Credit Card Fraud Detection Using Machine learning

Shubham Koundinya
20718539,ECE
University of Waterloo
skoundinya@uwaterloo.ca

Abstract

Frauds are social nuisance applicable in wide variety of industries. This includes frauds in banking, finance, insurance and IOT devices. This project aims to capture credit card crimes and classifying them as fraudulent and normal transactions. This is a binary classification problem where the fraudulent transactions are represented by 1 and normal transactions are represented with 0. Data set for the project is taken from Kaggle Website. The datasets contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172 percent of all transactions. Since the dataset is highly skewed i.e majority of the classes are non Fraudulent, ADASYN (Adaptive Synthetic Sampling) was used to oversample the minority class. This project includes Initial Data Analysis, Data Preprocessing, Model Building and Model Evaluation. All the work was implemented in Python. Since the dataset is highly skewed accuracy should not be the criteria of comparing the results, hence different evaluation metrics were considered for comparing the results of the project This project follows the option B highlighted in the course - that is Empirical Evaluation of the content. This project implements 5 different Algorithms - Gaussian Naive Bayes, Logistic Regression, K Nearest Neighbors, Ensemble Methods which include hard and Soft Voting Classifier and finally a Deep learning Model. Finally a Comparative analysis for these 5 Algorithms on different Evaluation metrics is done.

1 Introduction

Frauds and anomaly detection are one of the challenges applicable to a wide variety of industries including banking, finance, insurance, and IOT devices.With big data revolution and petabytes of data being generated , Machine Learning can play an important aspect in identifying these anomalies , whether in banking, finance or IOT based devices. This project in particular focuses on financial fraud with respect to credit cards. With use of credit/ debit/ATM cards being dominated, the number of transactions made by these cards are in millions. Manually monitoring these transactions is very difficult. Hence Machine Learning comes into play , where by using different Algorithms we can form Complex hypothesis, based on different features, like location, spending habit of customer etc. The final model can then be deployed in real time and identify the fraudulent transactions and help the financial institutes and individuals as well.

The rest of this project report is organized as - Section2(Related Works)- This section briefly reviews some of the related work done in the field of credit card fraud detection. Section3 covers briefly the Background Section of each of the algorithms and different evaluation criteria to be considered. Some theory is highlighted for each classifier and then importance with respect to this project is highlighted. Section4 then covers the empirical results and the experiments done for each of the methods used in the project. This section also critically examines the performance and evaluation

.

of all the different Algorithms used in the project based on different evaluation criteria. Finally the Section5 provides the conclusions for the project.

2 Related Works

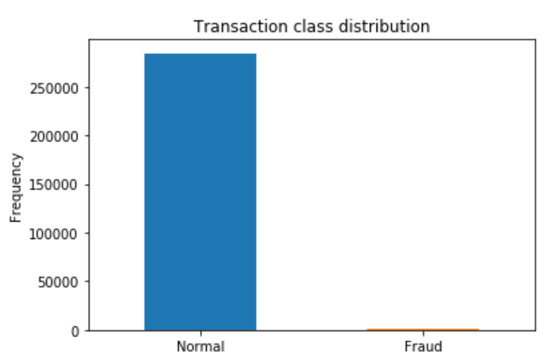
Lot of studies including comparative studies, analytical studies and experimental studies have been done on the credit card transactions in the past. S. Benson Edwin Raj et. all did an analysis on the study of Credit Card Fraud Detection Methods [1]. This was a comparative study highlighting different architectures and the relevant application areas of each of them. There was no experimental evaluation by the authors. They authors concluded that a Fuzzy Darwinian system performs the best amongst the other techniques used for fraud detection.They further concluded that the hybrid approaches for different classifiers can provide better results as compared to individual Algorithms Jyoti et.[3] all used a Decision Tree ased inductive Algorithm to carry out Fraud detection. Vijayshree[4] et. all used a SVM and Decision Tree based approach to identify the fraudulent and non fraudulent and non Fraudulent transactions.

