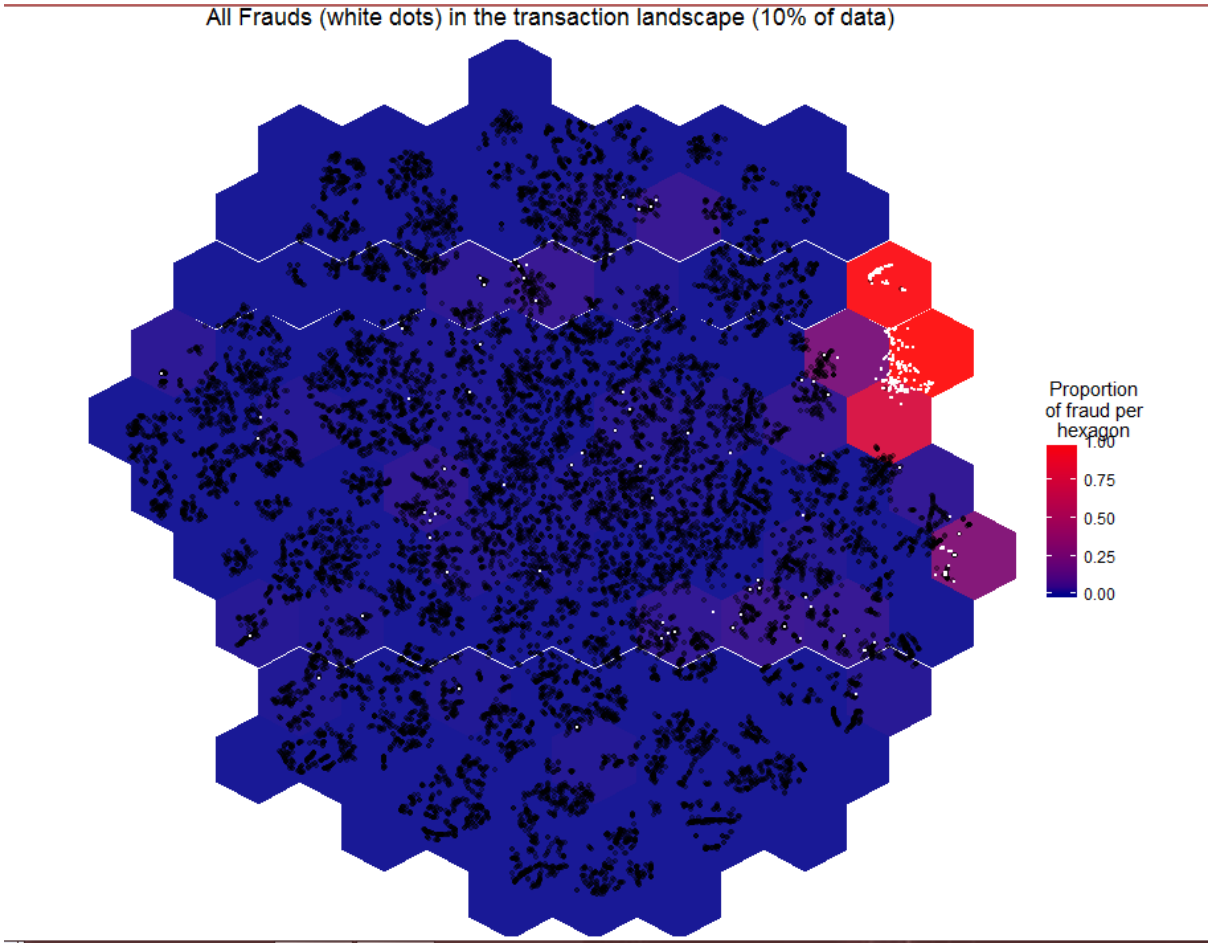
The confusion matrix of the prediction is given below



b. Challenges faced

t-SNE has been deployed in order to take a look at the PCA dimensions. The plot is a hexagonal diagram.

