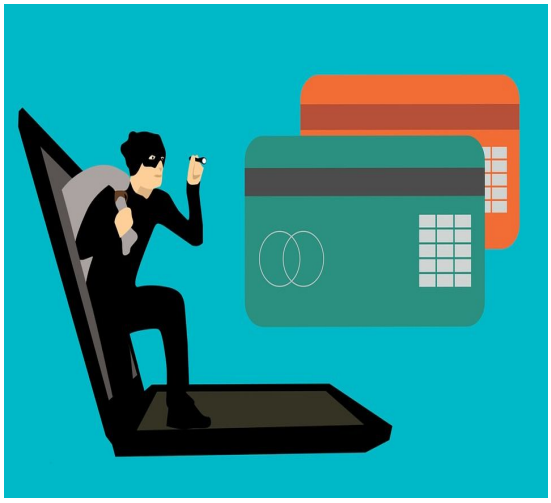
Machine Learning For Credit Card Fraud Detection

Team: The Predictators



Team Members:

Name	Username
Sindhuja Kasula	skasula
Vrushali Mahuli	vmahuli
Akshay Babu	ababu1
Abishek	a5

Introduction

Increase in digitization, especially in the financial industry has resulted in almost everyone using payment cards instead of cash. In this digitized era, where currency is transacted dynamically in cashless banking finance, it is an important factor for banking systems to see that fraudulent transactions through a credit card are at their least. Today it is easier than before to transact a fraud, as a lot of transactions are done over the phone, or internet. As of 2015, these types of frauds account for 60% of the total credit card frauds[5]. Two major cases of Illegal transactions on credit cards are, one where the card is stolen and transactions are made by someone posing as the card owner which is known as ‘External card fraud’ and the other is ‘Inner card fraud’ which occurs as a result of consent between cardholders and bank by using false identity. ‘External card fraud accounts for the majority of credit card frauds. Detection of these dishonourable transactions has become a significant activity for credit card payment process.

S.Xuan [3] states there are two kinds of methods for fraud detection - misuse detection and anomaly detection. The first one is used to know about the existing types of fraud to make models by learning the various fraud patterns, while the other one is to build the profile of normal transaction behavior of a cardholder based on his/her historical transaction data, and decide on a new transaction as a potential fraud if it deviates from the normal transaction behavior.

Motivation

We all know someone who has had fraudulent transactions on their card. Also, there have been multiple times, when a legit transaction is stopped, because it looks like a fraud. We wanted to see if we could improve the accuracy to minimize such situations. We wanted to work on a real time problem in finance industry and this seemed like a serious problem that has increased over time with the advancement in technology.

Problem Statement

Our goal is to implement machine learning classification algorithm which classifies the fraudulent and non-fraudulent transactions of a credit cards by gaining insights from data visualization, and test our preliminary observations using different classification algorithms and understanding the results.

Dataset

The dataset which we intend to use is available on Kaggle as part of a competition[9], where Vesta corporation (e-payment service provider) provided a feature rich dataset to help getting good solutions to prevent fraud as it has information of card holder's spending profile and transactions. Some of the main features in the dataset are: Transaction Id, device type, amount, date, time, product code, payment card information, associated cards, address, time between transactions, and target variable is a boolean indicating whether the transaction was fraud or not. But most of the features are desensitized for privacy reasons.

Challenges

Our first challenge was to find the data that has good enough samples. We found the dataset above on kaggle after thorough research. This dataset has about 400+ features and 50K+ samples, which was our next challenge to find an effective platform to run our code, save and collaborate with each other. We used google colab's GPU effectively to run our program and github to save and collaborate. After initial profiling we found there were many missing values in the data, and we needed an effective technique - we found KNN to work best for our data. Given there is only one fraudulent transaction out 1000 transactions, the target classes were highly disproportionate, we used SMOTE to overcome this. Finding the hypothesis to the problem rather than improving the accuracy was a major challenge.

Concise Summary of our Approach

We created ID's to identify unique cards in this transaction oriented dataset. From data visualization, we observed that if a card can be compromised once then it can be compromised multiple times. We used this as our base hypothesis and flagged the compromised cards. Using KNN for imputation of the null/missing values and PCA for reducing the number of features in the dataset, we reduced the dataset into one that could efficiently be used for machine learning models. We compared the results of classification on 4 machine learning algorithms: Naive Bayes, GBM, Neural Networks and LDA by using the SMOTE technique for sampling the data.

Backgrounds/Related Work

Literature Survey

The research paper by Ashphak[1] uses K-means to cluster classes and then applying Hidden Markov Model which studies the spending of card holder and detect fraud did not work well. Though it was a good idea to use uncertainess as a factor using HMM algorithm, this algorithm classifies all the transactions whose probability is lower than sample as fraudulent transactions but they may be genuine transactions made by card owner. These one off transactions made by owners which differ highly from general spending are actually outliers, which were not taken into consideration which resulted in high false positive rates.