John O. Awoyemi et all did a comparative analysis using Credit Card Fraud Detection Techniques[2]. They did a comparative study on three different classifiers , Naive Bayes, Logistic Regression and K nearest Neighbhors, showing that the KNN performs better than all the other classifiers. This project is mainly inspired by this paper. However in addition, to implementing three mentioned above classifiers, this project incorporate other approaches, including ADASYN(Adaptive Synthetic Oversampling), Ensemble Classifier(Hard Voting/Soft Voting) and Deep Learning based approach.

3 Background Section

This section provides an overview of all the major algorithms used in this project . The first section provides an overview of the dataset , since this is important to understand why different approaches have been used in this project and then a brief introduction of various algorithms.

3a.)Dataset Description:
Credit Card datasets are very difficult to obtain owing to privacy issues of customers. The dataset for the project is obtained from the Kaggle website[5]. This dataset contains the transactions made by European Cardholder. It has a total of 284,807 transactions. The dataset has a total of 31 features , out of which 30 features (V1, V2,....V28, 'Time','Amount') will serve as input features and the "Class" feature which signifies the class as Fraudulent (o/p=1) or Non Fraudulent (o/p=0) will serve as Target Feature.28 input Features V1, V2,... V28 are provided as a result of PCA transformation, as the exact value is not provided due to confidentiality issues. This is a binary classification problem. This dataset is highly skewed as the number of fraudulent transactions are very less. The number of frauds (o/p=1), accounts for the total of 0.172 percent of all the transactions. Out of a total of 284,807 transactions only 492 fraudulent transactions are present. The below figure shows the distribution of Fraudulent and non Fraudulent classes in the dataset.