The hexagons show the local density of fraudulent transactions (white points). Red colors mean high density of fraud (typically > 75% of points included in the hexagon) whereas blueish colors are associated with a small fraction of fraud.



c. Evaluation

threshold	tpr	fpr	tp	fp	tn	fn	cost	auc
0.0000000	0.9139073	0.0259344	138	2212	83080	13	2342	0.94399
0.0101010	0.9006623	0.0086995	136	742	84550	15	892	0.94598
0.0202020	0.8940397	0.0044553	135	380	84912	16	540	0.94479
0.0303030	0.8874172	0.0028373	134	242	85050	17	412	0.94229
0.0404040	0.8874172	0.0020401	134	174	85118	17	344	0.94269
0.0505051	0.8874172	0.0014656	134	125	85167	17	295	0.94298
0.0606061	0.8741722	0.0011724	132	100	85192	19	290	0.93650
0.0707071	0.8741722	0.0009849	132	84	85208	19	274	0.93659
0.0808081	0.8741722	0.0008324	132	71	85221	19	261	0.93667
0.0909091	0.8675497	0.0007386	131	63	85229	20	263	0.93341
0.1010101	0.8675497	0.0006097	131	52	85240	20	252	0.93347
0.1111111	0.8675497	0.0005159	131	44	85248	20	244	0.93352
0.1212121	0.8675497	0.0004924	131	42	85250	20	242	0.93353
0.1313131	0.8609272	0.0004455	130	38	85254	21	248	0.93024
0.1414141	0.8609272	0.0004455	130	38	85254	21	248	0.93024
0.1515152	0.8609272	0.0004104	130	35	85257	21	245	0.93026

d. Validation metrics

- **Area under the curve (AUC)**

The AUROC has several equivalent interpretations

1. The expectation that a uniformly drawn random positive is ranked before a uniformly drawn random negative.
2. The expected proportion of positives ranked before a uniformly drawn random negative.
3. The expected true positive rate if the ranking is split just before a uniformly drawn random negative.
4. The expected proportion of negatives ranked after a uniformly drawn positive.
5. The expected false positive rate if the ranking is split just after a uniformly drawn random positive.

Computing the AUROC

Assume we have a probabilistic, binary classifier such as logistic regression.

Before presenting the ROC curve(Receiver Operating Characteristic curve), the concept of confusion matrix should understand. When we make a binary prediction, there can be four types of outcomes

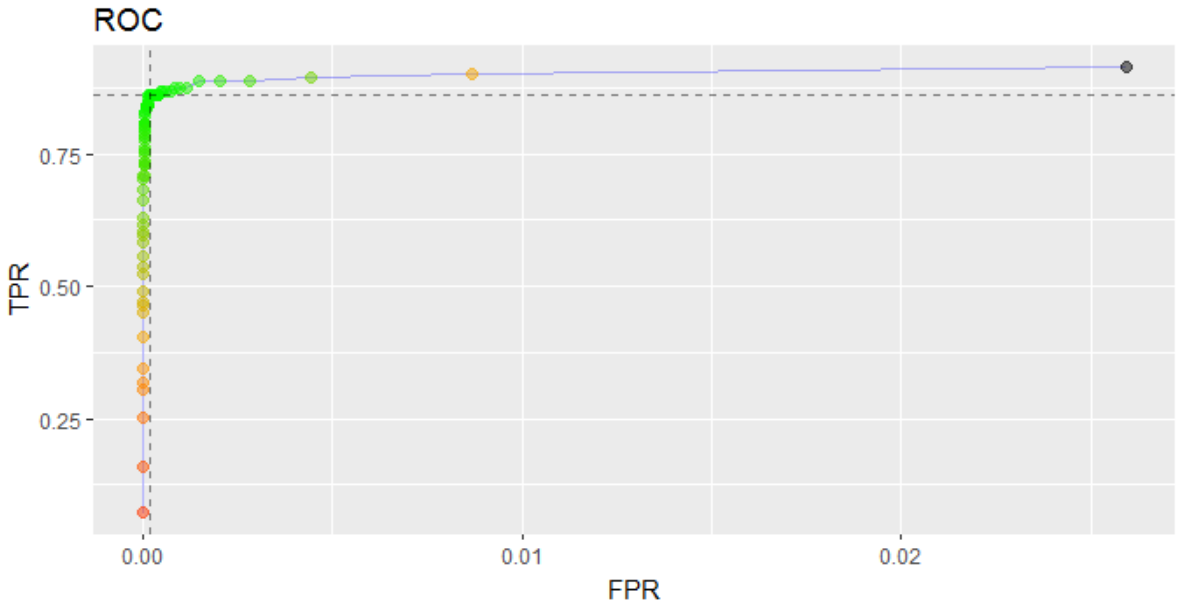
- We predict 0 while we should have the class as 0 – This is called “True Negative”. i.e., We correctly predict the class is negative (0).
- We predict 0 while we should have the class as 1 – This is called “False Negative”. i.e., We incorrectly predict the class is negative (0).
- We predict 1 while we should have the class as 0 – This is called “False Positive”. i.e., We incorrectly predict the class is positive (1).
- We predict 1 while we should have the class as 1 – This is called “True Positive”. i.e., We correctly predict the class is positive (1).

