IIIT Hyderabad
Statistical Methods in AI (Monsoon 2022)

FRAUD DETECTION USING MACHINE LEARNING

Dhulipati Lakshmi Girija
Mandyam Brunda
Mahesh sathvika
Yeduru Rupasree

1 Introduction

An effective fraud detection system should be able to detect fraudulent transactions with high accuracy and efficiency.It is necessary to prevent bad actors from executing fraudulent transactions, it is also very critical to ensure genuine users are not prevented from accessing the payments system. A large number of false positives may translate into bad customer experience and may lead customers to take their business elsewhere.The existence of severely unbalanced data sets is a significant obstacle in using ML in fraud detection.In the vast majority of datasets that are currently available, only a very small percentage of transactions are fraudulent. Designing an accurate and effective fraud detection system that effectively detects fraudulent activity while producing few false positives is a challenging task.

2 Dataset

This experiment used data that was obtained via Kaggle.The dataset was created using transactional logs from a mobile money service provider with operations in over 14 different countries, all of which are located in Africa. The dataset consists of nine attributes and a target with descriptions as follows:

- type—CASH-IN, CASH-OUT, DEBIT, PAYMENT, and TRANSFER, which are the main types of transactions under MMT.
- amount of the transaction in local currency.
- nameOrig—customer who started the transaction
- oldbalanceOrig—initial balance before the transaction
- newbalanceOrig—new balance after the transaction
- nameDest—customer who is the recipient of the transaction
- oldbalanceDest—initial balance recipient before the transaction
- newbalanceDest—new balance recipient after the transaction.
- isFraud—This is the transaction made by the fraudulent.

3 Data Analysis:

As more fraudulent transactions are observed in TRANSFER and CASH OUT transactions,we performed an analysis on them. When PCA was performed on numerical features and top two principal cohttps://www.overleaf.com/project/638a1ff63f80436fa7678093mponents were plotted,it

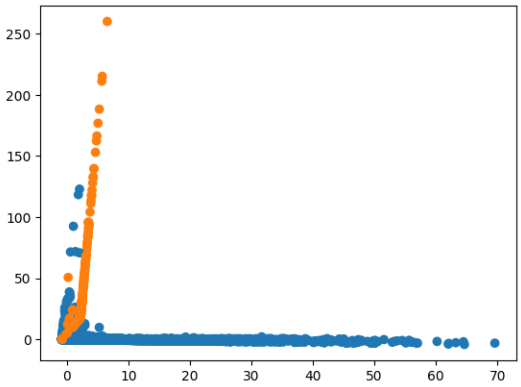


Figure 1: top two features on transfer after PCA

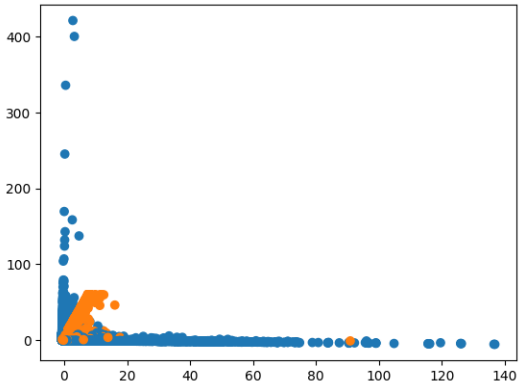


Figure 2: top two features on cash out after PCA

is observed the two classes fraud and non-fraud can be linearly separable for TRANSFER type of transactions. Above graphs show the top two features of the data set(TRANSFER and CASH OUT) after performing PCA.

4 Data splitting

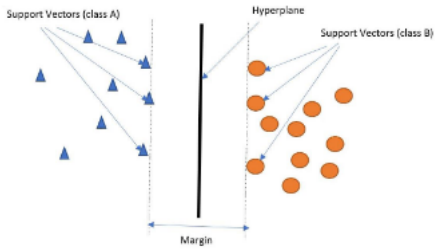
We divide our data set based on different transaction types described in the data set section. In particular, we use TRANSFER and CASH OUT transactions for our experiments since they contain fraud transactions.For both types, we divide respective data sets into three splits - 70 percent for training,15 percent for CV and 15 percent for testing purposes.We use stratified sampling in creating train/CV/test splits.Stratified sampling allows us to maintain the same proportion of each class in a split as in the original data set.

