CPS 592 Z3 – Machine Learning with Cybersecurity

Credit Card Fraud Detection using KNN, Decision Tree, Random Forest and Logistic Regression methods

Report by

Sivakrishna Vase

(vases1@udayton.edu | 1017107940)

ABSTRACT

Our project Credit Card Fraud Detection Using KNN, Decision Tree, Random Forest and Logistic Regression methods basically includes the various classifications to check which model gives us the best results. Models were built on the imbalanced data and hyperparameters were tuned. Then SMOTE and ADASYN techniques were used to balance the data. Models was tried on both SMOTE and ADASYN data to see which one is producing better result.

ACKNOWLADGMENT

The satisfaction that accompanies that the successful completion of any task would be incomplete withoutthe mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

We are grateful to our professor Dr. Zhongmei Yao for the guidance, inspiration throughout the Machine Learning in Cybersecurity Course that were very helpful for us in the preparation of this project.

I would also like to thank our TA Fangshi Zhou who have helped in successful completion of the project and course.

INTRODUCTION

Description of the project

Credit cards are widely used due to the popularization of e-commerce and the development of mobile intelligent devices. Card-not-present transactions (i.e., online transaction without a physical card) is more popular, especially all credit card operations are performed by web payment gateways, e.g., PayPal and Alipay. Credit card has made an online transaction easier and more convenient. However, there is a growing trend of transaction frauds resulting in a great loss of money every year. It is estimated that losses are increased yearly at double digit rates by 2020. Since the physical card is not needed in the online transaction environment and the card's information is enough to complete a payment, it is easier to conduct a fraud than before. Transaction fraud has become a top barrier to the development of e-commerce and has a dramatic influence on the economy. Hence, fraud detection is essential and necessary. "Fraud detection is a set of activities that are taken to prevent money or property from being obtained through false pretenses."

Fraud can be committed in different ways and in many industries. Most detection methods combine a variety of fraud detection datasets to form a connected overview of both valid and non-valid payment data to decide. This decision must consider IP address, geolocation, device identification, "BIN" data, global latitude/longitude, historic transaction patterns, and the actual transaction information. In practice, this means that merchants and issuers deploy analytically based responses that use internal and external data to apply a set of business rules or analytical algorithms to detect fraud.

Credit Card Fraud Detection with Machine Learning is a process of data investigation by a Data Science team and the development of a model that will provide the best results in revealing and preventing fraudulent transactions. This is achieved through bringing together all meaningful features of card users' transactions, such as Date, User Zone, Product Category, Amount, Provider, Client's Behavioral Patterns, etc. The information is then run through a subtly trained model that finds patterns and rules so that it can classify whether a transaction is fraudulent or is legitimate. All big banks like Chase use fraud monitoring and detection systems.

Fraud detection is a process of monitoring the transaction behavior of a cardholder in order to detect whether an incoming transaction is done by the cardholder or others. Generally, there are two kinds of methods for fraud detection. misuse detection and anomaly detection. detection uses classification methods to determine whether an incoming transaction is fraud or not. Usually, such an approach must know about the existing types of fraud to make models by learning the various fraud patterns. Anomaly detection is to build the profile of normal transaction behavior of a cardholder based on his/her historical transaction data and decide a newly transaction as a potential fraud if it deviates from the normal transaction behavior. However, an anomaly detection method needs enough successive sample data to characterize the normal transaction behavior of a cardholder.