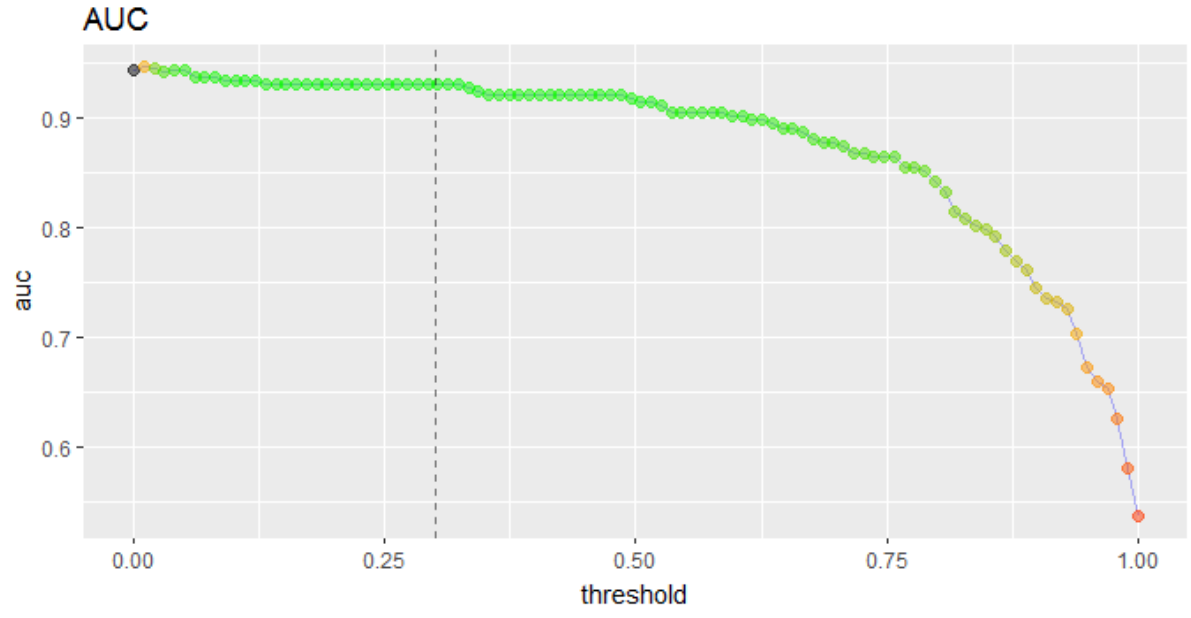
Since to compare two different models it is often more convenient to have a single metric rather than several ones, we compute two metrics from the confusion matrix which we will later combine into one.

