Data Mining for Business (BUDT758T)

Project Title: Predicting Credit Card Fraud Transactions

Team Members: Ishan Chaturvedi, Georges Colbert, Shreyas Kupekar, Akshat Maltare, Charles Wilson

(SIGN THE FOLLOWING STATEMENT AND INCLUDE IT ON THE COVER PAGE OF YOUR PROPOSAL)

ORIGINAL WORK STATEMENT

We the undersigned certify that the actual composition of this proposal was done by us and is original work.

	Typed Name	**Signature**
Contact Author	Ishan Chaturvedi	
	Georges Colbert	
	Shreyas Kupekar	
	Akshat Maltare	
	Charles Wilson	

Contents

Executive Summary

With development of technology, Credit Card Fraud has been increasing day by day. Although incidents of fraud are limited to 0.1% of all transactions (1), they result in a loss of billions of dollars for financial companies and consumers. So fraud detection systems have become a must have for financial firms, to minimize their losses. However, lack of transactional data due to data privacy laws have made it difficult to research on fraud detection methods.

In this paper, we are proposing various machine learning models to analyze 284,807 credit card transactions and compare them to select an algorithm that reduces the losses of financial firms. The main challenge is to handle a highly unbalanced data set. To do this, we have used various sampling techniques for balancing the data first and applied each model to each sampling technique. This provides us with 42 different models which we have compared on the performance measures discussed further in this paper. Through our research, we found out that Recall will be an appropriate measure to measure the performance of our classifier as the transactions which are fraud and classified as non-fraud have a higher cost associated with them. It's possible to increase the recall of a classification algorithm by using the sampling methods discussed further in the report.

Data Description and Pre-processing

The dataset contains credit card transactions made in September 2013 by European cardholders. It represents two days of 284,807 transactions where 492 were flagged as fraudulent. The dataset is highly imbalanced, the positive class (fraud) account for only 0.172% of all transactions. The dataset contains 30 independent variables for which 28 (V1, V2, ... V28) are the result of a Principal Component Analysis (PCA) transformation. Since the original features were scrubbed for privacy we will not be able to make any explanatory judgment from our models. The feature, 'Class' is the response variable and it takes value '1' in case of a fraudulent transaction and '0' otherwise. The only untransformed independent variables are 'Time' and 'Amount'. The feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction dollar amount.

- *Data Source*: Credit Card Fraud dataset is available on Kaggle.com (https://www.kaggle.com/dalpozz/creditcardfraud)

- n = 284,807
- k = 30
    - 28 numerical features (V1, V2, ...V28)
    - Time (seconds)
    - Amount (dollars)
    - Class (binary factor)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
2	0	-1.36	-0.07	2.54	1.38	-0.34	0.46	0.24	0.10	0.36	0.09	-0.55	-0.62	-0.99	-0.31	1.47	-0.47	0.21	0.03	0.40	0.25	-0.02	0.28	-0.11	0.07	0.13	-0.19	0.13	-0.02	149.62	0
3	0	1.19	0.27	0.17	0.45	0.06	-0.08	-0.08	0.09	-0.26	-0.17	1.61	1.07	0.49	-0.14	0.64	0.46	-0.11	-0.18	-0.15	-0.07	-0.23	-0.64	0.10	-0.34	0.17	0.13	-0.01	0.01	2.69	0
4	1	-1.36	-1.34	1.77	0.38	-0.50	1.80	0.79	0.25	-1.51	0.21	0.62	0.07	0.72	-0.17	2.35	-2.89	1.11	-0.12	-2.26	0.52	0.25	0.77	0.91	-0.69	-0.33	-0.14	-0.06	-0.06	378.66	0
5	1	-0.97	-0.19	1.79	-0.86	-0.01	1.25	0.24	0.38	-1.39	-0.05	-0.23	0.18	0.51	-0.29	-0.63	-1.06	-0.68	1.97	-1.23	-0.21	-0.11	0.01	-0.19	-1.18	0.65	-0.22	0.06	0.06	123.5	0
6	472	-3.04	-3.16	1.09	2.29	1.36	-1.06	0.33	-0.07	-0.27	-0.84	-0.41	-0.50	0.68	-1.69	2.00	0.67	0.60	1.73	0.28	2.10	0.66	0.44	1.38	-0.29	0.28	-0.15	-0.25	0.04	529	1
7	2	-1.16	0.88	1.55	0.40	-0.41	0.10	0.59	-0.27	0.82	0.75	-0.82	0.54	1.35	-1.12	0.18	-0.45	-0.24	-0.04	0.80	0.41	-0.01	0.80	-0.14	0.14	-0.21	0.50	0.22	0.22	69.99	0
8	2	-0.43	0.96	1.14	-0.17	0.42	-0.03	0.48	0.26	-0.57	-0.37	1.34	0.36	-0.36	-0.14	0.52	0.40	-0.06	0.07	-0.03	0.08	-0.21	-0.56	-0.03	-0.37	-0.23	0.11	0.25	0.08	3.67	0