The most commonly techniques used fraud detection methods are Naïve Bayes (NB), Support Vector Machines (SVM), K-Nearest Neighbor algorithms (KNN). These techniques can be used alone or in collaboration using ensemble or meta-learning techniques to build classifiers. But amongst all existing method, ensemble learning methods are identified as popular and common method, not because of its quite straightforward implementation, but also due to its exceptional predictive performance on practical problems. In this paper we trained various data mining techniques used in credit card fraud detection and evaluate each methodology based on certain design criteria. After several trial and comparisons, we introduced the bagging classifier based on decision three, as the best classifier to construct the fraud detection model. The performance evaluation is performed on real life credit card transactions dataset to demonstrate the benefit of the bagging ensemble algorithm. Credit card fraud is usually caused either by card owner's negligence with his data or by breach in a website's security. Here are some examples:

- A consumer reveals his credit card number to unfamiliar individuals.
- A card is lost or stolen and someone else uses it.
- Mail is stolen from the intended recipient and used by criminals.
- Business employees copy cards or card numbers of its owner.
- Making a counterfeit credit card.

Credit Card Fraud Detection Systems:

- Off-the-shelf fraud risk scores pulled from third parties (e.g. LexisNexis or MicroBilt).
- Predictive machine learning models that learn from prior data and estimate the probability of a fraudulent credit card transaction.
- Business rules that set conditions that the transaction must pass to be approved (e.g. no OFAC alert, SSN matches, below deposit/withdrawal limit, etc.).

Among these fraud analytics techniques, predictive Machine Learning models belong to smart Internet security solutions.

Fraud Detection System Implementation Steps:

- Data Mining: Implies classifying, grouping, and segmenting of data to search millions of transactions to find patterns and detect fraud.
- Pattern Recognition: Implies detecting the classes, clusters, and patterns of suspicious behavior. Machine Learning here represents the choice of a model/set of models that best fit a certain business problem. For example, the neural networks approach helps automatically identify the characteristics most often found in fraudulent transactions; this method is most effective if you have a lot of transaction samples.

Once the Machine Learning-driven Fraud Protection module is integrated into the E-commerce platform, it starts tracking the transactions. Whenever a user requests a transaction, it is processed for some time. Depending on the level of predicted fraud probability, there are three

possible outcomes:

- If the probability is less than 10%, the transaction is allowed.
- If the probability is between 10% and 80%, an additional authentication factor (e.g. a one-time SMS code, a fingerprint, or a Secret Question) should be applied.
- If the probability is more than 80%, the transaction is frozen, so it should be processed manually.