5 Methods

5.1 Logistic regression

It is used to classify data with linear decision boundary. It takes in input x, multiplies a weight vector θ Then it uses the sigmoid function to get the probability of an input belonging to a class(fraud) in binary classification model.

5.2 Support vector machines



Support vector machines are supervised machine learning algorithms that can be used to solve classification and regression issues, with the latter being the more common application. It is capable of finding complex underlying patterns.When dealing with two-dimensional data, SVM attempts to partition the two classes of data along a hyperplane that accomplishes this as effectively as feasible.This optimises the margin between the support vectors on both sides by increasing the distance between the closest points (support vectors) to the hyperplane on both classes.

5.3 SVM with RBF kernel

RBF kernels induced in support vector machines induce non-linearity in the decision boundary.The kernel function for it

$$K(x,z) = exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$$

5.4 Random forests

The random forest creates forecasts by averaging the predictions of all decision trees it uses as input. It generally performs well with default parameters and has substantially higher prediction accuracy than a single decision tree.

5.5 Naive Bayes

Bayesian classification techniques are the foundation of naive Bayes classifiers. They rely on the Bayes theorem, an equation that explains the connection between conditional probabilities of statistical quantities. In Bayesian classification, our goal is to determine the likelihood of a label given a set of observed features. This probability is written as $P(L \mid \text{features})$.We can describe this in terms of more directly calculable values by using Bayes' theorem:

$$P(L \mid \text{features}) = \frac{P(\text{features} \mid L)P(L)}{P(\text{features})}$$

If we are trying to decide between two labels—let's call them L1 and L2,we compare $P(L_1 \mid \text{features})$ and $P(L_2 \mid \text{features})$.

5.6 Multilayer perceptron model

Multi layer perceptron is a feedforward neural network that takes a set of input and produces a set of outputs. It uses gradient descent using adam optimizer and cross entropy loss to refine the output. It improves the output with every epoch(a forward pass and a backward pass) and reduces the gap between the actual and predicted output.

5.7 Gradient boost decision trees

Gradient boosting decision trees merge a number of weak learners into a single strong learner. Individual decision trees are the weak learners in this situation. The trees are connected in sequence, with each tree attempting to reduce the mistake of the one before it.

5.8 Performance evaluation

The confusion matrix's output serves as the primary analytical tool for the vast majority of machine learning classifiers. True Positive (TP), the correctly classified fraudulent cases, False Negative (FN), the incorrectly classified legitimate observations, False Positive (FP), the incorrectly classified legitimate transactions, and True Negative (TN), the correctly classified legitimate observations.

Precision is the ratio of correctly predicted positive occurrences to the total of predicted positive observations. It is written mathematically as

$$Precision = TP/TP + FP$$

Recall is also the ratio of true positives to the total of true positives and false negatives. It is given mathematically by

$$Recall = TP/TP + FN$$

F measure is defined as the weighted average of precision and recall.It is given mathematically as