Another approach by A. C. Bahnsen[5] was to detect fraud using periodic features. While a lot of transactional features are used with credit card detection, periodic features were created by analyzing the behavior of the user and the time of the transactions using von Mises distribution(this is a close approximation to wrapped normal distribution). The accuracy of the fraud detection was improved by 13% using this approach. When creating a fraud detection model, the customer's spending pattern is analyzed, by transaction type, country, amount spent etc. Dissimilar continuous transactions, country and amount of transactions and time between them are taken into consideration. To further improve accuracy, the model was rated by using a cost matrix, that takes into account the amount of transactions. For cost sensitive classification, Bayes minimum risk(BMR), cost sensitive logistic regression and cost sensitive decision tree(CSDT) can be used, and CSDT seems to perform the best.

As there are always new ways for fraudulent transactions, it makes sense to try and apply machine learning algorithms with online/ real time data. One class support vector machine (OCSVM) and T^2 control chart can be used. Slack variables can be used for handling outliers in the training data. Using this method gives a higher accuracy and lower false alarm rates (Tran[4]).

Another research paper by S. Xuan[3] used ‘misuse detection’ to detect credit card fraud. In misuse detection, the model already knows about the existing types of fraud. The authors used two types of random forest to train the model for learning fraud behavior, first type being random tree based random forest and another being CART (Cart based random forest tree) based.
In random tree based random forest, instead of using a single tree model which has a possibility of overfitting, the authors used ensemble approach which is a combination of multiple trees and each tree is dependent on a random dataset that is independent. The training set for each tree was a bootstrapped sample randomly selected from the standard training set. The other random forest algorithm used in this research paper is the cart based random forest tree, where training set comes from a collection of bootstrapped samples. And at each node the data is split using the best attribute in a subset of attribute according to Gini impurity which measures the

uncertainty of the data set. In the first algorithm, the data is distributed by the distance among instances, this is fast in computing the centers but slow in distribution of data and in CART the data is distributed from attribute which has minimum Ginni impurity, this makes it slow in computing the Ginni impurity and fast in distributing the data. They found the second algorithm (CART) very useful as it had a better accuracy, recall and F-measure, even though the precision was a bit less. The major problem is the imbalanced data and the assumption that each base classifier has equal weight.

Liu[6] mentions about Isolation forest algorithm where the partition are created by first randomly selecting a feature and then splitting the value between min and max randomly to find outliers. Also it does not require distance or density for detecting the anomaly making it is very cost efficient, this also means that it requires less computation. This method could be useful for large datasets as it can be scaled up to many features.

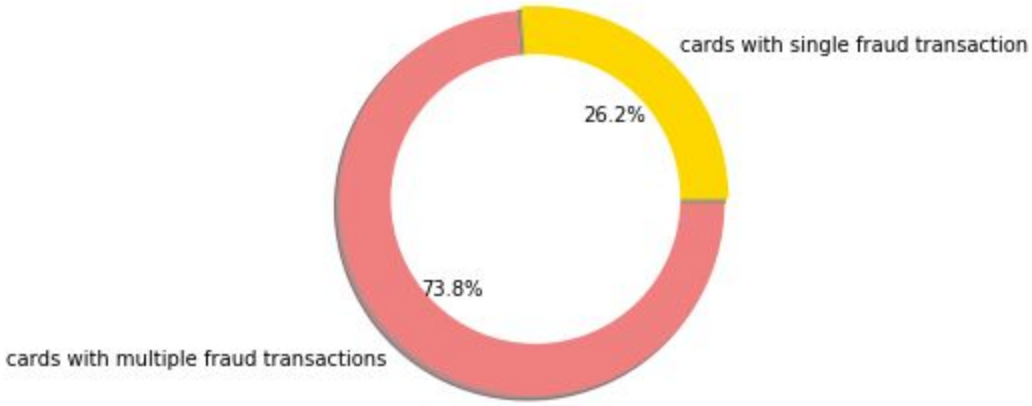
Research paper by J. O. Awoyemi [2] investigates the effect of hybrid sampling which is a combination of under-sampling and oversampling on performance of Naive Bayes, k-nearest neighbour and logistic regression classifiers to detect fraud experimenting on a highly skewed credit card fraud data. The positive class (fraud transaction) is over-sampled and negative class (legitimate transaction) is under-sampled. A hybrid of under-sampling and oversampling is carried out on the highly unbalanced dataset to achieve two sets of distribution (10:90 and 34:64) for analysis. All the above mentioned algorithms are trained and tested. The performance metrics used are accuracy, sensitivity, specificity, precision, Matthews correlation coefficient and balanced classification rate. All the classifiers showed improvement in performance when hybrid sampling was used. The Logistic regression classifier showed the least performance among the three classifiers evaluated, K-NN and Naive Bayes show competitive results. From this, we understood that sampling techniques have huge impact on the performance of anomaly detection problems. The sampling techniques mentioned in this paper will be used in our approach because we think it had a huge impact on the experiment mentioned in the paper.

Method

Our Hypothesis(Novelty)

Our hypothesis was to assume that a card, if compromised once; could be compromised again. As our dataset only had transaction details, there was no direct way to identify a unique card in the transactions. For this, we had to combine a few columns of the dataset to get a unique card id. We used, the card type, purchaser email domain, bank type and the day the card was issued; to get the unique card. We subtracted the days before the first transaction to get the value of the date, when the card was issued. (Date values were not available) in the dataset.
After getting the unique ID of the cards, we added a new column, with a flag set for the compromised cards.