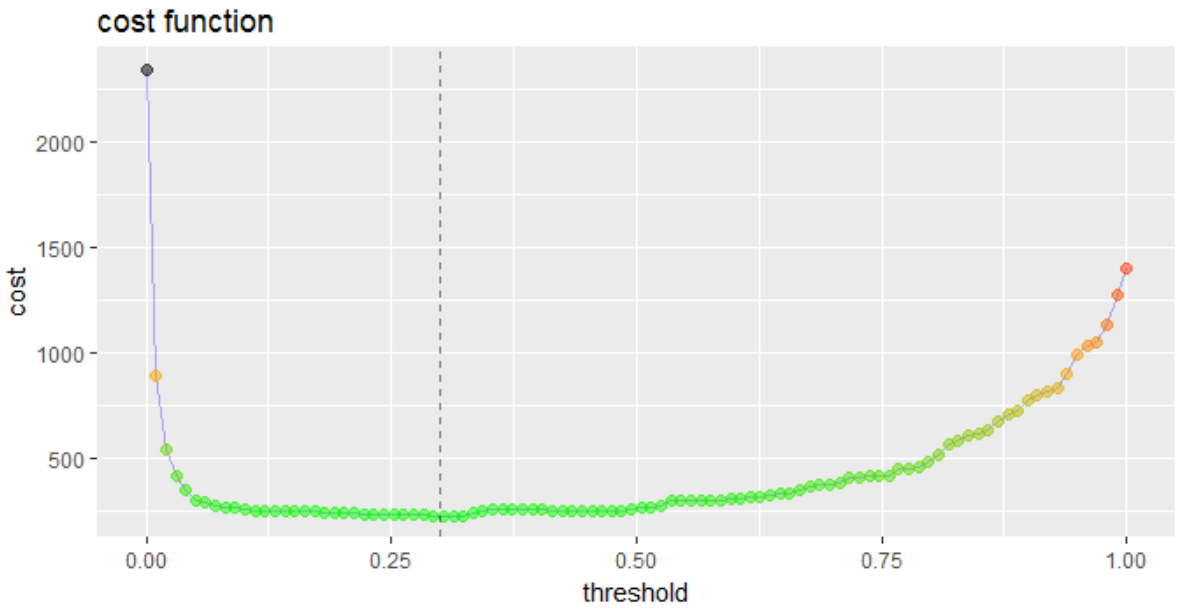
- True positive rate (TPR), aka. Sensitivity hit rate, and recall, which is defined as TPTP + FNTPTP + FN. Intuitively this method corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points. In other words, the higher TPR, the fewer positive data points we will miss.

- False positive rate (FPR), aka. Fall-out, which is defined as FPFP + TNFPFP + TN. Intuitively this metric corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points. In other words, the higher FPR, the more negative data points we will misclassify.

To combine the FPR and the TPR into one single metric, we first compute the two former metrics with many different threshold (for example: 0.01, 0.02, 1.0000) for the logistic regression, then plot them on a single graph, with the FPR values on the abscissa and the TPR values on the ordinate. The resulting curve is called ROC curve, and the metric we consider is the AUC of this curve, which we call AUROC.