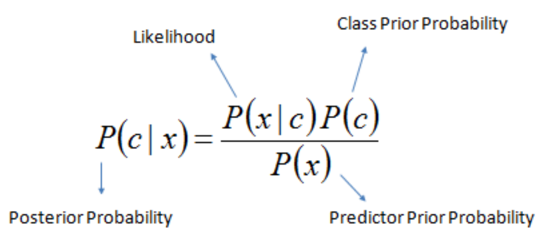
. The below figure gives a basic idea of the dataset by providing first few rows of the dataset:

```
df.head()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

3b.)ADAYSN Algorithm (To Handle Imbalanced Class):
As mentioned in the dataset description section , this is a highly imbalanced dataset as fraudulent transactions account for total of only .172 percent of the transactions, hence there is a need to handle imbalanced class. The project handles the imbalanced class by making the use of ADASYN : Adaptive Synthetic Sampling Approach for Imbalanced Learning[6]. The essential idea of ADASYN is to use a weighted distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data is generated for minority class examples that are harder to learn compared to those minority examples that are easier to learn[6].As a result ADASYN improves learning by reducing the bias introduced by class imbalance. The oversampling using ADASYN was done only on the training set and there was no oversampling on the test set. Training/ Test and Validation Splits are further discussed in Empirical Evaluation Section.

3c.Gaussian Naive Bayes):
Gaussian Naive Bayes is one of the Generative Classifiers which models joint distribution under distributional assumptions and then uses Bayes Rule To predict the target class. The formula is as highlighted in the below figure. It assumes conditional independence amongst features in the dataset, which is generally considered as a Naive Assumption for this classier. Once the Posterior for both the classes is obtained, it assigns the probability to each of these classes. This project uses Gaussian Naive Bayes for Binary Classification of Fraudulent and non Fraudulent classes.



3d.)Logistic Regression):
Logistic regression is a form of Discriminative classification algorithms that models posterior probability directly without any distributional assumptions. Once the Posterior for both the classes is obtained, it assigns the probability to each of these classes. In Binary classification, the class with the highest probability is assigned as the predicted class.The project uses Logistic Regression for modeling the probability of Fraudulent and non Fraudulent transaction and then gives predicted class as the result.

3e.)K Nearest Neighbors):
K Nearest neighbors is one of the non-parametric methods that makes no assumption on data and can be used for both classification and regression techniques. It belongs to the lazy class of Algorithms. The Algorithm can use Euclidean, Manhattan or Minkowski distance functions or weighted distance functions as well. This project used the Euclidean distance for calculating the number of nearest neighbors

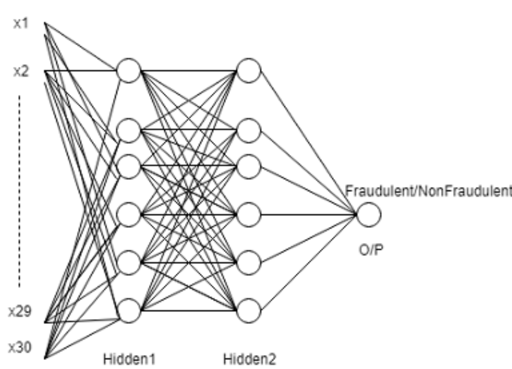
3f) *Voting/ Ensemble Classifier*:
The idea behind voting classifier is to conceptually combine different machine learning classifiers and use a majority vote(Hard Voting) to predict the class labels or the average predicted probabilities (Soft Voting) to predict the class labels. Such a classifier can be useful for a set of equally well performing model in order to balance out their individual weakness[7]. Below is a brief discussion on both these approaches

1. Hard Voting: In Hard voting the combined predicted class of the classifier is the majority or the model predicted by the combined classifiers.
e.g
Classifier A - Fraudulent Transaction
Classifier B - Non Fraudulent Transaction
Classifier C - Fraudulent Transaction
