REPORT

CREDIT CARD FRAUD DETECTION

By: Swati Jaiswal

Reg_no= 11905517

Submitted to: Miss Ankita Wadhawan Ma’am

GitHub link:
https://github.com/SwatiJaiswal8/Credit_card_fraud_detection

Dataset link: https://www.kaggle.com/mlg-ulb/creditcardfraud

INTRODUCTION

Fraud' in credit card transactions is unauthorized and unwanted usage of an account by someone other than the owner of that account. Necessary prevention measures can be taken to stop this abuse and the behaviour of such fraudulent practices can be studied to minimize it and protect against similar occurrences in the future. In other words, Credit Card Fraud can be defined as a case where a person uses someone card issuing authorities are unaware of the fact that the card is being used.
Fraud detection involves monitoring the activities of populations of users to estimate, perceive or avoid objectionable behaviour, which consist of fraud, intrusion, and defaulting.
This is a very relevant problem that demands the attention of communities such as machine learning and data science where the solution to this problem can be automated.
This problem is particularly challenging from the perspective of learning, as it is characterized by various factors such as class imbalance. The number of valid transactions far outnumber fraudulent ones. Also, the transaction patterns often change their statistical properties over the course of time.
These are not the only challenges in the implementation of a real-world fraud detection system, however. In real world examples, the massive stream of payment requests is quickly scanned by automatic tools that determine which transactions to authorize.
Machine learning algorithms are employed to analyse all the authorized transactions and report the suspicious ones. These reports are investigated by professionals who contact the cardholders to confirm if the transaction was genuine or fraudulent.
The investigators provide feedback to the automated system which is used to train and update the algorithm to eventually improve the fraud-detection performance over time.

Challenge with credit card fraud detection:

Credit card fraud detection is an extremely difficult, but an important problem to solve. However, there are several constraints associated with the problem. In what follows, we mention a few important points:

- Unavailability of Datasets: One major challenge associated with the problem is the lack of availability of public datasets [12, 13, 18]. Credit card companies maintain the datasets of their transactions, however, due to privacy and security concerns they are not able to release this data in the public domain. But any research work in this direction will need such data to build a model. However, there are some results which are performed on synthetically generated data [2, 6]. But none of these previous results disclose the features of the data and the parameters used in classifier models. Due to this, benchmarking different fraud detection systems is quite difficult. Also, the limited amount of data which is publicly available may not be enough to detect a pattern, as there are millions of possible places and e-commerce sites to use a credit card, which makes this problem extremely hard.
- Dynamic Fraudulent Behaviour: Fraudsters often change their behaviour over time to beat the current detection systems by modifying their pattern. Due to this, pattern of normal and fraudulent transactions changes constantly. It is quite possible that there exist past fraudulent transactions, which now fit the pattern of normal (legitimate) transactions. Consequently, the problem becomes very complex in nature, and it is difficult to predict even by human experts.

- Highly Skewed Dataset: Credit card fraud datasets are highly skewed – where a vast majority of the samples are normal transactions while only a small minority of them are fraudulent transactions. In most of the cases more than 99% of the total transactions are normal, and consequently less than 1% of them are fraudulent.
- Right Evaluation Parameters: Accuracy is one of the standard measures for determining the effectiveness of any classifier. However, in credit card fraud detection, accuracy may not be the correct measure because, due to skewed nature of the datasets, it is possible that even in a model with high accuracy, most of the fraudulent transactions are being mis classified. Therefore, it is important to evaluate such models on recall - correctly classifying fraudulent transaction - and precision - correctly classifying normal transactions.

Literature Review:

Rimpal R. Popat with Jayesh Chaudhary: They made a survey on credit card fraud detection, considering the major areas of credit card fraud detection that are bank fraud, corporate fraud, Insurance fraud. With these they have focused on the two ways of credit card transactions i) Virtually (card, not present) ii) With Card or physically present. They had focused on the techniques which are Regression, classification, Logistic regression, Support vector machine, Neural network, Artificial Immune system, K-nearest Neighbor, Naïve Bayes, Genetic Algorithm, Data mining, Decision Tree, Fuzzy logic-based system, etc. In which, they have explained six data mining approaches as theoretical background that are classification, clustering, prediction, outlier detection, Regression, and visualization. Then have explained about existing techniques based on statistical and computation which is Artificial Immune system (AIS), Bayesian Belief Network, Neural Network, Logistic Regression, Support Vector Machine, Tree, Selforganizing map,Hybrid Methods, As a result, they had concluded that all the present machine learning techniques mentioned above can provide high accuracy for the detection rate and industries are looking forward to finding new methods to increase their profit and reduce the cost. Machine learning can be a good choice for it. [A Survey on Credit Card Fraud Detection using Machine Learning].