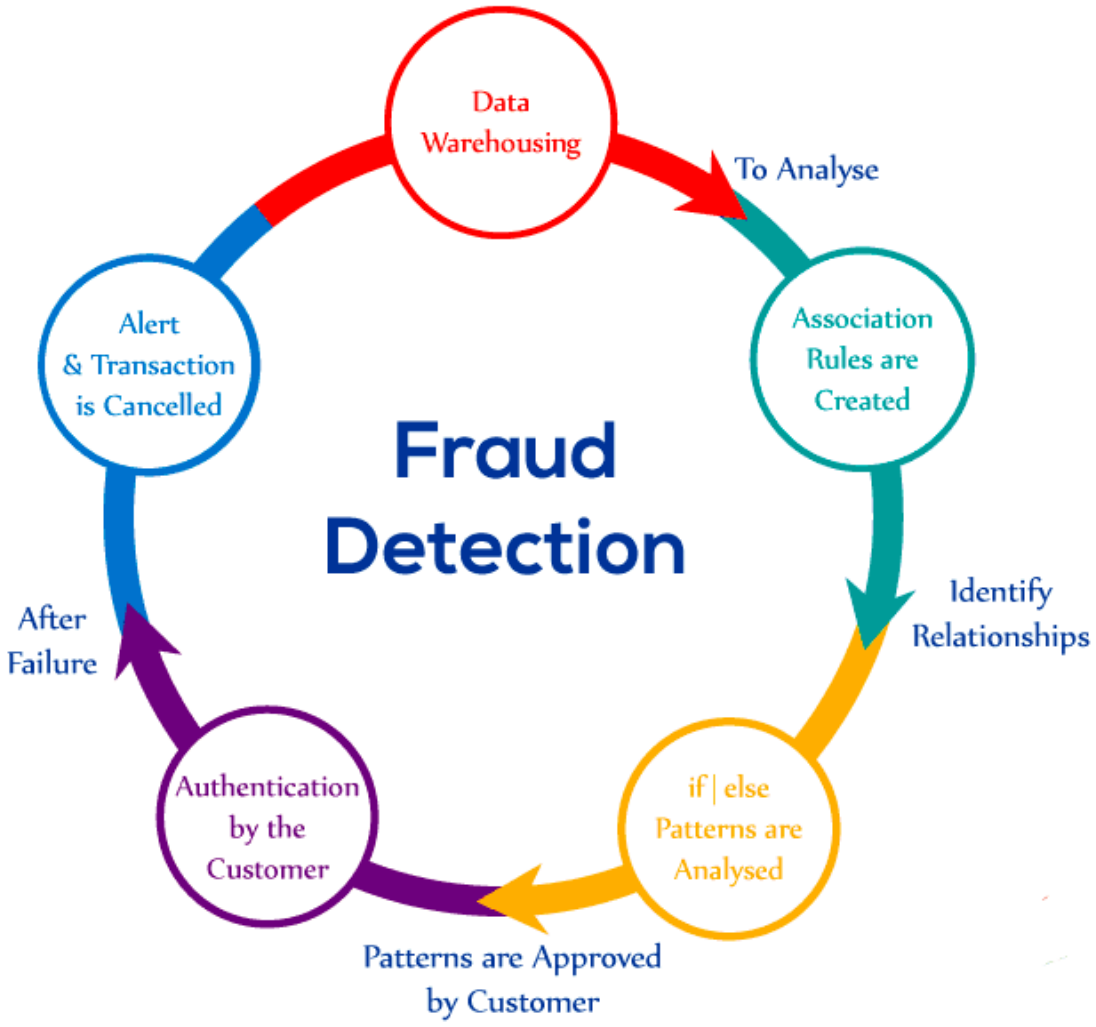


Image : The Flow of the Fraud Detection

RELATED WORKS

A comprehensive understanding of fraud detection technologies can be helpful for us to solve the problem of credit card fraud. The work in provides a comprehensive discussion on the challenges and problems of fraud detection research. Mohammad et.al., review the most popular types of credit card fraud and the existing nature-inspired detection methods that are used in detection methods. Basically, there are two types of credit card fraud: application fraud and behavior fraud. Application fraud is that criminals get new credit cards from issuing companies by forging false information or using other legitimate cardholders' information. Behavior fraud is that criminals steal the account and password of a card from the genuine cardholder and use them to spend

Advanced Credit Card Fraud Identification Methods and Their Advantages

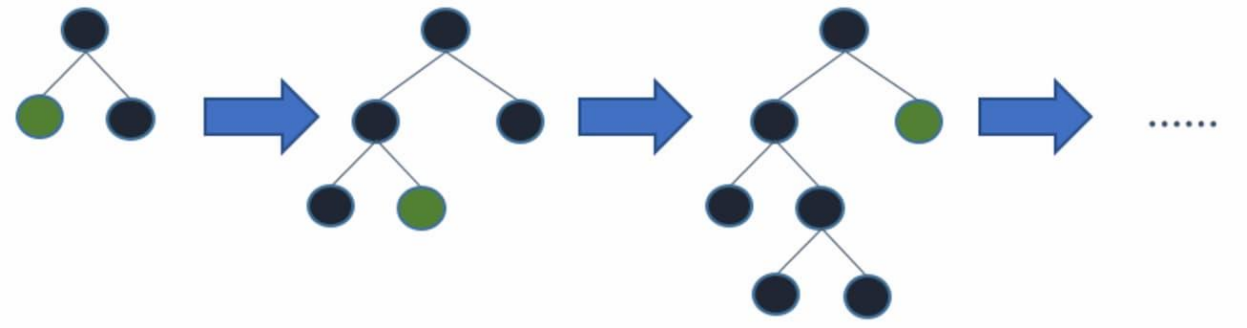
Advanced Credit Card Fraud Identification Methods are split into:

- Unsupervised: Such as PCA, LOF, One-class SVM, and Isolation Forest.
- Supervised: Such as Decision Trees (e.g. XGBoost and LightGBM), Random Forest, and KNN.