We checked our assumption by getting the number of cards with multiple fraudulent transactions and found that around 74% of the fraudulent cards had been used more than once for a fraud transaction. We therefore, decided to go forward with this hypothesis and check the results that the new column would have to our model. Below is the data visualization which suggests the same.



Pre-processing

To begin with, we used pandas profiling to find similar columns. The results weren't conclusive. Most of the columns had missing values, and correlation among the columns was not high, so we did not drop any columns at this point.

As mentioned in the dataset section above, we had two datasets - one for transactions and other for identities and we joined both the datasets by left join to gain insights about the identity of the card holder as well. After joining the 2 datasets of user identity and transaction details, our

overall features in the dataset were more than 400. Out of this 14 columns were categorical. As the number of features was already large, we used label encoder to encode these columns.
To decrease the features, we removed the columns that had more than 90% null values. We also removed the user id columns, as the details about them were unclear and we wanted to focus on the cards used in the transactions rather than the users themselves. The only user specific columns that we used were the ones which had information related to the type of device used during the transactions.After removing the null value columns, we still had around 350+ features in our dataset.

Instead of general imputation techniques like substituting with zeros or average of the columns, we used the K-nearest neighbor method, for imputing values. We used 6 nearest features and 5 iterations to impute the missing values. We rounded off the values for categorical columns. Even though the details and meanings of the columns were encoded, columns of the same category had similar names. We therefore, used PCA decomposition, to then reduce the size of similar columns to 1. After decomposing these columns we were finally left with 25 features.
To further decrease the space requirements of the dataset, we downsized all the default float value columns to int. Finally we used MinMax normalization, to normalize the columns.

Only 3% of our data, belongs to the fraud class. This created a problem of having a highly imbalanced dataset. After some research, and analyzing the data, we decided to use SMOTE sampling technique to oversample the minority class and generate synthetic data. Even though the data of the minority class had a lesser percentage, it was still large to accurately represent the class, if synthetic data was made.

Building a machine learning model

The first algorithm we used was Linear Discriminant Analysis (LDA). LDA uses the assumption that the resultant class of test sample would be the one which has max value for discriminant analysis.

$$y = arg\ max\ (\delta 1(x),\ \delta 2(x))$$

Assuming there are two classes k = 1, 2

The discriminant function for a class is defined as

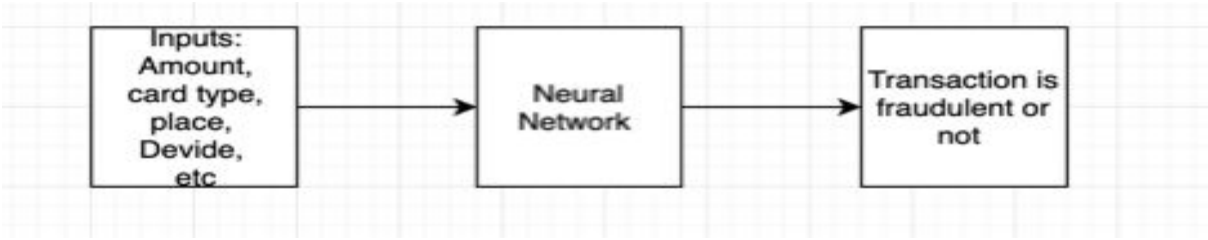
$$\boldsymbol{\delta} = -0.5\ ln(\boldsymbol{\Sigma}) - 0.5\ ((x - \boldsymbol{\mu}).T)\ inv(\boldsymbol{\Sigma})\ (x - \boldsymbol{\mu})) + ln\ P$$

Where is the covariance matrix of both classes. is mean for class k and P is probability for class k. Since covariance is the same for both classes. Let us say there is a sample x which actually belongs to a minority class, is evaluated through LDA, and the discriminant function value is calculated for both classes, in case of majority class, the value x- will be large, since it does not belong to to majority class, and the value of Probability will also be more as the

majority class as more samples, so the last two terms cancel each other minimizing the discriminant function, compared to minority class. And since this is a probabilistic model, we think it would work well on rebalanced dataset.

The next algorithm that we used is Gradient Boosting. Currently Gradient Boosting Algorithms are highly popular. Decision trees are easily understandable and have a better performance on imbalanced data[11]. Therefore, we believe that a boosting algorithm for decision trees will give great results for this dataset. We would like to use Light GBM, because it is fast, uses less memory and is more accurate[10].

Lastly we wanted to use neural networks, because it predicts whether the transaction is fraud or not based on the associative memory of patterns it had learned from the past. Neural networks has the benefit of being highly scalable and being able to work with non linear inputs and outputs, this makes it a good candidate for being used for comparison. One advantage that Neural network has over other models is that it learns from the past and improves results as time passes. By using this method, the organization or banks can detect fraudulent transaction in an efficient way. Our model will have output layer, hidden layer and an input layer. Over here the inputs are given, and based on the output; error is calculated. The weights are then adjusted accordingly. This process is repeated until it converges.