$$f1Score = 2 * Recall * Precision/Recall * Precision$$

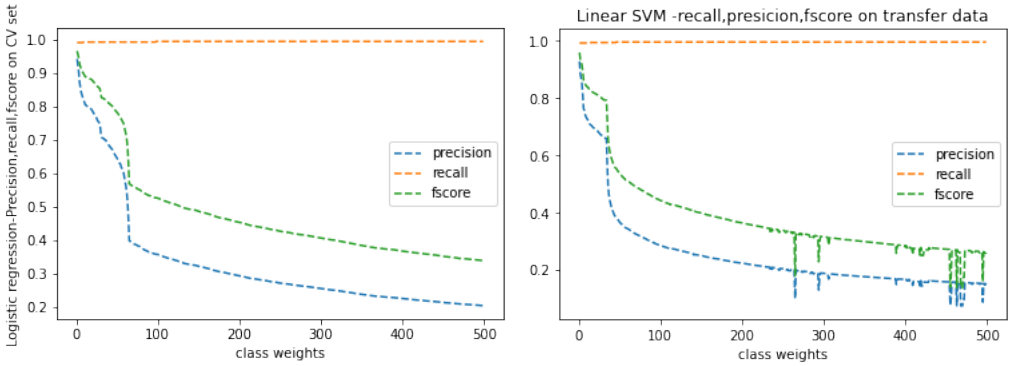
We chose to plot precision/recall curves (PRC) over ROC as PRCs are more sensitive to misclassifications when dealing with highly imbalanced datasets like ours.

5.9 Class weights approach

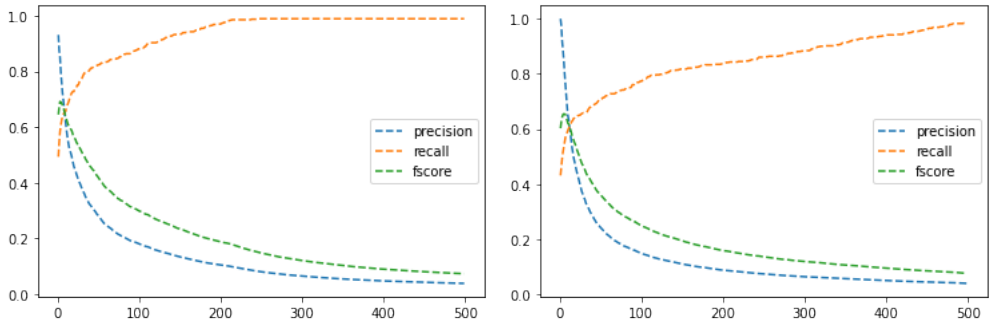
For each of the strategies, we give samples from the fraud and non-fraud groups varying weights. With only 0.13 percent of fraud transactions available to us, this strategy has been employed to address the problem of data imbalance. Catching possible fraud transactions is more important for a payments fraud detection system than making sure all legitimate transactions go off without a hitch. In our suggested methods, we penalise incorrectly classifying fraud samples more severely than incorrectly identifying non-fraud samples. For each strategy, we trained our models with greater class weights for fraud samples compared to non-fraud data.

6 Results

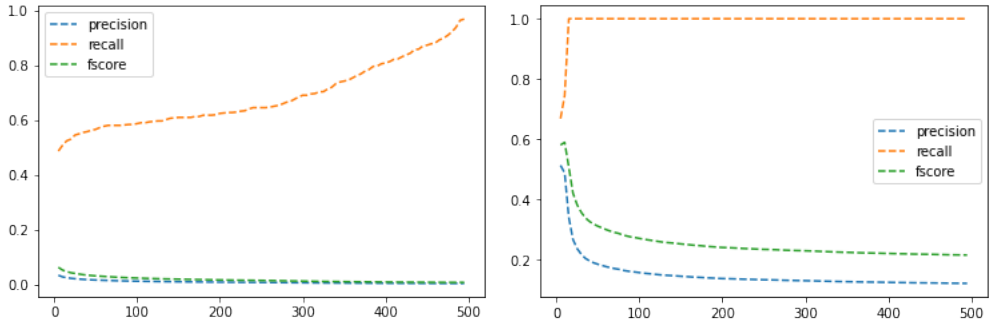
6.1 Results on validation for different Class weights



