Credit Card Fraud Detection

Shreya Jain, Harshini Mashruwala and Aarjav Dutt

College of Engineering
Northeastern University

***Abstract* - Credit card fraud is a growing problem worldwide which costs billions of dollars per year. Advanced classification methods provide the ability to detect fraudulent transactions without disturbance of legitimate transactions and unnecessarily spent resources on fraud forensics for financial institutions. This paper discusses automated credit card fraud detection by means of machine learning. The aim of the project is to compare the results obtained by various classifiers (Logistic regression, Synthetic Minority Oversampling Technique (SMOTE), Random Forest) and also by various deep learning modeling techniques to achieve the maximum accuracy to predict fraudulent credit card transactions. We'll be trying to scale and resample the imbalance dataset and find the best algorithms that identifies the unusual patterns that do not conform to expected behavior. One way to achieve this is by oversampling, which is adding copies of the under-represented class. Another is undersampling, which deletes instances from the over-represented class. We would be using various different additional features such as feature selection, under-over sampling of data, supervised-unsupervised modeling, scaling etc and note the accuracy.**

Introduction

In 2015, credit, debit and prepaid cards generated over $31 trillion in total volume worldwide with fraud losses reaching over $21 billion [1]. In that same year there were over 225 billion purchase transactions, a figure that is projected to surpass 600 billion by 2025 [2]. Fraud associated with credit, debit, and prepaid cards is a significant and growing issue for consumers, businesses, and the financial industry. Historically, software solutions used to combat credit card fraud by issuers closely followed progress in classification, clustering and pattern recognition [3, 4].

The structure of this paper is as follows:

First we introduce the readers to the domain of credit card fraud detection. Later, explain various classifiers as well as deep learning modeling techniques followed by conclusions

Credit Card Fraud

Today, most Fraud Detection Systems (FDS) continue to use increasingly sophisticated machine learning algorithms to learn and detect fraudulent patterns in real-time, as well as offline, with minimal disturbance of genuine transactions [5].

Generally, FDS need to address several inherent challenges related to the task: extreme unbalanced ness of the dataset as frauds represent only a fraction of total transactions, distributions that evolve due to changing consumer behaviour, and assessment challenges that come with real time data processing [6]. For example difficulties arise when learning from an unbalanced datasets as many machine intelligence methods are not designed to handle extremely large differences between class sizes [5].

Also dynamic trends within the data require robust algorithms with high tolerance for concept drift in legitimate consumer behaviours [7].

Although specialized techniques exist that may handle large class imbalance such as outlier detection, fuzzy inference systems and knowledge based systems [8, 9], current state-of the-art research suggests that conventional algorithms in fact may be used with success if the data is sampled to produce equivalent class sizes [10, 11, 12]. This is valuable as it means a wider range of, in some cases off-the shelf, typical classification algorithms may be used which mitigates existing algorithmic limitations due to extreme class imbalance. Not only can fraud detection capabilities potentially increase due to a larger scope of potential methods, the cost of development can be decreased due to reduced reliance on highly specialized niche methods, expert systems, and continued research into algorithmic methods which handle class imbalance directly.

Documentation

Dataset: The datasets contains transactions made by credit cards in September 2013 by european cardholders. This dataset present 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, we 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.

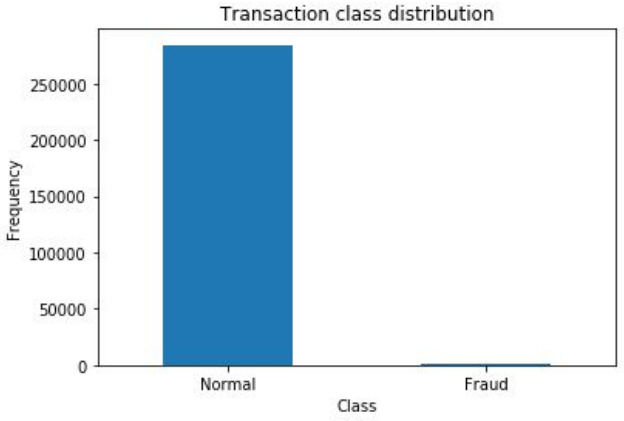

Fig 1: Dataset distribution

Methods

Logistic Regression

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables[13].