Unsupervised:
Unsupervised Machine Learning methods use unlabeled data to find patterns and dependencies in the credit card fraud detection dataset, making it possible to group data samples by similarities without manual labeling.
PCA (Principal Component Analysis) enables the execution of an exploratory data analysis to reveal the inner structure of the data and explain its variations. PCA is one of the most popular techniques for Anomaly Detection.
PCA searches for correlations among features — which in the case of credit card transactions, could be time, location, and amount of money spent — and determines which combination of values contributes to the variability in the outcomes. Such combined feature values allow the creation of a tighter feature space named principal components.
LOF (Local Outlier Factor) is the score factor that helps understand how high the chance is for a certain data sample to be an outlier (anomaly). This is another of the most popular Anomaly Detection methods.
To calculate LOF, the number of neighboring data points is considered to figure out its density and compare it to the density of other data points. If a certain data point has a substantially low density compared to its close neighbors, it is an outlier.
One-class SVM (Support Vector Machine) is a classification algorithm that helps to identify outliers in data. This algorithm allows one to deal with imbalanced data-related issues such as Fraud Detection.
The idea behind One-class SVM is to train only on a solid amount of legitimate transactions and then identify anomalies or novelties by comparing each new data point to them.
Isolation Forest (IF) is an Anomaly Detection method from the Decision Trees family. The main idea of IF, which differentiates it from other popular outlier detection algorithms, is that it

precisely detects anomalies instead of profiling the positive data points. Isolation Forest is built of Decision Trees where the separation of data points happens first because of randomly selecting a split value amidst the minimum and maximum value of the chosen feature.
Subsequently, if we have a set of legitimate transactions, the Isolation Forest algorithm will define fraudulent credit card transactions because of their values — which are often very different from the values positive transactions have (i.e. they take place further away from the normal data points in the feature space).

Supervised :
Supervised ML methods use labeled data samples, so the system will then predict these labels in future unseen before data. Among supervised ML fraud identification methods, we define Decision Trees, Random Forest, KNN, and Naive Bayes.
K-Nearest Neighbors is a Classification algorithm that counts similarities based on the distance in multi-dimensional space. The data point, therefore, will be assigned the class that the nearest neighbors have.
This method is not vulnerable to noise and missing data points, which means composing larger datasets in less time. Moreover, it is quite accurate and requires less work from a developer in order to tune the model.
XGBoost (Extreme Gradient Boosting) and **Light GBM (Gradient Boosting Machine)** are a single type of gradient-boosted Decision Trees algorithm, which was created for speed as well as maximizing the efficiency of computing time and memory resources. This algorithm is a blending technique where new models are added to fix the errors caused by existing models.
Light GBM differs from other tree-based techniques only in that it follows a leaf-wise direction to build conditions instead of a level-wise direction (fig.1,2). In general, the idea behind all tree-based gradient boosting based algorithms is the same.



Leaf-wise tree growth

To classify a transaction as a fraudulent charge, the result (probability) of many Decision Trees is summarized — whereas every future tree improves its results based on of the errors made by its predecessors.
Random Forest is a classification algorithm that is comprised of many Decision Trees. Each tree has nodes with conditions, which define the final decision based on the highest value.
The Random Forest algorithm for fraud detection and prevention has two cardinal factors that make it good at predicting things. The first one is randomness, meaning that the rows and