Mohamad Zamini: Purposed an unsupervised fraud detection method using autoencoder based clustering.The autoencoder is anauto associatorneural network they have used it to lower the dimensionality, extract the useful features, and increase the efficiency of learning in a neural network. They had used European dataset with 284807 transactions in which 0.17% is the fraud and trained there autoencoder based clustering with the following parameters Number of iterations = 300 Number of clusters = 2 Clustering initialization = k-means++ Divergence tolerance = 0.001 Learning rate of the model = 0.1 Number of epochs = 200 Activation function = elu, Relu. As a result, they got their training loss as 0.024 and validation loss as 0.027 and the mean of not fraud data 75% less than the mean of reconstructive error that is 25% the design of there is model is context-free. In concern about the model predictions, the True positive are 56,257, Falsenegative is 607, False positive are 18, True negativesis 80 and the best preferred are (56,257 + 80 = 56,337). The right predictions made are 56,337 out of 284807. [Credit Card Fraud Detection using autoencoder based clustering].

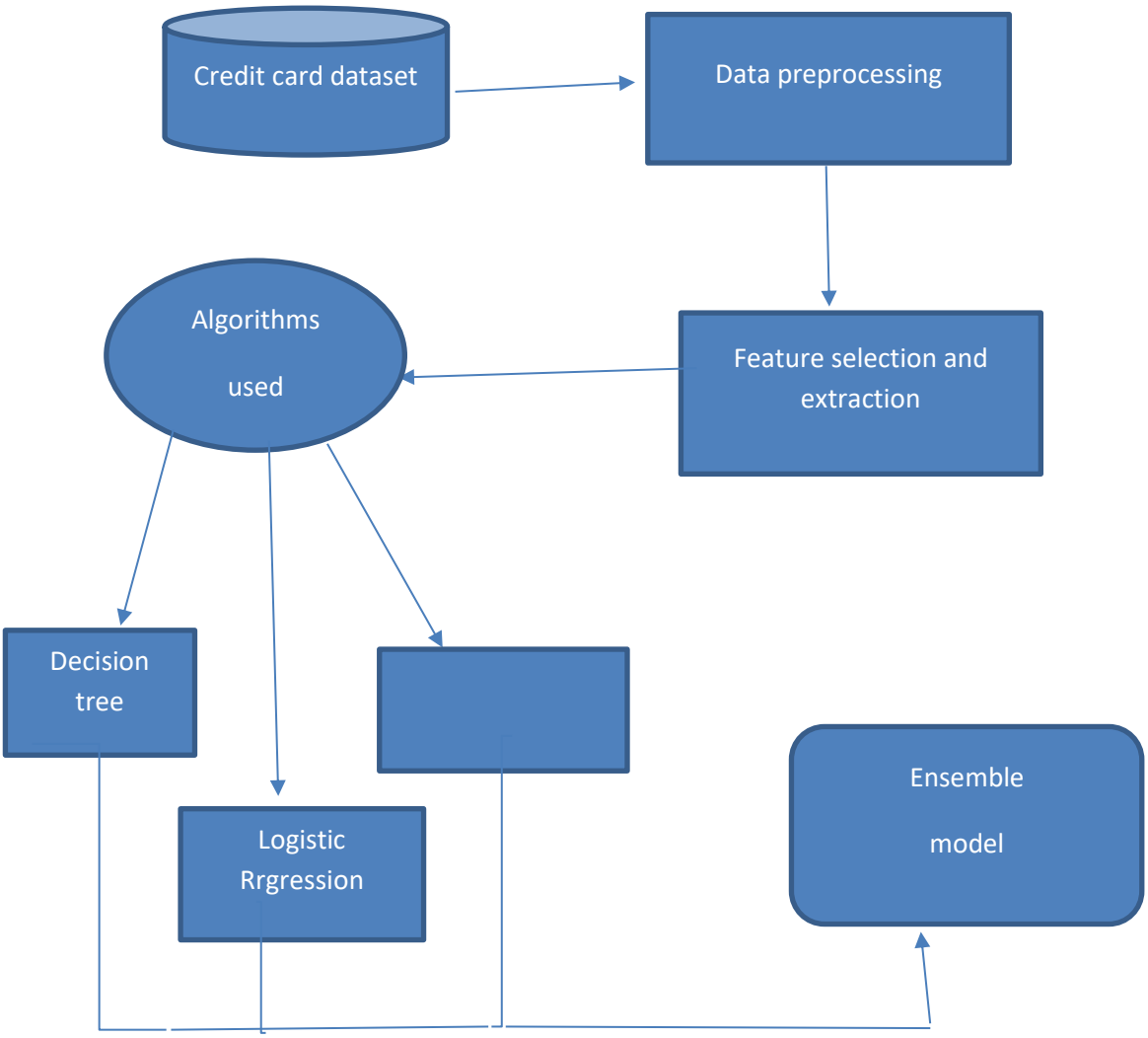
Changjun Jiang: Proposed a novel fraud detection method that has four stages they first utilize the historical transaction data to divide them into groups to form clusters of transactions having the same behavior then thus they came up with a sliding windowstrategy to aggregate transactions.This algorithm is used to characterize the behavioral pattern of a