Oversampling

Oversampling replicates the observations from minority class to balance the data.[14] The usual reason for oversampling is to correct for a bias in the original dataset. In our project we have used the SMOTE technique to perform Oversampling.

SMOTE

In Synthetic Minority Oversampling Technique, the minority class is over-sampled by creating ``synthetic" examples rather than by over-sampling with replacement.[15].

Random Forest

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 mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.[16]

Undersampling

Undersampling reduces the number of observations from majority class to make the data set balanced. This method is best to use when the data set is huge and reducing the number of training samples helps to improve run time and storage troubles.[14]

Autoencoder

An autoencoder is an artificial neural network used for unsupervised learning of efficient codings. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for the purpose of dimensionality reduction.

By inputting data and abstracting the data structure, the multidimensional or high-dimensional data is compressed into low-dimensional data representations, the encoder is encoded, and then the compressed and reduced-dimensioned data input is used as a decoder for representation, so that it more accurately represents the original output results[17]. Here's a visual representation of what an Autoencoder might learn:

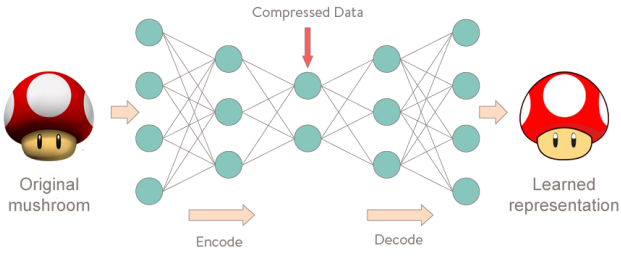

Fig. 2: Visual representation of autoencoder

Results

The first technique we used is *logistic regression*. Using the regression on the unscaled data we got an accuracy of 80% (Fig. 3).

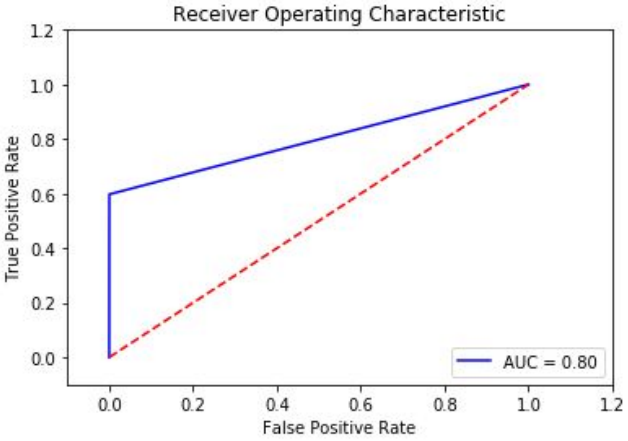

Fig. 3: ROC curve for logistic regression

The next technique we used is *Oversampling*. We used SMOTE on the oversampled train data to obtain a balanced train data set. After implementing SMOTE on our dataset, we performed training of *random forest* on the oversampled data. By doing so, we increased the accuracy to over 90%. Further, we would perform random forest on the undersampled data as well.