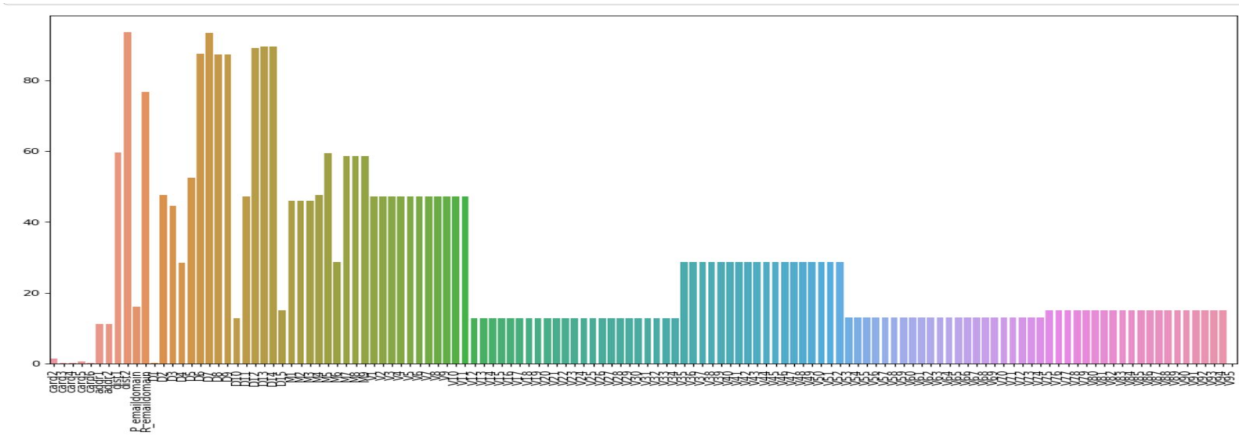


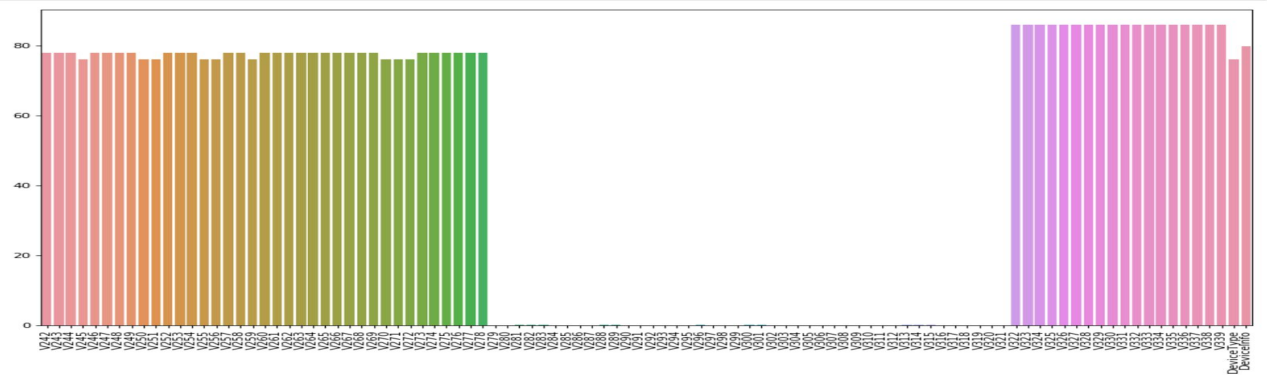
For all the algorithms mentioned above for our approach, we will partition the data in the standard 80-20 for train and test sets. Run the model, evaluate the model. We would use Area under curve metric of Precision Recall curves as they summarize the trade-off between the true positive rate and the positive predictive value for a predictive model which makes them appropriate for imbalance datasets..Finally, we compared our results with Naive Bayes algorithm to use it as a benchmark.

Experiments

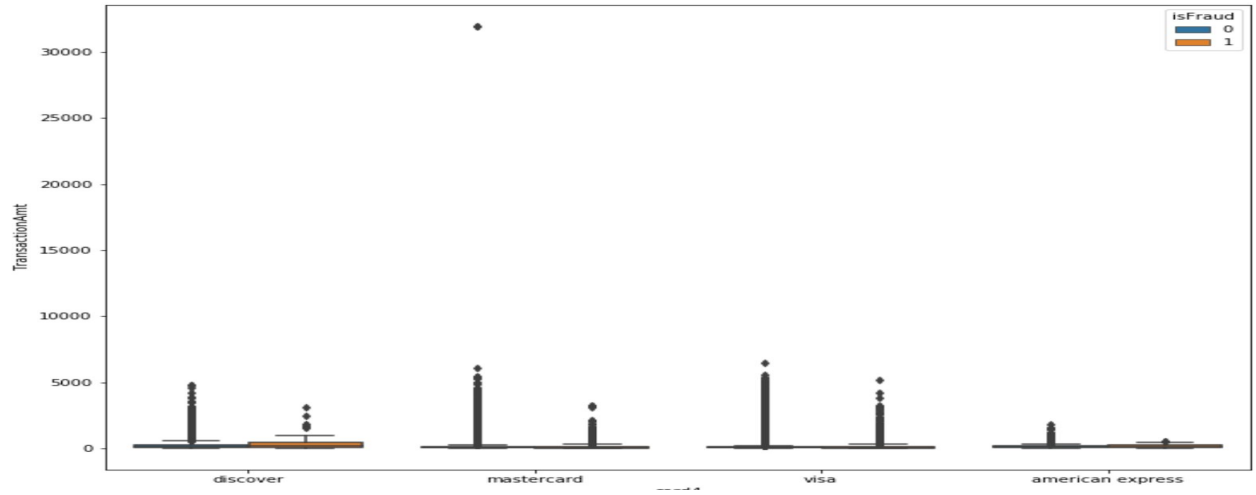
We started our experiment with data preprocessing and visualization. We joined our both datasets, dropped columns which had more than 90% of null values.