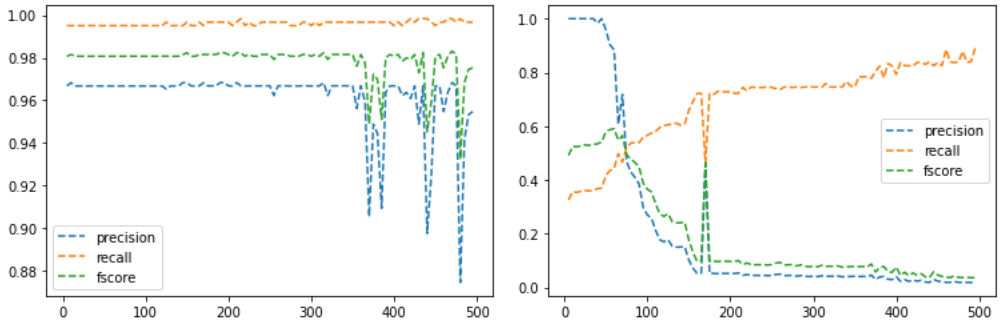
recall,precision,f1score on LR,SVM for transfer



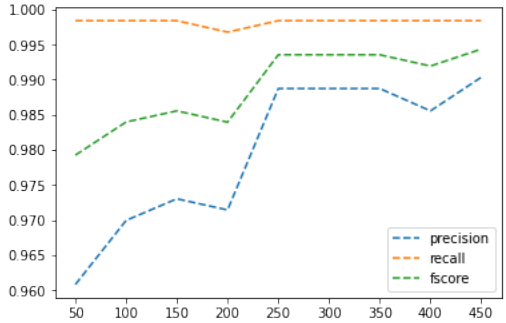
recall,precision,f1score on LR,SVM for cash out



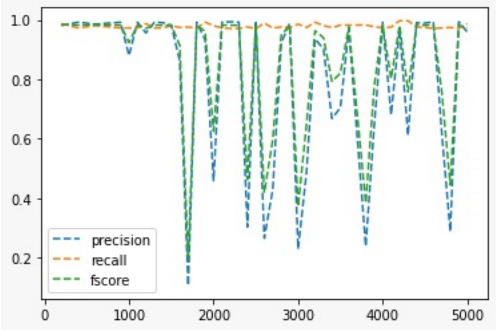
recall,precision,f1score on naive bayes for transfer,cash out



recall,precision,f1score on decision trees for transfer,cashout



recall,precision,f1score on naive bayes for transfer



recall,precision,f1score on mlp for transfer

The effectiveness of linear SVM and logistic regression is comparable (and hence similar linear decision boundaries and PR curves). For this collection of transactions, SVM with RBF offers a higher recall but a lower average precision. This result might have been caused by a non-linear decision boundary that was calculated using the RBF kernel function. However, if we are more tolerant of false positives, we can achieve high recall scores for all three methods.

6.2 Weights

To train each of our models, we used the class weight-based approach mentioned in the section above. For the fraud class samples, each model was trained numerous times with increasing weights. We assessed the performance of our trained models on CV split at the conclusion of each iteration. We selected the class weights for each model that gave us the highest recall on the fraud class with no more than 1 percentfalse positives.

Algorithm	Non-fraud	Fraud
Logistic Regression	1	70
Linear SVM	1	39
SVM with rbf kernel	1	16
Random forest	1	80
Navie Bayes	1	100

Weights obtained for transfer data

Algorithm	Non-fraud	Fraud
Logistic Regression	1	145
Linear SVM	1	132
SVM with rbf kernel	1	128
Random forest	1	65
Navie Bayes	1	55

Weights obtained for cash out data

6.3 Precision recall curves

Finally, we made predictions on our test dataset split using the models that were trained using the selected set of class weights. By calculating metrics such as recall, precision, f1 score, and area under the precision-recall curve, we assessed the performance of our models.