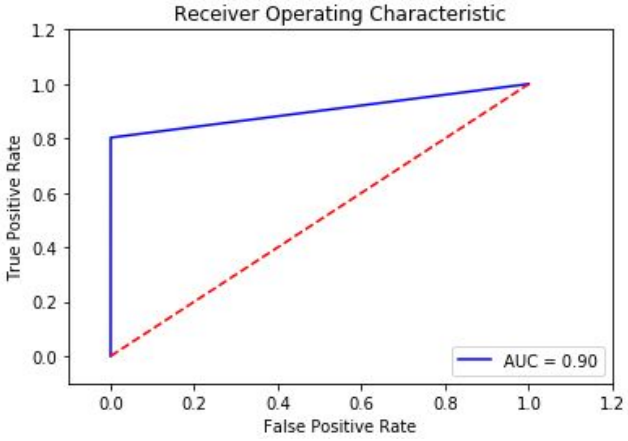

Fig. 4: ROC curve for Oversampled data after SMOTE and Random Forest

For *scaling*, We have first scaled the columns of Time and Amount. There are 492 cases of fraud in our dataset so we can randomly get 492 cases of non-fraud to create our new sub dataframe. We

concat the 492 cases of fraud and non fraud, creating a new sub-sample. Undersampling being the most used and efficient method, performed very well here as well. We got an accuracy of 93%.

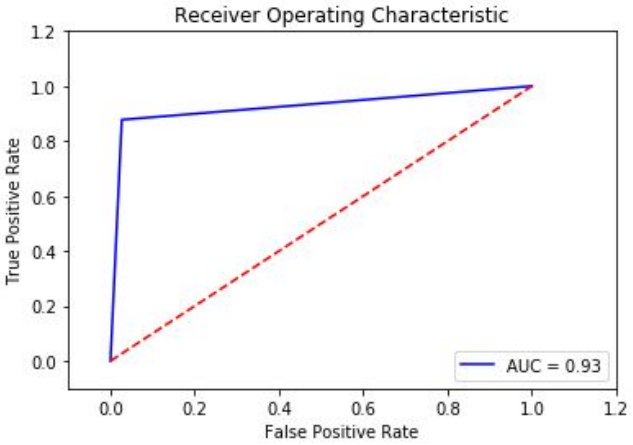

Fig. 5: ROC curve for Oversampled data after Random Forest

We have also performed logistic regression on scaled data, to compare how well it performs in comparison to the baseline results. After scaling, we could see the class column was equally distributed.

Fig. 6: Scaling data on 'Class'

As the below figure shows, the AUC curve improved a lot as compared to any of the methods used above. We got an accuracy of around 98%.

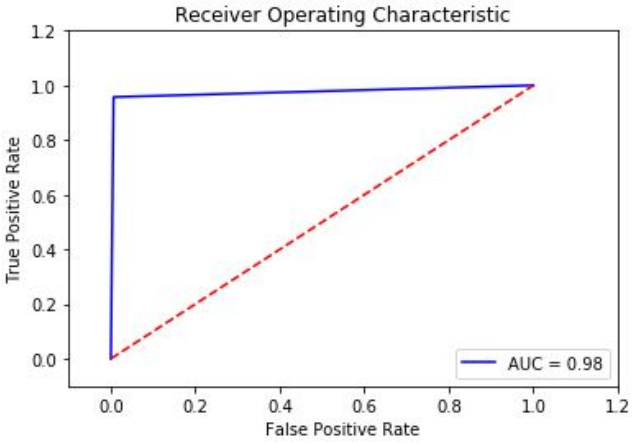

Fig. 7: ROC curve for logistic regression after scaling

Clustering with t - Stochastic Neighbor Embedding. t-SNE algorithm can pretty accurately cluster the cases that were fraud and non-fraud in our dataset. Although the subsample is pretty small, the t-SNE algorithm is able to detect clusters pretty accurately in every scenario This gives us an indication that further predictive models will perform pretty well in separating fraud cases from non-fraud cases.