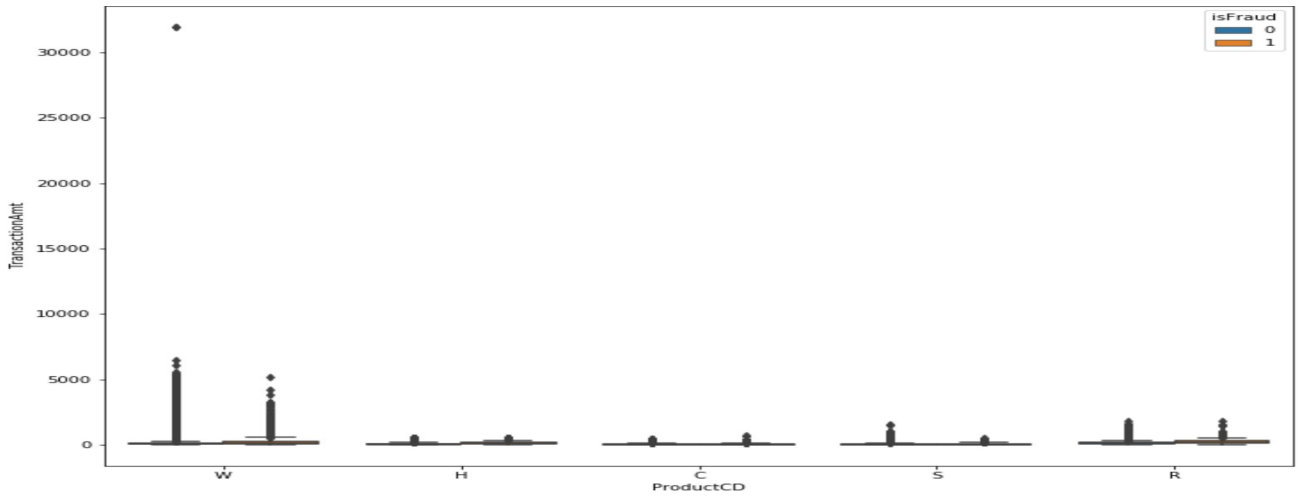
Understanding the columns.