We applied both Forward Stepwise Selection and Backward Stepwise selection algorithms for selecting relevant features. They selected all features except ‘Time’. So we built our model on 29 features (V1-V28 and Amount) with Class as a dependent variable. Then, we split the data 70/30 into training and testing samples. Testing sample was used to test every model on unseen data. We applied several sampling methods to the training data and trained each varying model on each sampling technique.

Research Question

How can data science mitigate losses for consumers and merchants from credit card fraud without the frustration of false positives?

Predictive analytics and machine learning (ML) offer a solution to fraud detection and prevention. However, there is an underlying problem when attempting to build predictive models on this type of data, *imbalanced class of interest*. Even with the serious volume of fraud, fraudulent transactions are hidden among trillions of dollars in total credit card transaction volume ($3.09 trillion in the US alone). Imbalanced data makes it more difficult to accurately predict the minority class (fraud). With imbalanced datasets, most ML algorithms will be very good at predicting/classifying the majority class but, terrible at predicting/classifying the minority class.

Proper management of imbalanced datasets are at the foundation of answering this question and solving similar conundrums. To illustrate using our credit card fraud example, consider these excerpts from several large research studies:

- Javelin Strategy & Research report that credit card fraud costs consumers over $16 billion, annually and Lexisnexis estimates the global cost of credit card fraud is upwards of $190 billion.
- A 2016 Lexisnexis, “True Cost of Fraud” study reported, “for every dollar of losses, merchants are losing $2.40 based on chargebacks, fees and merchandise replacement.”
- On the other side of the coin, a creditcards.com survey found, “About 4 in 10 Americans have had a transaction blocked or questioned by their card company. Of those who were alerted to possible fraudulent activity, more than half (53 percent) said all or most of the questioned charges were false alarms.”

One can easily determine that Type I errors are extremely costly for merchants. However, false positives can frustrate and/or embarrass customers (especially when a charge is flagged at a brick and mortar store). Merchants are weighing the trade-off of losses associated with fraud, including lost customers due to false positives, versus solution investments to mitigate such issues in the first place.

Methodology

The primary means of dealing with imbalanced data is using ‘Sampling Methods’ with the aim to modify the imbalanced data into a balanced distribution. Below are the methods we explored in our research:

- Undersampling the Majority Class
- Oversampling the Minority Class
- Both under and over technique
- Synthetic Data Generation (ROSE & SMOTE)

Undersampling

Undersampling refers to artificially *reducing the number of observations from the majority class* to create a balanced distribution. This method is helpful with extremely large datasets because it

reduces the overall number of observations and, therefore, computational requirements. Obviously, removing observations has the potential of also removing important training data of the majority class and affect the predictive performance of ML algorithms.

OverSampling

Oversampling refers to artificially *increasing the number of observations from the minority class* to create a balanced distribution. Also known as *upsampling,* the advantage is that there is no information loss like that of *under sampling.* Since *oversampling* add replicated observations to the dataset, it duplicates 'types' of observations. Therefore, this method is prone to over fitting.

Synthetic Data Generation

Synthetic Data Generation seeks to reap the benefits of under and over sampling without the consequences. This method creates artificial observations instead of simply adding the observations from the minority class. Because of this, it's also considered an *oversampling* technique. *Synthetic Minority Oversampling Technique (SMOTE)*, is an algorithm that creates artificial data based on feature space (rather than data space) similarities from the minority observations. SMOTE calculates the nearest minority class observation for each minority observation, k. Then, synthetically generates event k1 such that k1 lies between k and i. Finally, it randomly selects an observation out of 5 nearest points to apply to the dataset.