Then the net prediction of the combined classifier will be - Fraudulent Transaction, as its vote is more(2) as compared to Non- Fraudulent transaction whose vote is 1.

2. Soft Voting:
Soft Voting approach uses class label as the argmax of the sum of predicted probabilities,Specific weights (mentioned by w1 , w2 , w3) can be also assigned to a classifier to make it more dominant. The below table provides a simple example. Consider a binary classification problem with two classes, Fraudulent and Non Fraudulent . Consider there are three Classifiers mentioned by Classifier1, Classifier2, and Classifier3. Then the Class Predicted by the ensemble method will be the weighted sum of probabilities. In this example the class predicted will be Class A. Here w1, w2, w3 are equal to 1.

S.no	Fraudulent	Non -Fraudulent
Classifier1	w1 *0.8	w1* 0.2
Classifier2	W2 *0.7	w2 * 0.3
Classifier3	w3 *0.6	w3 * 0.4
Weighted Average	0.7	0.3

3g.) Deep Learning:
Deep Learning models are inspired by biological nervous systems, and theoretically they can learn any complex hypothesis.Deep Learning can easily learn multiple levels of abstraction. The main motivation for deep learning for this project was to consider how it performs as comparision to other classifiers mentioned in above sections. Due to time and resource constraints , not many deep learning architectures were evaluated but a simple architecture was used as shown in the below figure was implemented. 30 features as given as inputs , followed by 2 hidden layers of 6 units each, and a final output layer..Hidden units made use of relu activation units and the final layer used sigmoid activation. Adam optimizer was used for optimizing the loss function binary cross entropy loss, since this was a binary classification problem. Other Hyper parameters have been mentioned in the Empirical Evaluation Section.



3h.) Evaluation Metrics:
Since this project has a highly skewed data set, Accuracy alone is never a good judge of the classifier's performance. Hence different criteria have been considered for evaluation.
1. First a confusion matrix for each classifier has been evaluated. The confusion matrix for a classifier is as shown in the below figure.

		Prediction	
		0	1
Actual	0	TN	FP
	1	FN	TP

where TN = True Negative ,
TP = True Positive ,
FN = False Negative , also called as Type 2 error,
FP = False Positive , also called as Type 1 error.

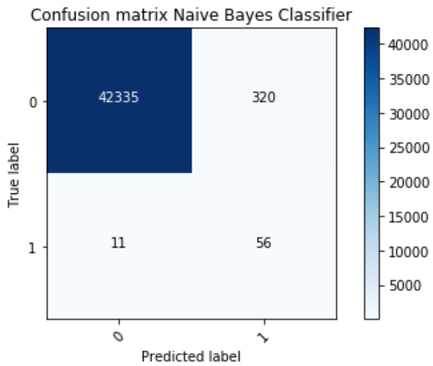
2. After confusion matrix is drawn , six basic matrices are used to draw the comparisons amongst different classifiers.
a. Sensitivity/TPR= TP/ (TP+FN) , TPR is also called as True Positive Rate
b. Specificity/TNR =TN /(TN+ FP) , TNR is also called as True Negative Rate
c. FallOut/FPR = FP /(FP+TN), FPR is also called as False Positive Rate
d. Miss Rate/FNR = FN /(TP+ FN), FNR is also called as False Negative Rate
e. Classification Accuracy = (TP+TN)/(TP + FP + TN + FN)
f. Classification error = (FP+TN)/ (TP +TN +FP +FN)

Since this is a binary classification problem, and the dataset is highly skewed the major criteria for a deciding the classifier should be Miss Rate i.e the Fraudulent Transaction, the classifier fails to catch, and FallOut i.e the Non Fraudulent Transactions that the classifier marked as Fraudulent. The Miss Rate and FallOut should be as small as possible for a classifier.

4 Empirical Evaluation