Algorithm	precision	recall	fscore	AUPRC
LR	0.405634	0.998267	0.576865	0.9216
linear	0.0645089	0.998267	0.611465	0.9064
rbf	0.387949	0.993068	0.557936	0.9867

precision,recall,fscore,auprc score for transfer data on test set

Algorithm	precision	recall	fscore	AUPRC
LR	0.390476	0.99308	0.560547	0.9323
linear	0.0124466	0.99827	0.599791	0.918
rbf	0.37947	0.991349	0.548851	0.9894

precision,recall,fscore,auprc score for transfer data on validation set

Algorithm	precision	recall	fscore	AUPRC
LR	0.392512	0.996288	0.563156	0.9296
linear	0.00977911	0.999629	0.599576	0.9159
rbf	0.391088	0.99369	0.561275	0.9876

precision,recall,fscore,auprc score for transfer data on train set

Algorithm	precision	recall	fscore	AUPRC
LR	0.32541	0.99481	0.490405	0.8832
linear	0.0105438	0.99827	0.413075	0.8291
rbf	0.258326	0.99308	0.41	0.9912

precision,recall,fscore,auprc score for cashout data on validation set

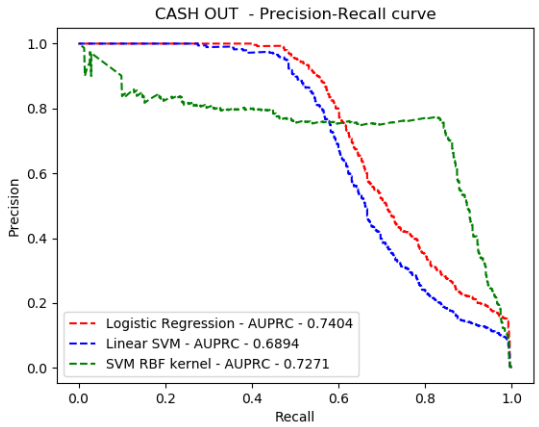
Algorithm	precision	recall	fscore	AUPRC
LR	0.327282	0.99703	0.492799	0.8838
linear	0.00955778	0.999629	0.418877	0.8332
rbf	0.270581	0.995546	0.425512	0.9903

precision,recall,fscore,auprc score for cashout data on train set

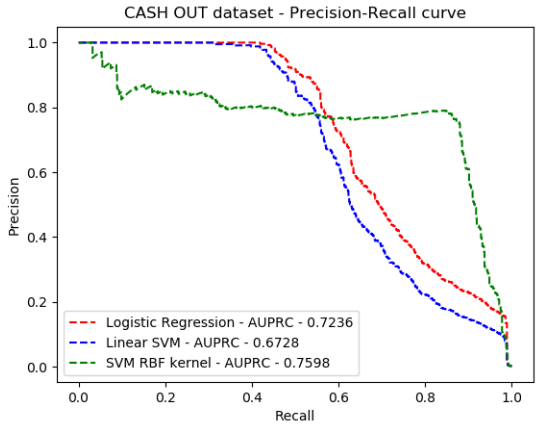
Algorithm	precision	recall	fscore	AUPRC
LR	0.336056	0.998267	0.502837	0.8733
linear	0.0467153	0.998267	0.420438	0.8204
rbf	0.268888	0.993068	0.423191	0.987

precision,recall,fscore,auprc score for cash out data on train set

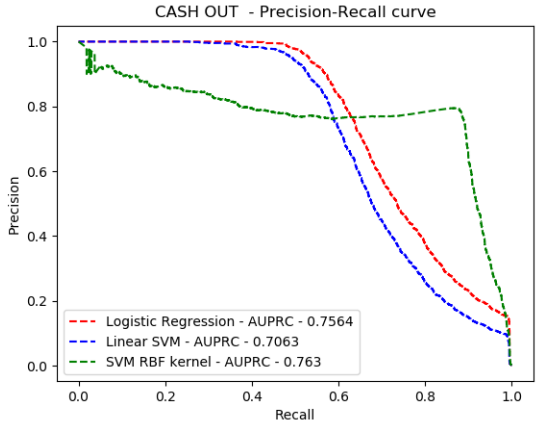
The following are precision recall curves with AUPRC scores for selected weights:



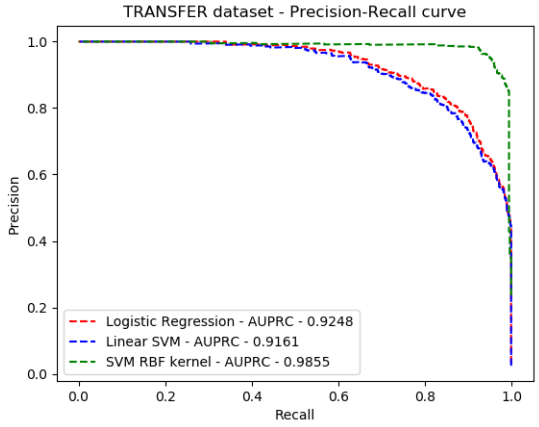
precision recall curve for LR,SVM,RBF for cash out data on test set



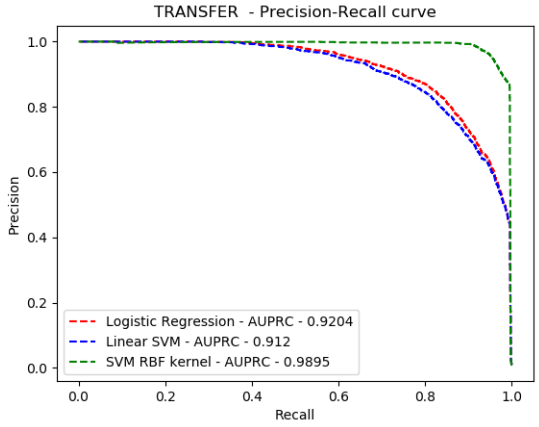
precision recall curve for LR,SVM,RBF for cash out data on validation set



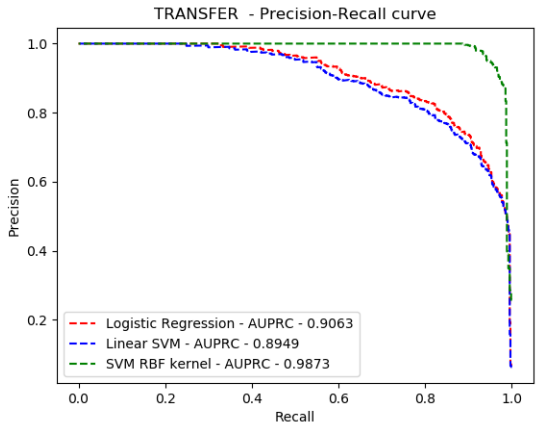
precision recall curve for LR,SVM,RBF for cash out data on train set



precision recall curve for LR,SVM,RBF for transfer data on validation set



precision recall curve for LR,SVM,RBF for transfer data on train set



precision recall curve for LR,SVM,RBF for transfer data on test set

For the fraud samples in our experiments, we employed increasing weights. At first, we thought of setting class weights to the imbalance ratio in our dataset. This method produced a large number of false positives (>1 percent), particularly for CASH OUT, but it also proved to provide good recall. We decided against using this method and instead fine-tuned our models by experimenting with various weight combinations on our CV split. We found that, overall, greater class weights resulted in higher recall at the expense of worse precision on our CV split.

7 Conclusion

In this study, we conducted tests to compare how well Logistic Regression, Gradient Boosted Decision Tree (tree-based), Random Forests, Support Vector Machine (linear,kernel-based) , mlp algorithms performed at predicting fraud in Mobile Money transactions, which is a common problem in developing nations. Accuracy, precision, recall, and F1-Score, which are regarded as fairly strong measures for assessing the effectiveness of classification algorithms from unbalanced datasets, were used to analyse the experiment findings. The outcomes of our tests demonstrated that the Random Forests and gradient boosted decision tree can accurately forecast fraudulent transactions.
Finally, this is solely a design/business decision in the real world and relies on how many false positives a payments company is willing to accept.