cardholder then after aggregation, we use the new window formed the feature extraction is done. At last, the classification takes place and classifies behavioral patterns and assignments. As a result, their method ofLogistic Regression with raw data (RawLR), Random Forest with aggregation data (AggRF), and Random Forest and feedback technique with aggregation data (AggRF +FB) are the best method with 80% accuracy as compared to other methods.[Credit Card Fraud Detection: A Novel Approach Using Aggregation Strategy and Feedback Mechanism].

Sahil Dhankhad: They applied supervised machine learning algorithms on the real-world data set and then used those algorithms to implement a super classifier using ensemble learning and then they compared the performance of supervised algorithms with their implementation of a super classifier. They used ten machine learning algorithms such as Random Forest, Stacking Classifier, XGB Classifier, Gradient Boosting, Logistic Regression, MLP Classifier, SVM, Decision Tree, KNN, Naïve Bayes. And compared the accuracy, Recall Precision, confusion matrix with the result of their super classifier. As a result, they found that the Logistic Regression is better for predicting fraud transactions. [Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study].

Kuldeep Randhawa: They used twelve machine learning algorithms for credit card fraud detection in which their range standard from a neural network to deep learning. They are tracing the performance of benchmark and real-world datasets. In addition, the AdaBoost and majority voting methods are applied for forming the hybrid models. As they're related study explains about single and hybrid models. For both the parameters (Benchmark and real-world datasets) they had given the results using there twelve selected algorithms that are Naïve Bayes, Random Forest, Decision Tree, Gradient Boosted Tree, Decision Stump, Random Tree, Neural Network, Linear Regression, Deep Learning, Logistic Regression, SVM, Multilayer Perceptron. As a result, when standard algorithms used with AdaBoost and majority voting methods under benchmark data the best accuracy and sensitivity acquired by Random Forest algorithm 95% and 91% respectively. When experimented with real-world data the accuracy rate is still above 90% even with 30% noise in the dataset. MCC (Mathew's correlation coefficient) is standard to measure the performance of a model so in case of majority voting the best MCC score is 0.823 whereas 0.942 with 30% of noise added to the dataset. [Credit Card Fraud Detection Using AdaBoost and Majority Voting]

Our Approach:



1-Dataset Analysis:

There are a total of 284,807 transactions with only 492 of them being fraud. Let's import the necessary modules, load our dataset. 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.