columns of data are chosen randomly from the dataset and fit into different Decision Trees. Say Tree Number 1 receives the first 1,000 rows, Tree Number 2 receives Rows 4,000 to 5,000, and the Tree Number 3 has Rows 8,000 to 9,000.
The second factor is diversity, meaning that there's a forest of trees that contribute to the final decision instead of just one decision tree. The biggest advantage here is that this diversity decreases the chance of model overfitting, while the bias remains the same.
Different ML models can be used to detect fraud; each of them has its pros and cons. Some models are very hard to interpret, explain, and debug, but they have good accuracy (e.g. Neural Networks, Boosting, Ensembles, etc.); others are simpler, so they can be easily interpreted and visualized as a bunch of rules (e.g. Decision Trees).
It is very important to train the Fraud Detection model continuously whenever new data arrives, so new fraud schemas/patterns can be learned, and fraudulent data detected as early as possible.

Fraud is a major problem for the whole credit card industry that grows bigger with the increasing popularity of electronic money transfers. To effectively prevent the criminal actions that lead to the leakage of bank account information leak, skimming, counterfeit credit cards, the theft of billions of dollars annually, and the loss of reputation and customer loyalty, credit card issuers should consider the implementation of advanced Credit Card Fraud Prevention and Fraud Detection methods. Machine Learning-based methods can continuously improve the accuracy of fraud prevention based on information about each cardholder's behavior.

APPROACH FROM REFERENCE PAPER

In the reference paper author used and implemented the "Random Forest" classifier. random forest as a classifier. The popularity of decision tree models in data mining is owed to their simplification in algorithm and flexibility in handling different data attribute types. However, single-tree model is possibly sensitive to specific training data and easy to overfit. Ensemble methods can solve these problems by combine a group of individual decisions in some way and are more accurate than single classifiers. Random forest, one of ensemble methods, is a combination of multiple tree predictors such that each tree depends on a random independent dataset and all trees in the forest are of the same distribution. The capacity of random forest not only depends on the strength of individual tree but also the correlation between different trees. The stronger the strength of single tree and the less the correlation of different tress, the better the performance of random forest. The variation of trees comes from their randomness which involves bootstrapped samples and randomly selects a subset of data attributes.

Although there possibly exist some mislabeled instances in our dataset, random forest is still robust to noise and outliers. We introduce two kinds of random forests, named as random forest I and random forest II, which are different in their base classifiers (i.e., a tree in random forest). For readability, some notations are introduced here. Considering a given dataset D with n examples (i.e. |D| = n), we denote: D = {(xi, yi)},i = 1, ..., n, where xi ∈ X is an instance in the m-dimensional feature space X = {f1, f2, ..., fm} and yi ∈ Y = {0, 1} is the class label associated with instance xi.

Random-tree-based random forest

A base classifier of random forest, which is a simple implement of decision tree, is called a random tree. The training set of each tree is a collection of bootstrapped samples selected randomly from the standard training set with replacement. At each internal node, it randomly selects a subset of attributes and computes the centers of different classes of the data in current node. The centers of class 0 and 1 are denoted as left Center and right Center, respectively. The kth element of a center is computed based on the following equations.

$$leftCenter[k] = \frac{1}{n} \sum_{i=1}^{n} x_{ik} I(y = 0)$$

$$rightCenter[k] = \frac{1}{n} \sum_{i=1}^{n} x_{ik} I(y = 1)$$

where I(y = 0) and I(y = 1) are the dictator functions. At the current node, each record of the dataset is allocated to the corresponding class according to the Manhattan distance between the record and the center.

Algorithm for Random Tree based:

Input: Dataset D and the number of trees N T .

Output: A random forest. For i = 1 to N T :

- Draw a bootstrap sample Di from the training set D whose size is n.