Trying to find relationship between card type, product type and fraud

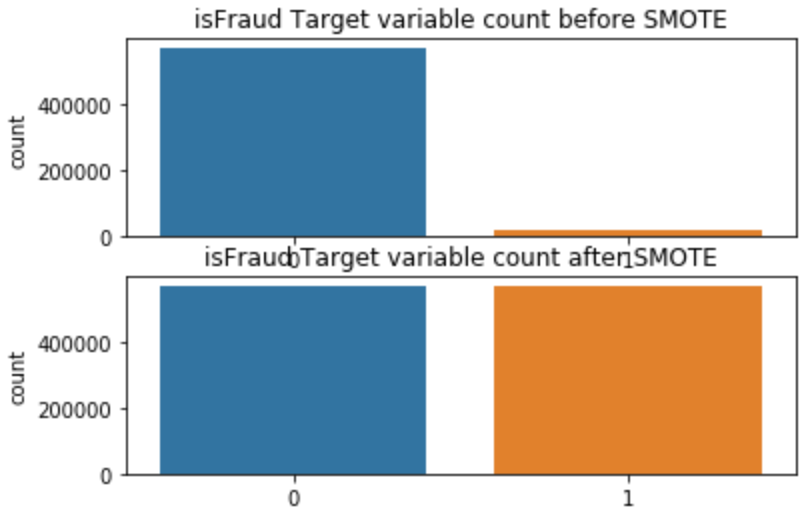




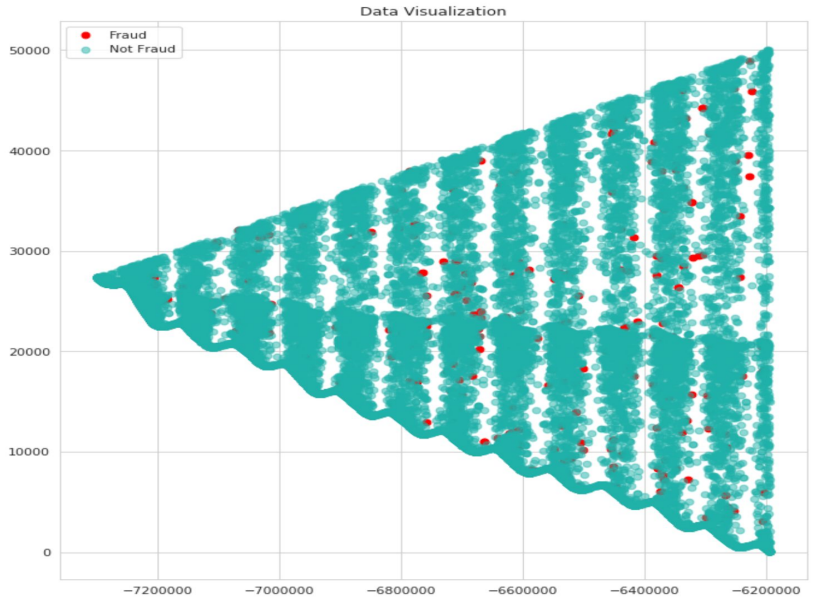
And then performed imputation with KNN by using 6 nearest features and 5 iterations. We then downsized the datatypes so it would use less memory. We extracted new features required for our hypotheses, the fraud_card column.We performed on label encoding on categorical columns. We used MinMax scaling to standardize the dataset and lastly combined the similar types of columns with PCA. The below are the list of columns before and after imputation with KNN, label encoding and PCA.

We used the sampling technique of SMOTE to overcome the dataset imbalance challenge. In this the minority class is oversampled, and synthetic data is generated.

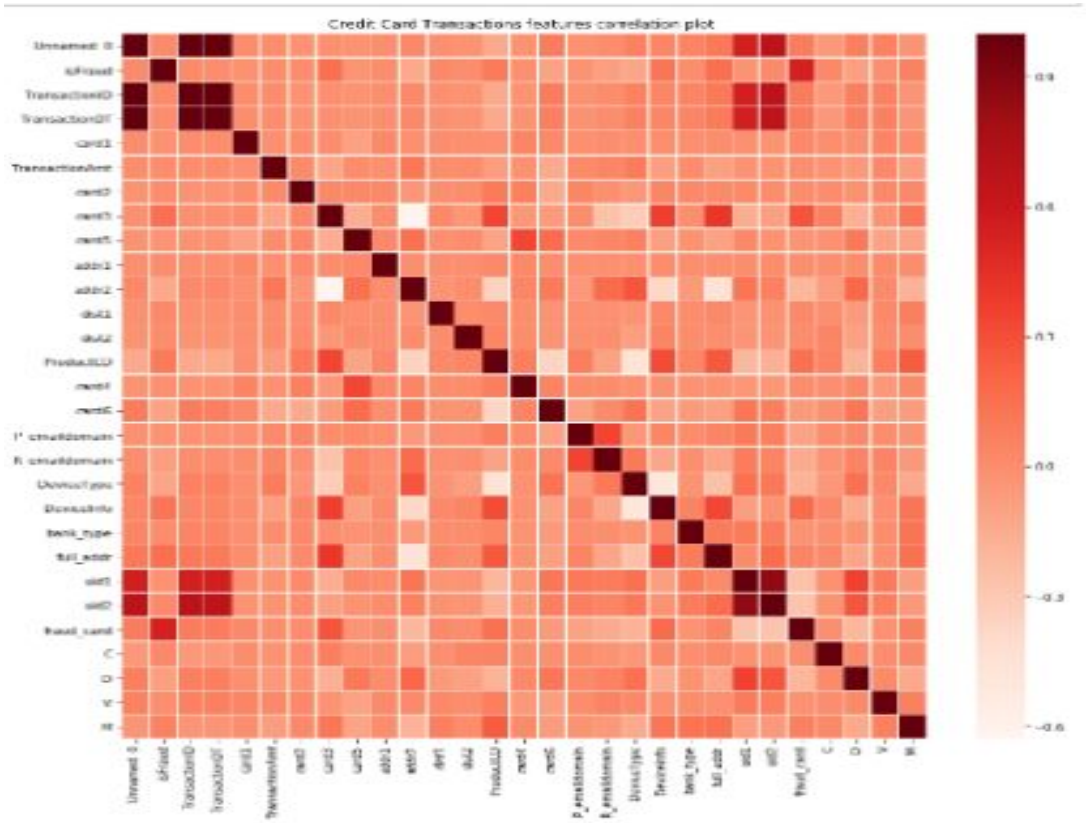
Data after SMOTE sampling.



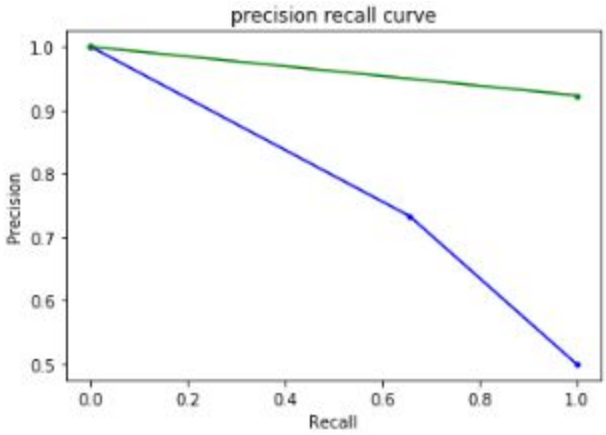
We decomposed all columns using PCA into 2 dimensions, and tried to plot a graph of the data, to see if any obvious patterns were visible. But, no obvious differences were visible.