2-Data Preprocessing:

The task of data preprocessing is to organize the original business data with the new "business model", clear those attributes irrelevant to the aim of data mining, supply clean, accurate, simplified data to improve the quality and efficiency of excavation under the guidance of domain knowledge [7]. The data preprocessing mainly concludes data cleaning, integration, transformation, and reduction. In this way, the dirty, incomplete, and inconsistent data in real world can be corrected. Data cleaning: By filling the null, smoothing noise data, identify and delete the isolated data, and solve inconsistency to attain the goal of clearing data. Data integration: Save the data belonging to several data sources to a consistent data storage (such as data warehouse), these data sources may include several databases, data cubes or ordinary files. Data conversion: Convert the data into one form that is suitable for excavation, for instance, zoom the attributive data in proportion, make it falls into a comparatively smaller specified zone. Data reduction: The compressed data used to acquire the dataset is much smaller than the original data, but it keeps its integrity. Thus, the data mining will have more effect on the condensed dataset and produce the same (or almost same) analysis result. Data preprocessing is indispensable to data mining. Statistics suggests that data preprocessing takes up 60 percent of the time in a complete process of data mining.

3-Data Preparation: Before continuing with our analysis, it is important not to forget that while the anonymized features have been scaled and seem to be centred around zero, our time and amount features have not. Not scaling them as well would result in certain machine learning algorithms that give weights to features (logistic regression) or rely on a distance measure (KNN) performing much worse. To avoid this issue, I standardized both the time and amount column. Luckily, there are no missing values and we, therefore, do not need to worry about missing value imputation.

And credit card system database is application-oriented rather than subject-oriented, the first step is to select the data relevant to the subject. In other words, all the data table items related to the goal of knowledge discovery should be extracted from the original database [8].

4-Data extraction: There are 100 tables of data sheet in the credit card database, such as customer application data, account information, business data, credit card transaction data flow, stop-payment list data, high-quality customer data. After analysis, we selected six tables which are related to the theme from these tables as follows: personal information table, consumer client table, card information table, transaction log table, overdraft history table, balance of history table. Personal information table stores basic information of all customers which includes 13 data items. There are four items related to the target theme, customer_id, passport _number, type of document' credit_rank. Consumer client table includes twenty fields: customer_id, birthday, gender_code, married, family_population, occupation_code, position, professional_title_code, unit, unit_kind, salary, extra_income, credit_card_id, education_code, customer_character_code, address, telephone, zip, cell_phone, email. According to the analysis of these fields, there are eight fields which are regarded as relevant fields to the target theme, including customer_id, birthday, gender_code, married, occupation_code, unit, unit_kind, salary, education_code. Otherwise, the other fields either miss data seriously or have nothing to with the expecting result, such as unit field.

Card information table has the credit card information of all customers. Only six fields are considered as the associated fields with the analysis theme in this table, such as card_id, account_number, account_balance, card_type, credit_line and issuing_date. Card trading log table records the history of all trading information for each credit card account. This table has thirteen fields named terminal_number, card_id, card_type, trading_money, liquidation_amount, terminal_serial_number, system_serial_number, trading_date, expiration_date, liquidation_date, business_number, trade_company. The valid fields are card_id, card_type, trading_date, trading_money, business_number, trade_company. There are four fields, account_id, accounting_date, account_balance, interest-bearing_balances in the balance of history table which is to record the balance changes of each credit card account. Overdraft history table that saves the overdraft information of each credit card account includes four fields such as account_balance, overdraft_date, overdraft_day.

5-Classifications Algorithms Onto the part you've probably been waiting for all this time: training machine learning algorithms. To be able to test the performance of our algorithms, I first performed an 80/20 train-test split, splitting our balanced data set into two pieces. To avoid overfitting, I used the very common resampling technique of kfold cross-validation. This simply means that you separate your training data into k parts (folds) and then fit your model on k-1 folds before making predictions for the kth hold-out fold. You then repeat this process for every single fold and average the resulting predictions.

To get a better feeling of which algorithm would perform best on our data, let's quickly spot-check some of the most popular classification algorithms:

- Logistic Regression
- Support Vector Classifier
- Random Forest Classifier
- Classification Trees

Logistic Regression: It is commonly used to estimate the probabilities than on instance belong to a particular class. For example, *it tells what is the probability that the transaction is fraud?* If the probability is more than 50%, then the model predicts that the instance belongs to that class otherwise does not belong to that class. It is basically a *binary classifier*.

How are the probabilities calculated?

Just like Linear Regression, the Logistic Regression model computes a weighted sum of input features and bias, but instead of outputting the result, it passes through a logistic function.