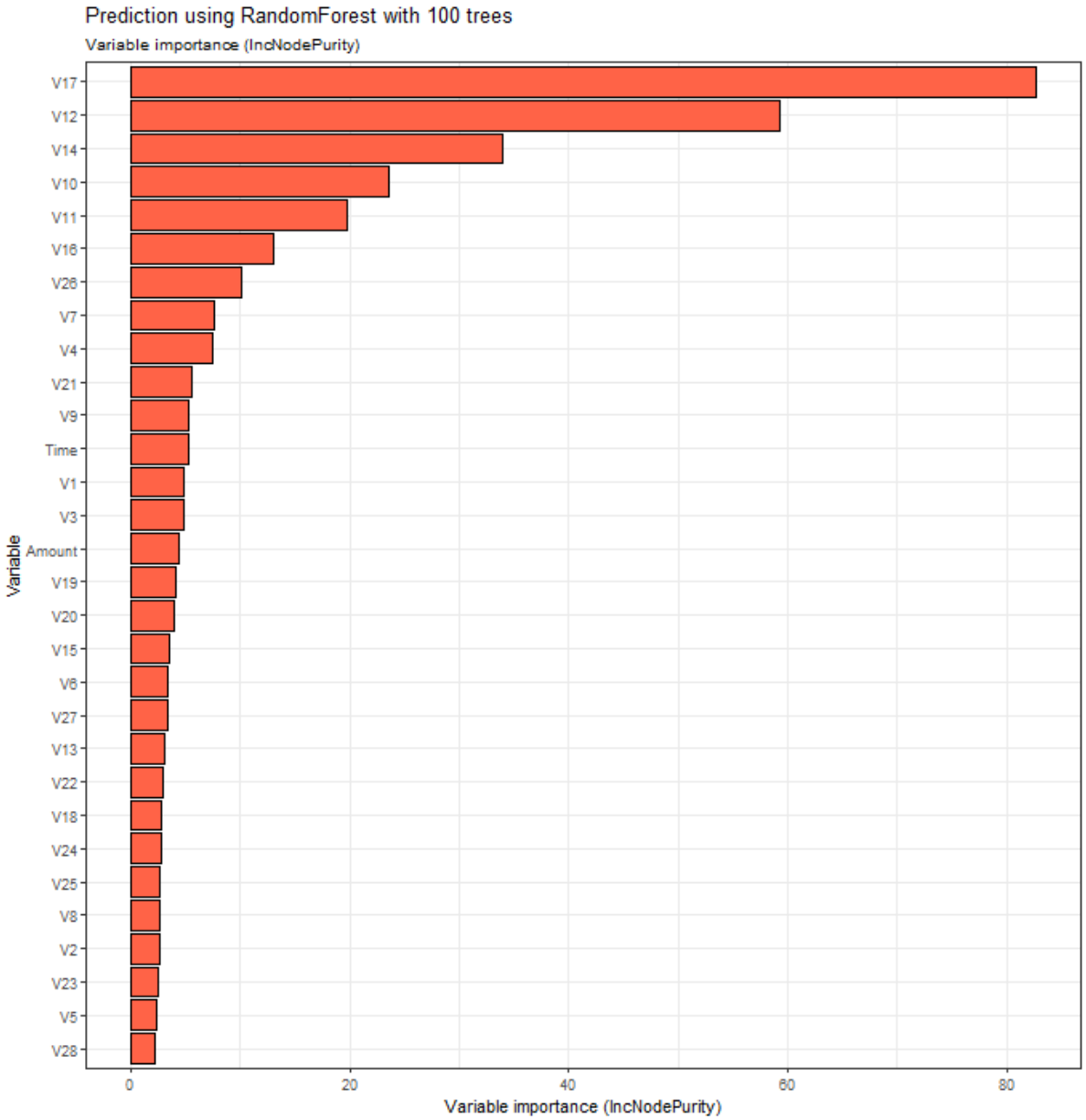


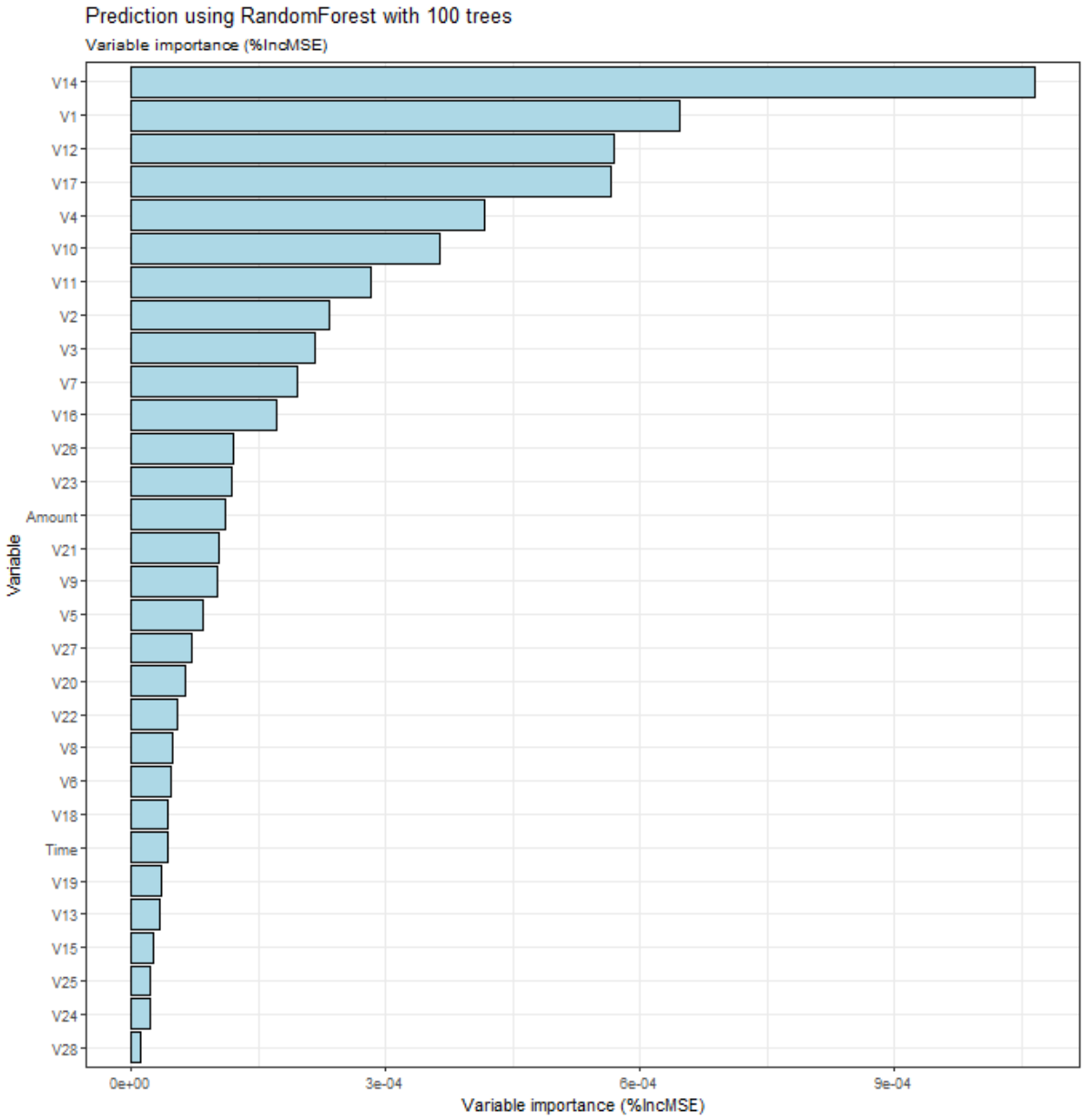




threshold at 0.30 - cost of FP = 1, cost of FN = 10

e. Variable importance and model interpretation




Prediction using RandomForest with 100 trees
Variable importance (%IncMSE)
V14
V1
V12
V17
V4
V10
V11
V2
V3
V7
V16
V28
V23
Amount
V21
V9
V5
V27
V20
V22
V8
V6
V18
Time
V19
V13
V15
V25
V24
V28
Variable
0e+00
3e-04
6e-04
9e-04
Variable importance (%IncMSE)

9. Conclusion

a. Summary of project outcome

The calculated accuracy is not very relevant in the conditions where there is a very large unbalance between the number of `fraud` and `non-fraud` events in the dataset. In such cases, we can see a very large accuracy.

More relevant is the value of ROC-AUC (Area under Curve for the Receiver Operator Characteristic). The value obtained (0.93) is relatively good.

10. References

https://en.wikipedia.org

https://www.rstudio.com/products/rstudio/

www.statisticssolutions.com

https://www.itl.nist.gov

https://github.com

https://stats.stackexchange.com

https://www.rdocumentation.org

https://www.analyticsvidhya.com

https://www.kaggle.com

https://sciencedirect.com

www.wikihow.com