- Construct a binary tree of the bootstrapped data recur-sively from root node. Repeatedly perform the following steps until all records of current node belong to a class.

a) Randomly select a subset of $\sqrt{m}$ attributes.
b) For j = 1 to $\sqrt{m}$:
   i) Compute lef tCenter[j] and rightCenter[j].
c) For k = 1 to |Dic|:
   i) Compute the Manhattan distance dLk and dRk between the recordk and each center.
   ii) if dLk ≤ dRk
       Allocate recordk to the left child of the current node.
       else
       Allocate recordk to the right child of the current node.
d) Split the node into a left child and a right child.

where Dic is the subset of Di in the current node.

The internal nodes are represented by circles. The variables in a cir-cle are attributes randomly chosen from X = {x1, x2, x3, x4}. The decisions are made according to their values. Each termi-nal node is represented by a rectangle and corresponds to a class. The number in a terminal node represents which class the node belongs to.

CART-based random forest

The base classifier of random forest II is CART (Classifica-tion and Regression Trees) whose training set also comes from bootstrapped samples. At each node, it splits dataset by choosing the best attribute in a subset of attributes according to Gini impurity which measures uncertainty of dataset. The subset of attributes are randomly selected from all attributes of dataset. According to the advice from Breiman, the size of the subset is set to the square root of the number of all attributes

The Gini impurity is defined in (4) and is described in
(5) under the condition of feature xi.

$$Gini(Node) = 1 - \sum_{k=1}^{C} p_k^2 \qquad (4)$$

Where C is the number of classes which is 2 in binary classification problem and pk is the probability that a record belongs to class k.

$$Gini(Node, x_i) = \frac{|Node_l|}{|Node|} Gini(Node_l) + \frac{|Node_r|}{|Node|} Gini(Node_r) \qquad (5)$$

Where N odel is the left child of the current node and |N ode| represents the number of records in the dataset w.r.t. the current node. A following algorithm II describes the process of producing a type-II random forest:

Algorithm for cart based random forest:

Input: Dataset D, the number of trees N T and the threshold
T of Gini impurity

Output: A random forest For i = 1 to N T :

- Draw a bootstrap sample Di of size n from the training set D.
- Construct a decision tree of the bootstrapped data recur-sively from root node. Repeatedly perform the following steps until Gini impurity less than T .

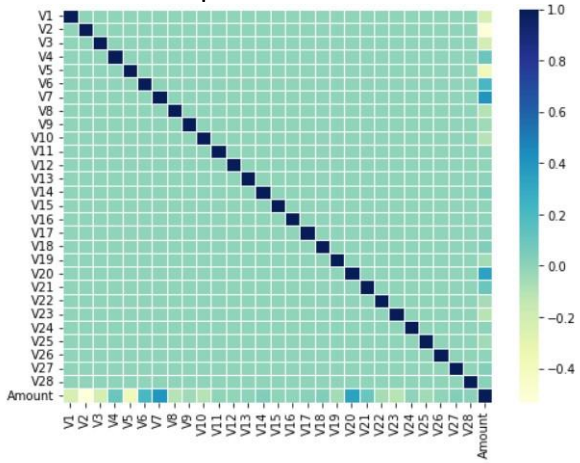
a) Randomly select a subset of $\sqrt{m}$ attributes.
b) For j = 1 to √m:
Compute Gini impurity for feature x_j .
c) Choose the feature and its value with the minimum Gini impurity as the split attribute and split value.
d) Split the internal node into two child nodes according to the split attribute and value.

The internal nodes are represented by circles. The variable and number in circles are the best splitting attribute and its value, respectively. The number labeled on the left edge from internal node means the value of this attribute greater than or equal to the splitting value, while the number labeled on the right edge means that the attribute has a less value. Each terminal node is represented by a rectangle. The number in a terminal node is the class the node belongs to.