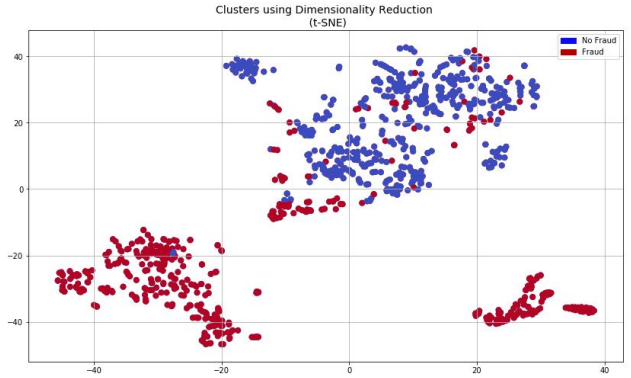

Fig. 8: t-SNE Clustering

Now, we have tried to implement 4 Different classifiers to determine which has higher accuracy. We tried comparing: Logistic Regression, K-Nearest neighbor, Support Vector Classifier and Decision Tree Classifier. Here are the accuracy results of these classifiers:

Logistic Regression: 0.999877260982

KNears Neighbors: 0.949664082687

Support Vector Classifier: 0.999954780362

Decision Tree Classifier: 0.99875

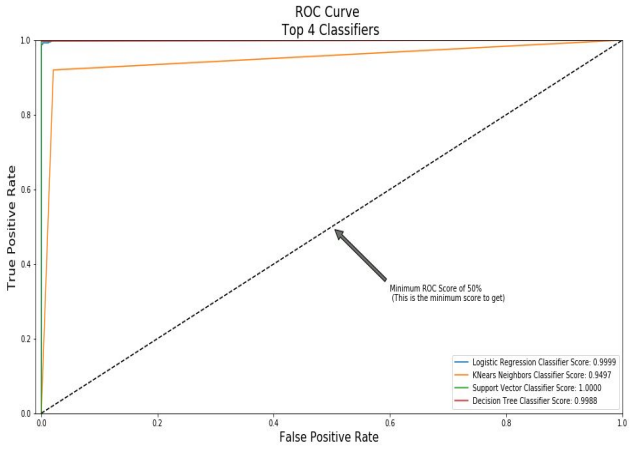

Fig. 9: ROC curve for 4 classifiers (Logistic Regression, K-Nearest neighbor, Support Vector Classifier and Decision Tree Classifier)

Next, We will try to implement a neural network to find out if the algorithm we used is able to predict an accurate fraud.

For the first case we will be implementing Auto-encoder as unsupervised learning. We trained and validated the model with 1 hidden layer, with 75:25 train to test ratio. We used TanH Activation function, RMSProp Gradient estimation, Gaussian Network initialization and 10 epochs. We got an accuracy of 96%.

Next, we implemented Auto-encoder on 80:20 train to test ratio. We used Cross Entropy Cost function, truncated_normal Network initialization, ADAM Gradient estimation and 80 epochs. We got a similar accuracy of around 95%.

We tried to implement a neural network without using auto-encoder this time, and using 2 hidden layers. We used Softmax and Relu with ADAM Gradient estimation and 50 Epochs. Our accuracy increased to around 97%.

We also tried to change the previous method, but with hinge loss entropy and using 70 epochs. The accuracy decreased to 88%.

The results show that after implementing all the algorithms and classifiers, the best accuracy was given by Logistic Regression on the scaled data with an accuracy of 98%. On implementing deep learning on our data, we got the second best accuracy of 97% from Softmax and Relu with ADAM Gradient estimation Optimizer.

Discussion

We used various predictive models to see how accurate they are in detecting whether a transaction is a normal payment or a fraud. As described in the dataset, the features are scaled and the names of the features are not shown due to privacy reasons. Nevertheless, we can still analyze some important aspects of the dataset.When we started the project, our goals were:

-To understand the little distribution of the data that was provided to us.

-Over sampling and under sampling of skewed data

-Perform SMOTE and random forest.

-Feature selection using Gaussian Distribution Model, to check the accuracy

-Create a 50/50 sub-dataframe ratio of "Fraud" and "Non-Fraud" transactions with scaled data.