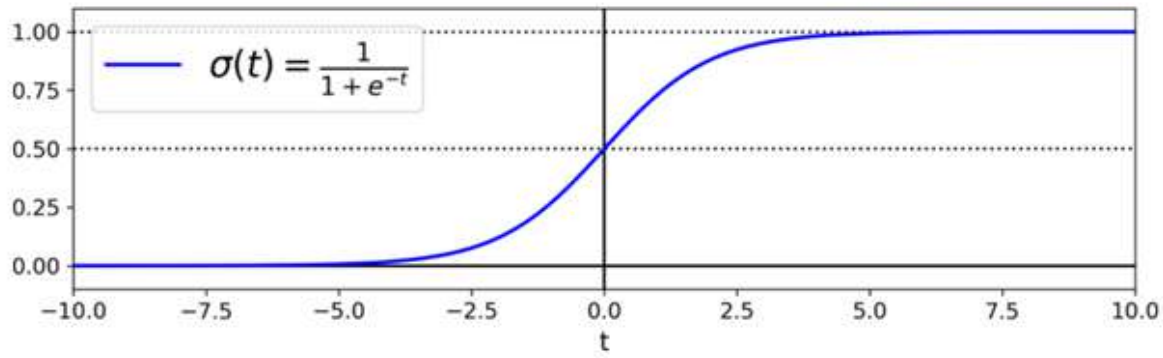
$$\hat{p} = h_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma(\boldsymbol{\theta}^{\mathsf{T}}\mathbf{x})$$

hθ = Hypothesis Function using model parameter θ

x = Instances feature vector

σ () = Sigmoid function (i.e. S-shaped) that output numbers between 0 and 1.

$$\sigma(t) = \frac{1}{1 + \exp(-t)}$$



$$\hat{y} = \begin{cases} 0 & \text{if } \hat{p} < 0.5 \\ 1 & \text{if } \hat{p} \geq 0.5 \end{cases}$$

By observing the graph describe the Sigmoid function. we notice that σ (t) < 0.5 when t < 0 and σ (t) ≥0.5 when t ≥0. This means if the model has estimated the probability that an instance **x** belongs to a positive class then it can easily make a prediction on ŷ that it will fall on the positive class.

Loss Function and Cost Function:

Loss Function helps in measuring how good our output P^ is when true labels are y. Usually, a function defined on a data point, i.e., prediction and label, and measures the penalty.

L(yˆ, y)= −[ylog(Pˆ)+(1−y)log(1−Pˆ)]

$$c(\boldsymbol{\theta}) = \begin{cases} -\log(\hat{p}) & \text{if } y = 1 \\ -\log(1 - \hat{p}) & \text{if } y = 0 \end{cases}$$

Let me put this in simple terms:

If y= 1: then we want to -log Pˆ to be small, which means Pˆ needs to be large.

If y =0: then we want -log(1- Pˆ) to be large, which means Pˆ needs to be as small as possible.

Cost Function helps in determining how well the model is performing in the entire dataset. It is more general and is a sum of loss functions over your training set. The logistic Regression cost function is known as **Log loss**

$$J(\boldsymbol{\theta}) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} log\left(\hat{p}^{(i)}\right) + \left(1 - y^{(i)}\right) log\left(1 - \hat{p}^{(i)}\right) \right]$$

Random forest:

Random Forest could be a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning could be a type of learning where different types of algorithms or same algorithm with multiple times to form a more powerful prediction model. The Random Forest combines multiple algorithms of the same type i.e. multiple decision trees, leading to a forest of trees, thus the name "Random Forest". The Random Forest can be used for regression and classification tasks.

The following are the essential steps concerned in performing the Random Forest Algorithm:

- Pick N random records from the data set.
- Build a decision tree based on these N records.
- Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
- For classification problem, each tree in the forest predicts the category to which the new record belongs.

Finally, the new record is assigned to the class and that wins the huge vote.

Decision Tree:

The decision tree is the simplest and most popular classification algorithm. For building the model the decision tree algorithm considers all the provided features of the data and comes up with the **important** features.

Because of this advantage, the decision tree algorithms also used in identifying the importance of the feature metrics. Which used in **handpicking** the features.