MY APPROACH ON PROJECT IMPLEMENTATION

In this project Credit Card Fraud Detection Using KNN, Decision Tree, Random Forest and Logistic Regression methods basically includes the various classifications to check which model gives us the best results. Models were built on the imbalanced data and hyperparameters were tuned. Then SMOTE and ADASYN techniques were used to balance the data. Models was tried on both SMOTE and ADASYN data to see which one is producing better result.
We have downloaded the dataset from the Kaggle.com, which provides very fine data for implementation of this project.
Implementation Steps of the project code:

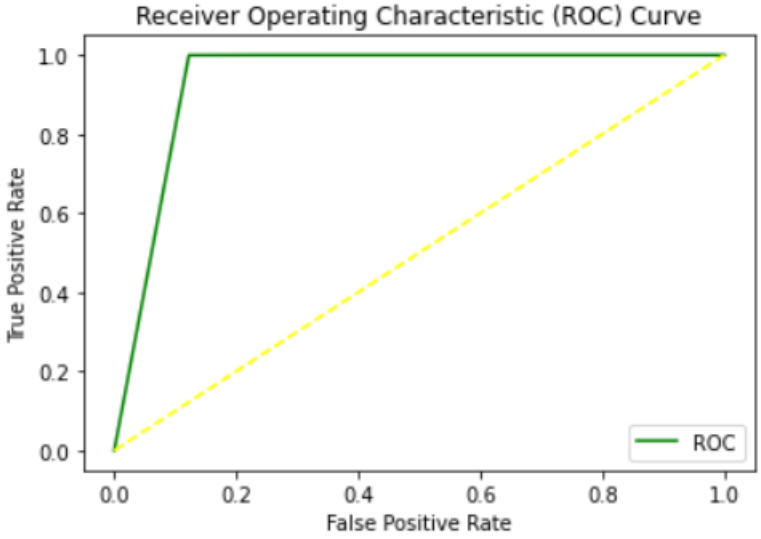
- Defining the libraries for the project
- Exploratory data Analysis: Here we provide the dataset path to the code. Checking the data for the project and verifying and dropping unnecessary columns and we have plotted the Heatmap to check correlation



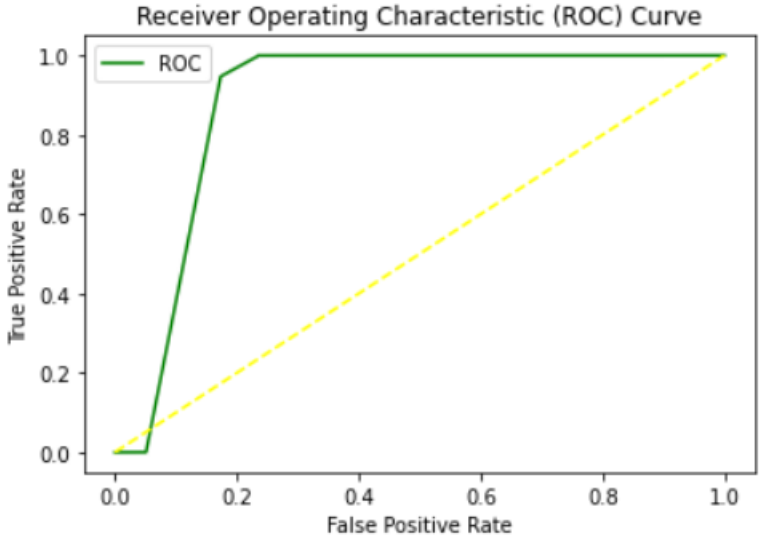
- V7 and V20 seem to have positive correlation with the feature 'Amount'. Since this is a PCA converted data, there isn't much to conclude from the heatmap.

- Next, we have spilt the data into Train and Test data.
- We have plotted the distribution of Variables.
- Model performance parameter: Here we have used the ROC curve and finding the AUC score as the performance matrix for the models. ROC curve measures the performance of the model at different thresholds which will help us find the optimum threshold for the mode.
- Model Building on Imbalanced Data: we have used very few parameters to tune the data. Here for cross validation, I have used GridSearchCV and Stratified Kfold (cross_val_score)
- Logistic Regression Model: For logistic regression model we have used the “LogisticRegression” classifier. The AUC score is 0.98 but the data is clearly overfitting due to the imbalanced data.
- K-nearest Neighbor Model : Here we import the “KNeighborsClassifier”. The KNN model with imbalanced data gives AUC of 0.94 which is pretty good but recall is 0.77 which is

the score we should look to improve in this case.