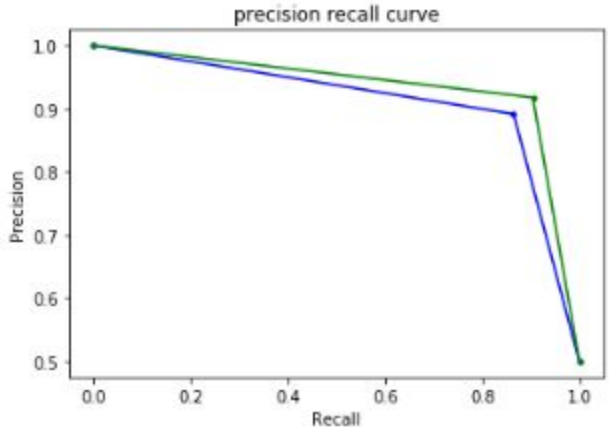
Visualizing the correlation among the features



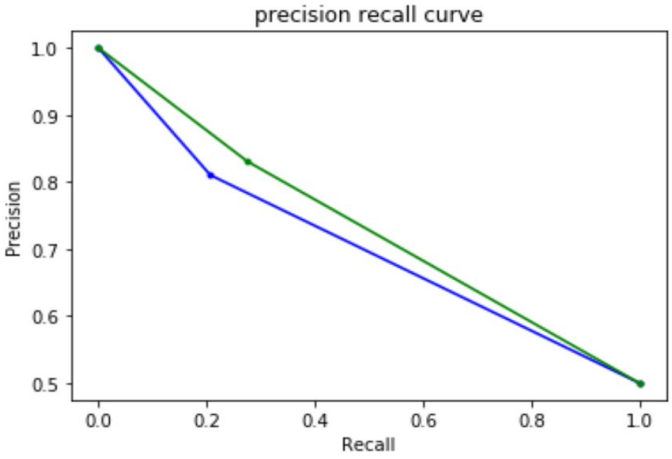
After all the preprocessing, we applied algorithms that we mentioned in the method. We split the dataset into training and test sets for both the data - one with our columns which we generated to support our hypothesis and another without those columns. On applying LDA we observed a significant change in results as seen in the precision recall curve. with our hypothesis.The blue line plots the curve, when the added features wasn't used and the green line plots the curve when the added column was used for training.



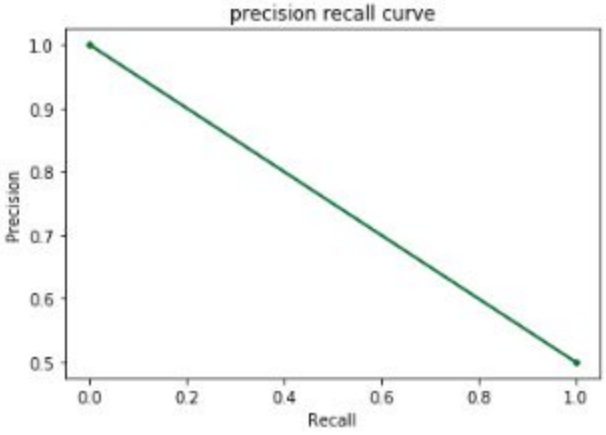
On applying GBM, the following results were observed. We can see that the column did not have as much of an effect, as it had while using LDA. This might be because GBM already had a better performance.



Surprisingly, Naive Bayes which was claimed to have great results[2], as a classifying algorithm during our research did not reach the impressive standard set by LDA for this particular problem.



Lastly we applied a deep neural network, with 5 layers which consists of 3 dense layers of 8 units each and a dropout layers with 2 units for regularization. And a dense layer of 2 units for output. We used “adam” optimiser and “categorical cross entropy” for loss which is negative log loss as model configuration. For neural network, we further split train data into training and validation set.To prevent overfitting, we also implemented early stopping to monitor validation loss, and stops after 2 iterations from best saved model.



Summary of Results

Algorithms	Accuracy	Precision	Recall	F1-score	AUC score
Naive Bayes	58.03%	20.89%	81.28%	33.24%	56.55%
Naive Bayes**[with our hypothesis]**	61.04%	27.65%	83.29%	41.52%	59.22%
Linear Discriminant Analysis	70.87%	65.65%	73.32%	69.27%	65.32%
Linear Discriminant Analysis**[with our hypothesis]**	95.84%	100.00%	92.33%	96.01%	92.33%
Neural Network	49.98%	-	-	-	50.02%
Neural Network**[with our hypothesis]**	50.02%	100.00%	50.02%	66.68%	50.02%
Gradient Boosting Machine	87.97%	86.48%	89.14%	87.79%	83.85%
Gradient Boosting Machine**[with our hypothesis]**	91.15%	90.43%	91.76%	91.09%	87.77%

Analysis of the result

From the results, we observed that LDA worked the best. This might be because LDA uses covariance in the discriminant function. From the correlation matrix in the visualization, it is seen that one of the new columns which we added “fraud_card”, which tags if that card was compromised had high correlation with the target variable. GBM worked better even without our hypothesis as expected since it uses decision trees making predictions and this algorithm also has a positive change with our hypothesis as observed in the results above. Naive bayes also support our hypothesis and we are using this one to benchmark our results. Shockingly, deep neural network performed poorly even with added regularization and early stopping with 8 neurons. We tried to improve the model parameters by changing the number of neurons to 128 but model kept crashing the colab notebook. Due to insufficient computing resources, we could find the optimised model configuration. And that is why we think the results are as expected.