Random Over-Sampling Examples

Random Over-Sampling Examples (ROSE) is another *oversampling* method that uses a smoothed bootstrap approach.

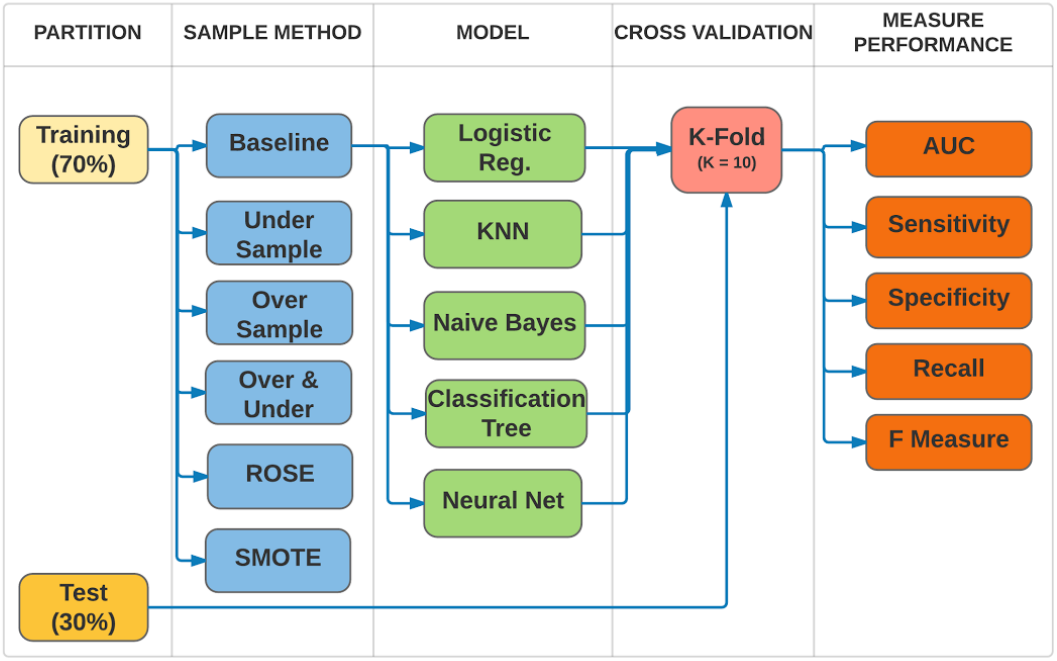
Modeling

We trained each model using each of the above mentioned sampling technique. We trained the following models:

- Logistic Regression
- K Nearest Neighbor (KNN)
- Naive Bayes

- Classification Tree
- Random Forest
- Boosting
- Neural Net

Since we used five different sampling techniques with five different models we developed 42 different combinations. Some sampling techniques were incompatible with certain models due to our computational limitations.



Performance Metrics

Our goal was to accurately predict fraud while minimizing prediction errors, especially Type I. The earlier referenced, 2016 Lexisnexis study calculated that fraudulent charges cost merchants $2.40 per every fraudulent dollar charged. Since the cost of customer annoyance is somewhat

abstract and calculating its dollar value is far beyond this study, we considered its net cost zero. Regardless, we sought to minimize false positives, subordinate to false negatives.

Accuracy and Error rate are poor measures of imbalanced data. Consider our dataset as an example: If we exercised the naive rule and classified every transaction as non-fraud (class 0), it would result in a deceptive 99.8% accuracy and a 0.172% error rate. Therefore we used Recall as a correct measure of performance as increasing recall means decrease in False Negatives. While we will try to increase recall, we need to take care that there are not a lot of False Positives as it will decrease the Specificity and area under the ROC curve (AUC) of the model.

		Predicted	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

***Recall*:** a measure of actual observations which are labeled (predicted) correctly i.e. how many observations of positive class are labeled correctly. It is also known as *'Sensitivity'*.

$$\text{Recall} = TP / (TP + FN)$$

Specificity: a measure of True Negative Rate, i.e., the number of transactions that were classified "not fraud" out of actual correct transactions.

$$\text{Specificity} = TN / (TN + FP)$$

The *ROC* (Receiver Operating Characteristics) *Curve* is formed by plotting TP rate (*Sensitivity*) and FP rate (1 - *Specificity*). Any point on ROC graph, corresponds to the performance of a single classifier on a given distribution. It is useful because it provides a visual representation of benefits (TP) and costs (FP) of a classification data. The larger the *area under ROC curve*, higher will be the accuracy.