-Determine the Classifiers we are going to use and decide which one has a higher accuracy

-Create a Neural Network and compare the accuracy to our classifier

Approach

First, we performed the EDA on our dataset to remove bad data. There are no "Null" values, so we don't have to work on ways to replace values. Most of the transactions were Non-Fraud (99.83%) of the time, while Fraud transactions occurs (0.17%) of the time in the dataframe.Then we performed logistic regression on unscaled and highly skewed data, auc achieved was 80%. To balance the dataset, we performed oversampling using SMOTE and undersampling and performed Random forest algorithm on the same. Trained the model on the training data. Made predictions on the test data and got accuracy of 90% and 93% respectively, proving that undersampling was effective than oversampling. We also tried to do logistic regression on a scaled

data set, which improved the accuracy to 99% which is the highest accuracy we have obtained in our project.

We implemented one of the deep learning models in our project, called the autoencoder, to see how the highly efficient deep learning neural network can help us in predicting fraud. We tried to use different activation functions, cost functions and changed the epochs and gradient estimations on both 1 layer and 2 layer neural network models. We got different accuracy results in each and every case. After applying several combinations of these parameters, we reached a conclusion that the maximum accuracy was given by Softmax and Relu with ADAM Gradient estimation Optimizer. Although the 97% accuracy mark was still lower than the 99% mark reached by Logistic regression.

Code with Documentation

The technical implementation of the research project is provided on github:

https://github.com/aarjav-dutt/Advance-Data-Science

References

[1] "The Nilson Report," HSN Consultants., Carpinteria, CA, Issue 1096, Oct. 2016. [Online] Available: https://nilsonreport.com/publication_newsletter_archive_issue.php? issue=1096

[2] "The Nilson Report," HSN Consultants., Carpinteria, CA, Issue 1101, Jan. 2017. [Online] Available: https://nilsonreport.com/publication_newsletter_archive_issue.php? issue=1101

[3] A. D. Pozzollo, "Adaptive Machine Learning for Credit Card Fraud Detection," Ph.D. dissertation, Dept. Comp. Sci., Univ. Libre de Bruxelles, Bussels, Belgium, 2015.

[4] L. Delamaire, H. Abdou, and J. Pointon. Credit card fraud and detection techniques: a review. Banks and Bank Systems, 4(2):57–68, 2009. [5] A. D. Pozzolo, et. al., "Calibrating Probability with Undersampling for Unbalanced Classification" in Symposium on Computational Intelligence and Data Mining (CIDM), 2015 © IEEE. doi: 10.1109/SSCI.2015.33

[6] A. D. Pozzollo, et. al., "Learned lessons in credit card fraud detection from a practitioner perspective", submitted for publication in Expert Systems with Applications, Feb 2014.

[7] A. D. Pozzollo, "Credit Card Fraud Detection and Concept-Drift Adaptation with Delayed Supervised Information" in International Joint Conference on Neural Networks (IJCNN), 2015 © IEEE. doi: 10.1109/IJCNN.2015.7280527

[8] M Krivko. "A hybrid model for plastic card fraud detection systems". Expert Systems with Applications, 37(8):6070–6076, 2010.

[9] Y. Sahin, S. Bulkan, and E. Duman. "A cost-sensitive decision tree approach for fraud detection". Expert Systems with Applications, 40(15):5916–5923, 2013.

[10] H. He and E. A Garcia. "Learning from imbalanced data. Knowledge and Data Engineering", IEEE Transactions on, 21(9):1263–1284, 2009.

[11] G. Batista, A. Carvalho, and M. Monard. "Applying one-sided selection to unbalanced datasets" in MICAI 2000:

[12] Kaggle. (2017, Jan. 12). Credit Card Fraud Detection [Online]. Available: https://www.kaggle.com/dalpozz/creditcardfraud

[13] https://www.statisticssolutions.com/what-is-logistic-regression/

[14] https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalanced-classification-problems/

[15] https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume16/chawla02a-html/node6.html#SECTION00042000000000000000

[16] https://en.wikipedia.org/wiki/Random_forest

[17] https://en.wikipedia.org/wiki/Autoencoder