- Decision tree Model: Here we have used the “DecisionTreeClassifier”classifier. The AUC score for decision tree is only 0.88 which is not satisfactory. The precision and recall are also lower than KNN and logistic regression model.



- Random Forest Classifier: In Rancom forest classifier we have used the “RandomForestClasifier” classifier. We are getting very good precision (0.97) for Fraudulent class which is very good along with the AUC of 0.97

- Choosing the Best Model: Here we have summarized the all the methods.
  - To save banks from high-value fraudulent transactions, we have to focus on a high recall in order to detect actual fraudulent transactions, but we cannot have a very low precision.

  - The top two models giving better AUC are KNN (with SMOTE) & Random Forest (with SMOTE).

Scores of Random Forest model:

- AUC : 0.98

- Recall: 0.88

- Precision: 0.42

- f1-Score: 0.57
- Scores of KNN model:

- AUC: 0.94

- Recall: 0.88

- Precision: 0.61

- f1-Score: 0.72
- Comparing both we can see that the Random Forest model has more AUC score than KNN, but the KNN model has a better f1-score (Which is a result of better precision and recall)

- Though the recall is same in both, having a better precision at a little trade off with AUC score will help the model generalize better. Having a good precision will help preventing a fair transaction being called fraudulent.

- So, the KNN model with SMOTE oversampling is our final model.

CONCLUSION

This project has implemented the performance of various kinds of credit card fraud detection models. A real-life dataset on credit card transactions is used in our project. Although random forest obtains good results on small set data, there are still some problems such as imbalanced data. Comparing both we can see that the Random Forest model has more AUC score than KNN, but the KNN model has a better f1-score (Which is a result of better precision and recall).

REFERENCES

Reference Research Papers:

- Random forest for credit card fraud detection by Shiyang Xuan; Guanjun Liu; Zhenchuan Li; Lutao Zheng; Shuo Wang; Changjun Jiang

  https://ieeexplore.ieee.org/document/8361343

- Fraud Detection in Credit Card Transactions Using SVM and Random Forest Algorithms by S K Saddam Hussain;E Sai Charan Reddy;K Gangadhar Akshay;T Akanksha

  https://ieeexplore-ieee-org.libproxy.udayton.edu/document/9640631

- Credit Card Fraud Detection Using Machine Learning by D. Tanouz;R Raja Subramanian;D. Eswar;G V Parameswara Reddy;A. Ranjith Kumar;CH V N M Praneeth

  https://ieeexplore-ieee-org.libproxy.udayton.edu/document/9432308

- Credit Card Fraud Detection Using Machine Learning by Ruttala Sailusha;V. Gnaneswar;R. Ramesh;G. Ramakoteswara Rao

  https://ieeexplore-ieee-org.libproxy.udayton.edu/document/9121114

- Credit Card Fraud Detection Using Lightgbm Model by Dingling Ge;Jianyang Gu;Shunyu Chang;JingHui Cai

  https://ieeexplore-ieee-org.libproxy.udayton.edu/document/9134072

Dataset For the Project:

- https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

References for Author Proposed Code:

- https://www.kaggle.com/code/hassanamin/credit-card-fraud-detection-using-random-forest/notebook

- https://github.com/Nirjoy/CSE445-Credit-Card-Fraud-Detection-using-Random-Forest-SMOTE

- https://github.com/julian0316/Credit-Card-Fraud-Detection-with-Random-Forest

- https://github.com/Prajwal10031999/Credit-Card-Fraud-Detection-using-Random-Forest