We will use a cutoff of 0.5 for all our classification methods. Cutoff can be set to a higher value if high Recall is an absolute requirement but it can deteriorate the performance of a classifier. Our goal is to select a sampling technique and model that provides us maximum Recall on unseen data (test sample) without compromising on the Accuracy and Specificity.

Results and Findings

As mentioned earlier we trained seven different algorithms for six different sampling methods and then calculated the performance measures for each of them. Based on the performance measures that we discussed in the last section, Recall, Specificity and AUC of all the models that we trained is provided in **Appendix A**.

The general trend that we see from these results is that SMOTE is providing us with the best Recall without compromising on the accuracy on test data. In some models like Logistic Regression and Classification Trees, SMOTE is providing us highest recall as well as accuracy which means that our model classifies most of the fraud and non-fraud transactions correctly. For Boosting, we saw that using SMOTE classifies all records correctly for our test data. This minimizes the cost of both type 1 and type 2 error.

If we look at the tables in Appendix A, we got best classifiers for every algorithms. Comparison of those are given below:

	Recall	Specificity	AUC
Logistic Regression	0.9	0.96	0.97
Tree	0.887	0.953	0.945
KNN	0.82	0.999	0.892
Naive Bayes	0.833	0.999	0.88
Random Forest	0.847	0.99	0.915

Boosting	1	1	0.997
Neural Networks	0.989	0.989	0.989

Boosting with SMOTE is clearly is the best model to predict fraud transactions

Conclusion

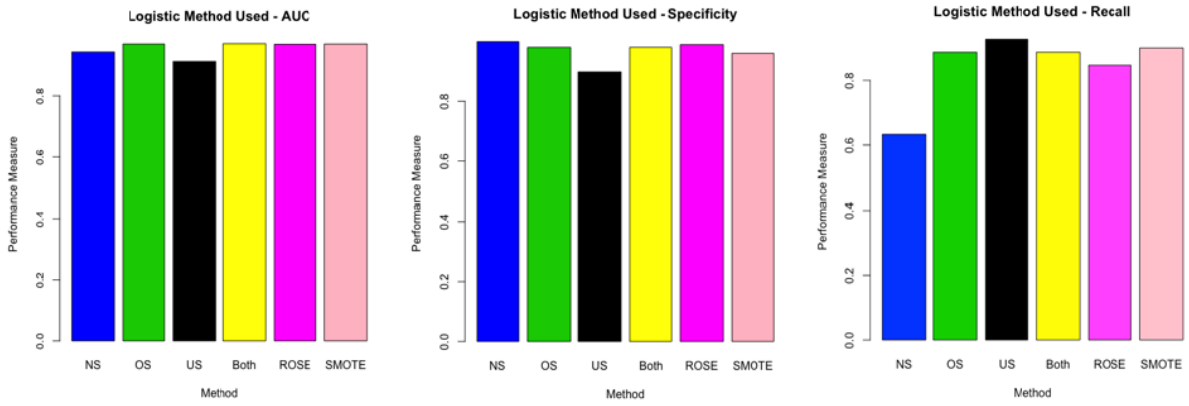
In our study, Boosting provided the best results. However, the academic literature suggests that no single imbalanced data strategy is coherently superior, to all others, in all conditions. Additionally, in the case of fraud, fraudsters adapt their methods to beat the system so it is likely to conclude that the strategy to detect fraud will also need to adapt. The class imbalance problem is well known and different techniques and metrics exist to deal with it. The best strategy is extremely dependent on the data, algorithm adopted, and the best performance measure for the scenario. When confronted with an imbalanced classification problem, it is our recommendation to consider the business problem, computational capability, data characteristics and select a combination of sampling techniques and algorithms that are complimentary to your particular scenario.