Team Contributions

Task	Sindhuja Kasula	Vrushali Mahuli	Akshay Babu	Abishek
Survey of literature regarding fraud detection	✔	✔	✔	✔
Finding a good dataset to work with		✔		✔
Explore and analyse dataset		✔	✔	
Pre-process the data	✔			✔
Research on the best model to use	✔	✔		
Create a model to detect fraud	✔	✔		
Mid Project Report	✔	✔		
Verify effects of different sampling techniques				✔
Experimenting different algorithms	✔		✔	
Report creation of the project	✔		✔	
Poster creation			✔	✔

Reflections

Comment 1: Team name?

- We added the team name in this report.

Comment 2: This is not grading point. Remember suggesting a different approach is more Important.

- Answered in comment 4 and 5

Comment 3: Well the proposed method is still typical ML pipeline with different algorithm choice. What is your difference claim?

- Answered in comment 4 and 5

Comment 4: This does not give you enough difference

- Added a new column for flagging fraud cards based upon the fact that the fraud had happened earlier on that card. We assume the probability of fraud transaction happening again is high, so we added a new column and used in our machine learning model. And this is the new approach we implemented.

Comment 5: Focus on this. This is something you can claim different. Deeply examine the effect of adopting this new feature. (In reference to our hypothesis, that cards may have multiple fraud transactions)

- We experimented on the hypothesis, and the results are presented above.

Comment 6: Thanks for carefully answering my comments. Please make sure to highlight the new features and related analysis.

- We highlighted the new feature(Written in bold in our short approach, added the steps separately in the detailed approach) and analyzed results, with and without the new feature column.

Conclusion

Detection of fraud in a highly imbalanced classification data is a complex issue that required a significant amount of preprocessing before using algorithms for machine learning. Assuming and verifying the hypothesis of multiple fraud transactions made a huge difference in this project. We were surprised at how well this worked and improved the results.

Due to the fact, that the data was sensitive, the columns were encoded and the names were removed. It would have been easier to understand and implement a better model, if we had more insights about what the columns represented. As the dataset we had was of a limited timeframe, our models work fine. In the future periodic features must be accounted for. Different sampling methods can also be used, for comparison.

This project helped us understand the importance of pre-processing for better results. Visualization of the data helped in understanding the correlation better.

References & Citation

1. Ashphak Khan, Tejpal Singh, Amit Sinhal, "Implement Credit Card Fraudulent Detection System Using Observation Probabilistic In Hidden Markov Model", NUiCONE-2012, DECEMBER. 2012

2. J. O. Awoyemi, A. O. Adetunmbi and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," 2017 International Conference on Computing Networking and Informatics (ICCNI), Lagos, 2017, pp. 1-9. doi:10.1109/ICCNI.2017.8123782

3. S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang and C. Jiang, "Random forest for credit card fraud detection," 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, 2018, pp. 1-6. doi: 10.1109/ICNSC.2018.8361343

4. Tran, P. H., Tran, K. P., Huong, T. T., Heuchenne, C., HienTran, P., & Le, T. M. H. (2018). Real time data-driven approaches for credit card fraud detection. In Proceedings of the 2018 International Conference on E-Business and Applications, ICEBA 2018, ACM, New York, NY, USA, pp. 6– 9.

5. A. C. Bahnsen, D. Aouada, A. Stojanovic and B. Ottersten, "Detecting Credit Card Fraud Using Periodic Features," 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, FL, 2015, pp. 208-213. doi:10.1109/ICMLA.2015.28

6. Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on (pp. 413–422). IEEE.

7. Rafael Pierre (June 2018), Detecting Financial Fraud Using Machine Learning. Link: https://towardsdatascience.com/detecting-financial-fraud-using-machine-learning-threeways-of-winning-the-war-against-imbalanced-a03f8815cce9

8. Eryk Lewinson (July 2018), Outlier Detection with Isolation Forest. Link: https://towardsdatascience.com/outlier-detection-with-isolation-forest-3d190448d45

9. IEEE Computational Intelligence Society. (2019 July). IEEE-CIS Fraud Detection,. Retrieved October 2019 from https://www.kaggle.com/c/ieee-fraud-detection/data

10. Pranjal Khandelwal(2017), Which algorithm takes the crown. Link: https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-lightgbm-vs-xgboost/

11. Andrich van Wyk (May 2018) An Overview of LightGBM. Link: https://www.avanwyk.com/an-overview-of-lightgbm/