Once the important features identified then the model trains with the training data to come up with a **set of rules**. These rules used in predicting future cases or for the test dataset.

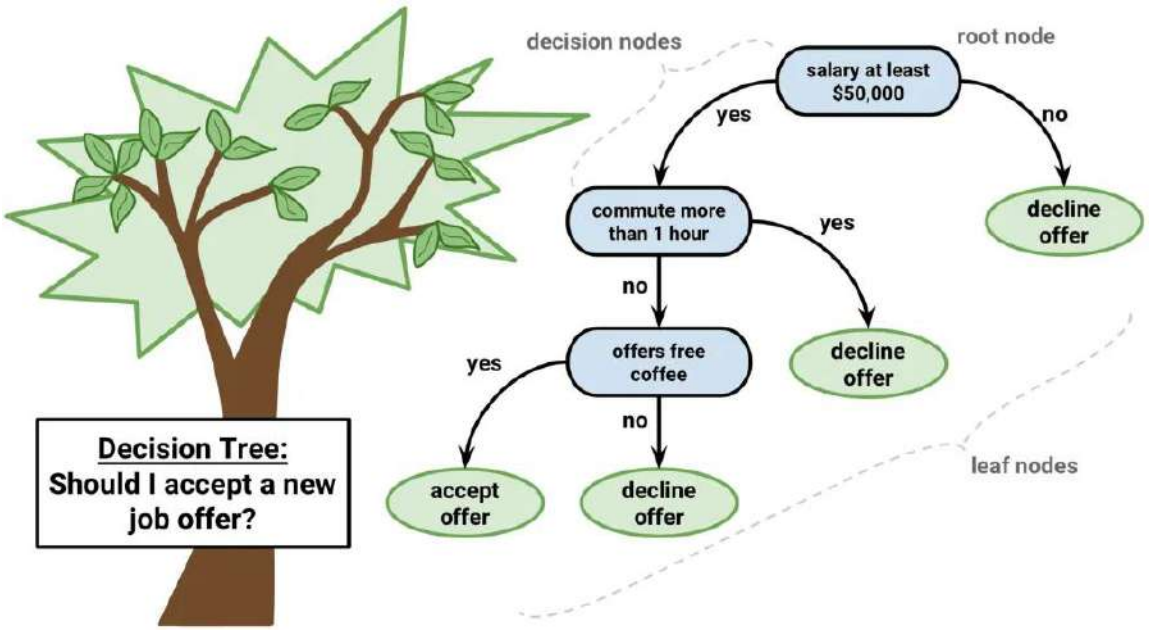
Decision Tree algorithm belongs to the family of supervised learning algorithm . Unlike other supervised learning algorithms, decision tree algorithm can be used for solving **regression and classification problems** too.

The general motive of using Decision Tree is to create a training model which can use to predict class or value of target variables by **learning decision rules** inferred from prior data(training data).

Decision Tree Algorithm Pseudocode

1. Place the best attribute of the dataset at the **root** of the tree.
2. Split the training set into **subsets**. Subsets should be made in such a way that each subset contains data with the same value for an attribute.

3. Repeat step 1 and step 2 on each subset until you find **leaf nodes** in all the branches of the tree.



Support Vector Classifier:

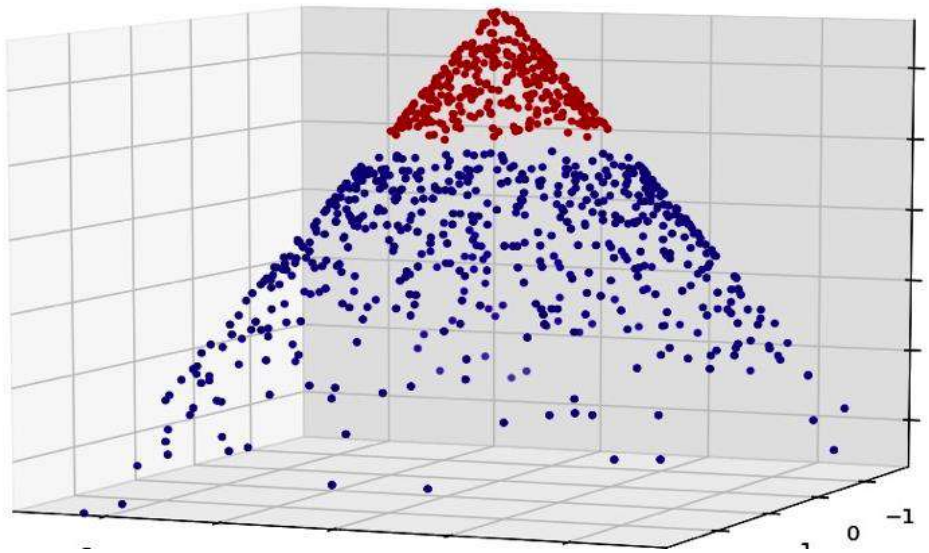
(SVMs) are a popular machine learning method for classification, regression, & other learning tasks. LIBSVM is a library for Support Vector Machines (SVMs). A typical use of LIBSVM involves two steps: first, training a data set to obtain a model & second, using the model to predict information of a testing data set. For SVC & SVR, LIBSVM can also output probability estimates. Many extensions of LIBSVM are available at libsvmtools. A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.

1. Set up the training data for model creation

2. Set up SVM's parameters

3. SVM Trainer

4. SVM Predictor



Experiments Setting:

Our evaluation method consists of three parts: data processing, model training, and ensemble learning. In the data processing stage, we remove the stop words, remove punctuation, stemming. In the model training stage, the text is converted into a tf-idf vector using TfidVectorizer in scikit-learn [11], which is used as the training feature of logistic regression, support vector machine and other models. Get the best performance from the model by adjusting the hyperparameters. Finally, using the ensemble learning method, which can further improve the prediction accuracy. During the data preprocessing stage, we removed the extra spaces in the text, used the stop vocabulary provided by Stanford1 to remove the stop words, used the NLTK toolkit2 for stemming operations, and removed the punctuation of the sentence.

In the model training stage, we select logistic regression as the meta-classifier to learn a second-level classifier. Before using ensemble learning, we need to set the hyperparameter of each classifier. We use train data and validation data to training each independent classifier, adjust the hyperparameters to achieve the best performance of the independent classifier on the validation set. We use scikit-learn3 to perform feature extraction and model training. Use the TfidfVectorizer tool provided by scikit-learn to convert the text data into TF-IDF feature vector, using the logistic regression, support vector machine, naive Bayes, K-nearest neighbor, decision trees, randomForest models provided by the scikit-learn toolkit for training. Ensemble learning uses the brew toolkit4 for model fusion. Brew uses the output of each classifier as a new feature value, uses a logistic regression model to learn the weights of each classifier, then outputs the classification results.

Experimental Results:

Model	**Accuracy**
Logistic Regression	92
Random Forest	92
Support Vector Machine	91.9
Decision Tree	91.7
Ensemble model	91.8

Conclusion:

Fraud detection is a complex issue that requires a substantial amount of planning before throwing machine learning algorithms at it. Nonetheless, it is also an application of data science and machine learning for the good, which makes sure that the customer's money is safe and not easily tampered with. Future work will include a comprehensive tuning of the Random Forest algorithm I talked about earlier. Having a data set with non-anonymized features would make this particularly interesting as outputting the feature importance would enable one to see what specific factors are most important for detecting fraudulent transactions. As always, if you have any questions or found mistakes, please do not hesitate to reach out to me. A link to the notebook with my code is provided at the beginning of this article.

References:

1- https://data-flair.training/blogs/credit-card-fraud-detection-python-machine-learning/
2- https://www.ijsdr.org/papers/IJSDR1605113.pdf
3- http://www.iaeng.org/publication/IMECS2011/IMECS2011_pp442-447.pdf
4- http://www.iaeng.org/publication/IMECS2011/IMECS2011_pp442-447.pdf
5- https://www.researchgate.net/publication/343223270_Fraud_Detection_for_Credit_Card_Transactions_Using_Random_Forest_Algorithm
6- https://medium.com/analytics-vidhya/credit-card-fraud-detection-logistic-regression-121d2dd35e2d
7- https://www.ripublication.com/ijaer18/ijaerv13n24_18.pdf
8- https://www.ijitee.org/wp-content/uploads/papers/v10i6/C84000110321.pdf