Appendix A

Logistic Regression:

Logistic Regression	NS	OS	US	Both	ROSE	SMOTE
Recall	0.633	0.887	0.927	0.887	0.847	0.9
Specificity	0.999	0.980	0.898	0.980	0.989	0.960

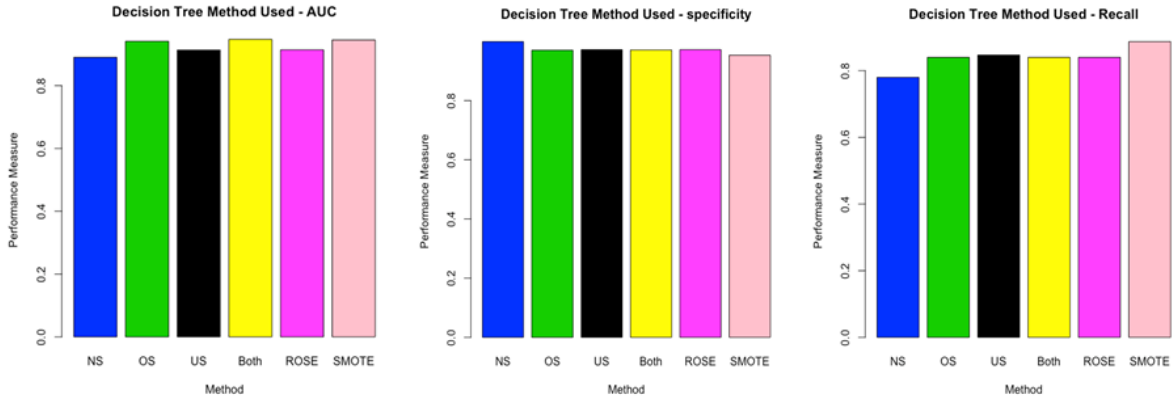
AUC	0.944	0.97	0.913	0.971	0.969	0.970



Performance Measures for Logistic Regression

Classification Trees:

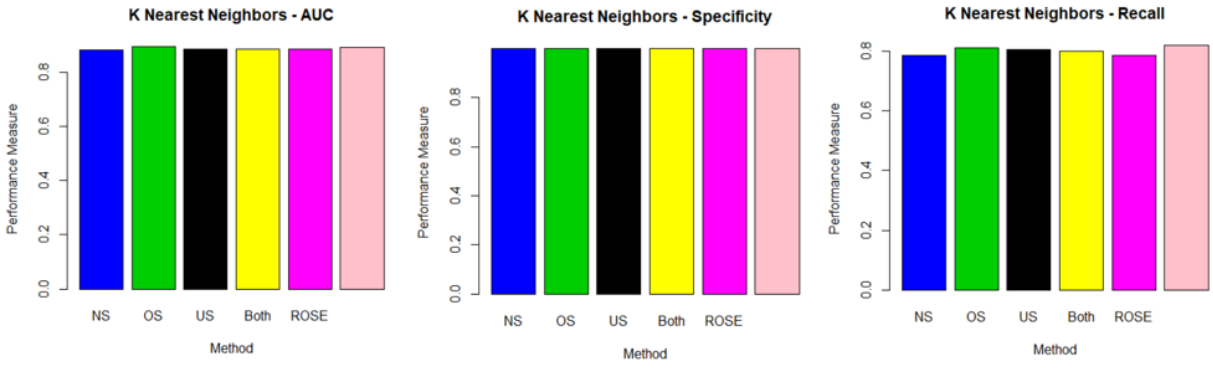
Trees	NS	OS	US	Both	ROSE	SMOTE
Recall	0.785	0.84	0.847	0.84	0.84	0.887
Specificity	0.999	0.970	0.972	0.971	0.972	0.953
AUC	0.890	0.941	0.913	0.947	0.914	0.945



Plots of Performance Measures for Trees

K Nearest Neighbors:

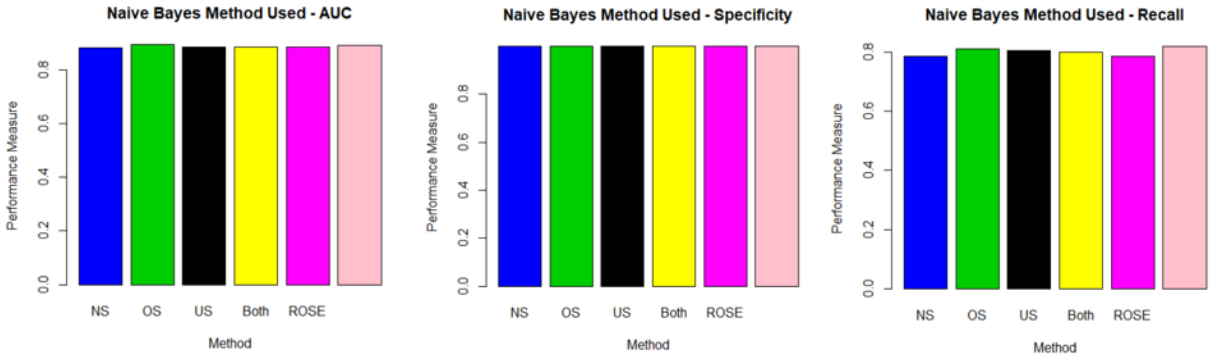
KNN	NS	OS	US	Both	ROSE	SMOTE
Recall	0.774	0.790	0.806	0.793	0.805	0.833
Specificity	0.999	0.952	0.953	0.958	0.960	0.951
AUC	0.844	0.845	0.881	0.876	0.866	0.880



Performance Measures for K Nearest Neighbors

Naive Bayes:

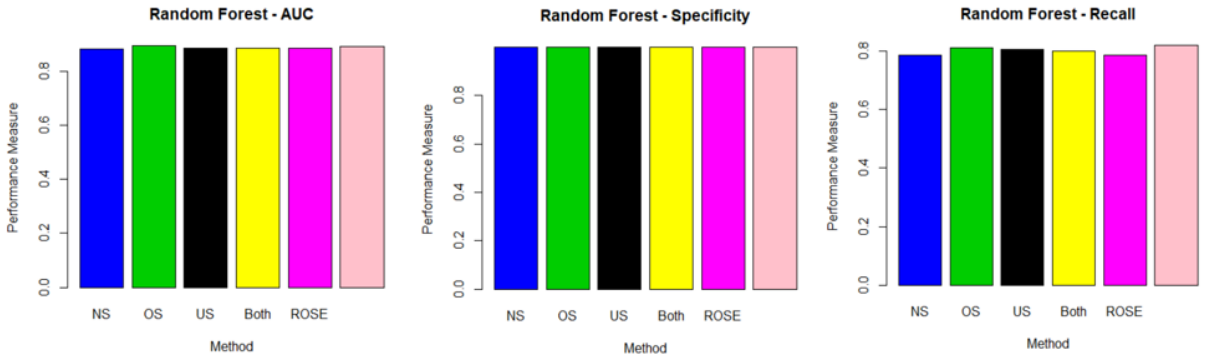
Naive Bayes	NS	OS	US	Both	ROSE	SMOTE
Recall	0.787	0.813	0.807	0.800	0.787	0.820
Specificity	0.979	0.974	0.968	0.974	0.981	0.960
AUC	0.883	0.894	0.887	0.887	0.884	0.892



Performance Measures for Naive Bayes

Random Forest:

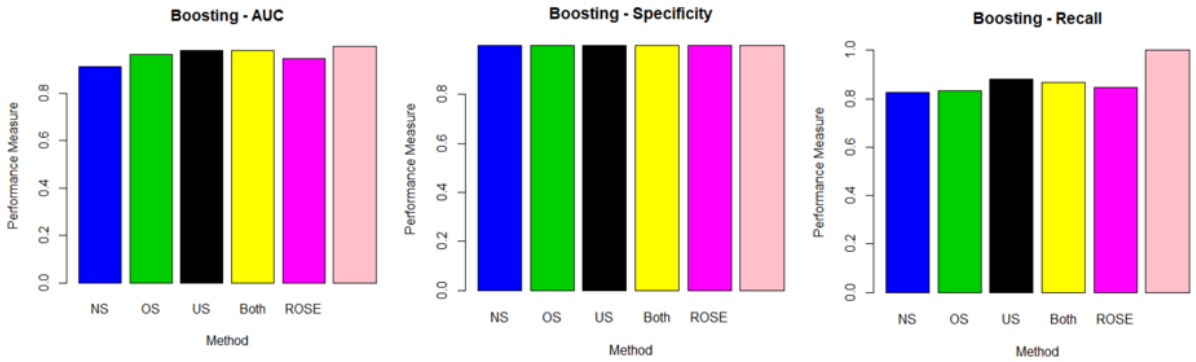
Random Forest	NS	OS	US	Both	ROSE	SMOTE
Recall	0.753	0.815	0.86	0.847	0.787	0.847
Specificity	0.968	0.972	0.978	0.949	0.984	0.99
AUC	0.877	0.900	0.919	0.918	0.877	0.915



Performance Measures for Random Forest

Boosting:

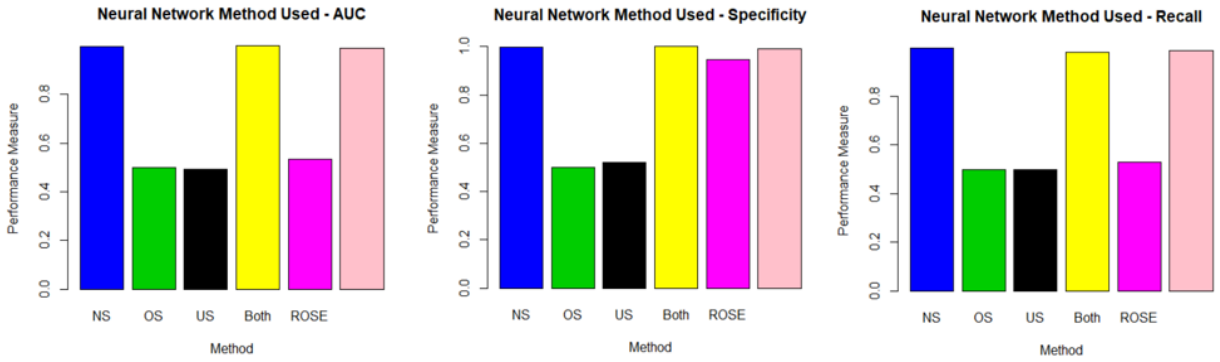
Boosting	NS	OS	US	Both	ROSE	SMOTE
Recall	0.825	0.831	0.880	0.867	0.847	1
Specificity	0.999	0.982	0.972	0.978	0.975	1
AUC	0.913	0.965	0.979	0.981	0.945	0.997



Performance Measures for Boosting

Neural Networks:

Neural Networks	NS	OS	US	Both	ROSE	SMOTE
Recall	0.998	0.499	0.498	0.982	0.529	0.989
Specificity	0.998	0.499	0.519	1.0	0.947	0.989
AUC	0.998	0.500	0.494	0.999	0.535	0.989



Performance Measures for Neural Networks

References

- Analytics Vidhya Content Team. *Practical Guide to deal with Imbalanced Classification Problems in R*. March 28, 2016. https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalanced-classification-problems/ Accessed: 10 Nov. 2017.

- Altini, Marco. *Dealing with Imbalanced Data: Undersampling, Oversampling and Proper Cross-Validation.* 17 Aug 2015. https://www.marcoaltini.com/blog/dealing-with-imbalanced-data-undersampling-oversampling-and-proper-cross-validation_ Accessed: 10 Nov. 2017.

- Crouch, Michelle. *Poll: Fraud alert false alarms common*. May 19, 2015. https://www.creditcards.com/credit-card-news/fraud-alert-blocked-poll.php. Accessed: 10 Dec. 2017.

- Javelin Strategy & Research. *Identity Fraud Hits Record High with 15.4 Million U.S. Victims in 2016, Up 16 Percent According to New Javelin Strategy & Research Study.* 01 Feb 2017. https://www.businesswire.com/news/home/20170201005166/en/Identity-Fraud-Hits-Record-High-15.4-Million. Business Wire. Accessed: 10 Dec. 2017.

- Lexisnexis. *The True Cost of Fraud*. 2016. https://www.lexisnexis.com/risk/downloads/assets/true-cost-fraud-2016.pdf. Accessed: 10 Dec. 2017.
- Marks, Gene. *Credit card fraud? These companies think they've solved that problem*. 28 Nov 2017. https://www.washingtonpost.com/news/on-small-business/wp/2017/11/28/credit-card-fraud-these-companies-think-theyve-solved-that-problem/?utm_term=.90ac8db7a882 The Washington Post. Accessed: 10 Dec. 2017.

- The Impact of Imbalanced Training Data for Convolutional Neural Networks
- https://www.kth.se/social/files/588617ebf2765401cfcc478c/PHensmanDMasko_dkand15.pdf