All the experiments in the project were done using Python3, Machine Learning Libraries like Scikit Learn, Keras and other data evaluation libraries like pandas. Since the dataset was considerably large, Train, Validation and test Split were done in the ratio 70:15:15 respectively. The hyper parameter tuning for the Algorithms was done on the Validation set and the accuracy was then recorded on the test set. Data preprocessing was done to oversample the minority class (Fraudulent transactions), with the use of a ADASYN, as mentioned in Section 3b. The oversampling was done only on the training set. 28 input Features V1, V2,... V28 are already present in the data set as a result of PCA transformation. The other two input features 'Time' and 'Amount' were standardized. This included centering and scaling of these two ('Time' and 'Amount' features).

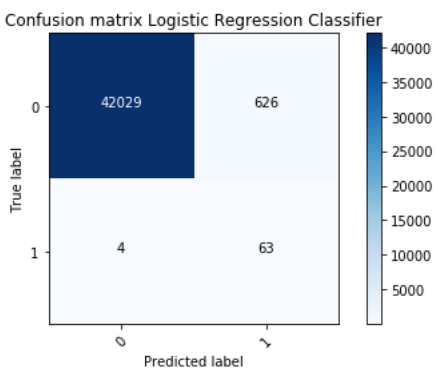
4a.) Naive Bayes Classifier :
Gaussian Naive Bayes was used for binary classification .The below figure highlights the Confusion matrix on the test data set and the table provides the results considering different evaluation metrices as mentioned in section 3 g. The accuracy of the classifier is good, but this can be because of the highly skewed data set. FNR/ Miss Rate i.e the percentage of the transactions that were Fraud but were not identified as Fraudulent is found to be = 16.4 percent and FPR/ FallOut which is the percentage of transactions that were not Fraudulent but were identified as Fraudulent is .7 percent. From the results we observe that the Miss Rate is too high and the FallOut rate is considerably good. Due to high miss rate, we can consider the classifier not doing well on test set.



Metrices	Naive Bayes
Classification Accuracy	.992
Classification Error	0.007
Sensitivity	.835
Specificity	.992
Fall Out	.007
Miss Rate	.164

4b.) Logistic Regression:

Logistic Regression was used for binary classification. There was no over fitting observed on Validation set, hence regularization was not used The Below figure provides the Confusion Matrix And the table provides different evaluation metrics observed on test set. The classification accuracy and the classification error are again very good. But the Miss Rate/ FNR is 5.9 percent, which is considerably better as compared to Naive bayes and the Fall Out rate is 1.4 percent. Overall the classifier can be considered as performing better than Gaussian Naive Bayes.

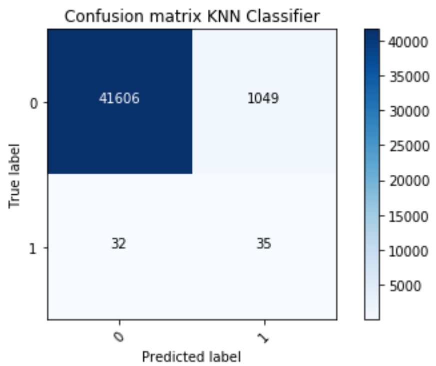


Metrices	Logistic Regression
Classification Accuracy	.985
Classification Error	0.014
Sensitivity	.940
Specificity	.985
Fall Out	.014
Miss Rate	.059

4c.) K Nearest Neighbors(KNN):

The number of nearest neighbors for the classifier were chosen using the validation set. The Mean miss classification rate was plotted for each of the classifier for k(nearest neighbors) ranging from 1-40 and the best k value was found out to be equal to 2. The figure below provides the Confusion Matrix and the table provides the results from different criteria of evaluation observed on the test set. The classification accuracy and the classification error are again very good owing to the highly imbalanced data set but the Miss rate/FNR is 47.7 percent which is very high. Fallout or FPR is

2.4 percent which can be considered as considerably small. But KNN is performing poor both as compared to Naive Bayes and Logistic Regression Classifier, as it has a high miss rate of 47.7 percent.

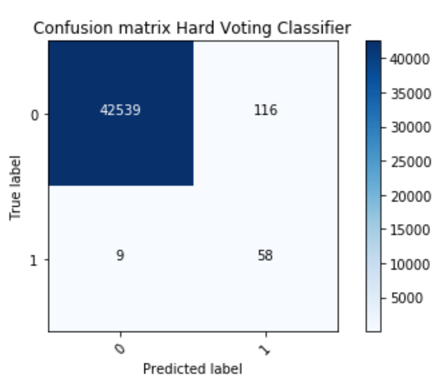


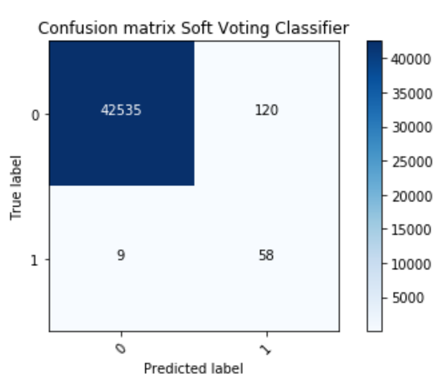
Metrices	KNN (n=2)
Classification Accuracy	.974
Classification Error	.025
Sensitivity	.522
Specificity	.975
Fall Out	.024
Miss Rate	.477

4d.) Ensemble/ Voting Classifier:

The Classifiers'- Gaussian Naive Bayes, K Nearest Neighbors and Logistic Regression were ensembled as per the Hard and Soft Classification Strategies mentioned in Section 3f. For the Soft Voting , each classifier was given a weight of 1. The left Confusion matrix is for the Hard Voting Classifier while the right Confusion Matrix is for the Soft Voting Classifier .

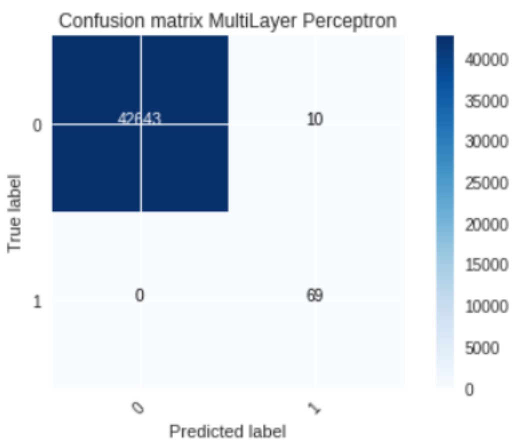
Below table highlights different evaluating criteria for Both Hard and Soft Voting Ensemble Classifiers. From the table we can conclude that the ensemble of Logistic Regression, Naive Bayes and K nearest Neighbors is performing better as compared to these individual classifiers. The miss rate for both the classifiers is 13.4 percent which is considerably reduced for the ensemble, as compared to individual classifiers. Comparatively ensemble methods are performing better as compared to all the classifiers individually.





Metrices	Hard Voting	Soft Voting
Classification Accuracy	.997	.997
Classification Error	0.002	.002
Sensitivity	.865	.8656
Specificity	.997	.997
Fall Out	.002	.002
Miss Rate	.134	.134

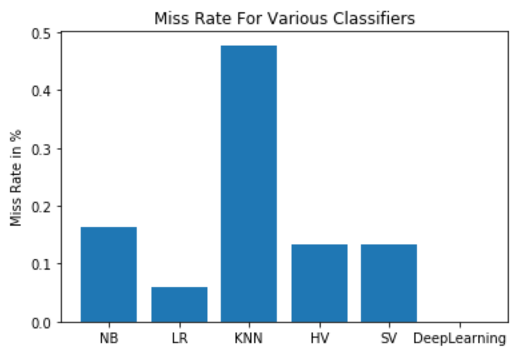
4e.)Deep Learning:
The architecture used for the project has been described in the section 3g . Adam optimizer was used with loss function as binary cross entropy loss. Batch size of 100 and 100 epochs were used for hyper parameter tuning on the validation set. The confusion matrix and the table highlights the results based on different evaluation criteria on test set. The figure clearly concludes the deep learning based approach remarkably outperforms all the other approaches including the ensemble of Naive Bayes, Logistic Regression and K Nearest Neighbors. The miss rate is 0, which concludes that there were no False Negatives i.e all transactions which were fraud were identified as Fraud. Similarly the False Positive Rate which identifies the percentage of non- Fraudulent transactions identified as fraud is nearly 0 percent. On similar lines Sensitivity and Specificity are best in Deep Learning amongst all the other classifiers.

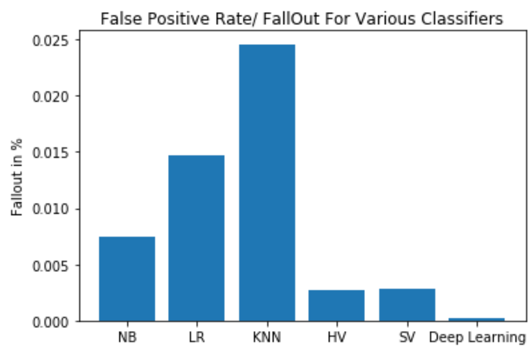


Metrices	Deep Learning
Classification Accuracy	.9997
Classification Error	.0002
Sensitivity	1.0
Specificity	.9997
Fall Out	.0002
Miss Rate	0.0

4f.) Comparative Analysis of All Classifiers:
The below table highlights the comparative Analysis of all the classifiers used in this project on the test set- Naive Bayes(NB), Logistic Regression(LR), K Nearest Neighbors(KNN),Ensemble Method(NB+LR+KNN) using Hard Vote, Soft Vote, and the deep learning approach.From the table we can see the NB, LR and KNN have good accuracy, but they have considerable miss rate and fall out. When these classifiers are ensembled(either Hard Voting or Soft Voting), Miss Rate and the Fall Out considerably goes small. But the experiments with Deep Learning gives remarkable results as the miss rate becomes zero, which concludes all Fraudulent transactions are caught by the Deep Learning classifier. Also the Fall Out in nearly zero. Further below the table, two main evaluation criteria - Miss rate and FallOut are plotted for the easy visualization and the comparative analysis of different classifiers. The results from the bar graphs clearly show the deep learning approach outperforming all the other classifiers, while the ensemble methods also outperform individual classifiers. The left bar graph is for the miss rate amongst classifiers and the right graph is for the Fallout amongst different classifiers.

Metrices	Naive Bayes	Logistic Regression	KNN	Hard Voting	Soft Vote	Deep Learning
Classification Accuracy	.992	.985	.974	.997	.997	.9997
Classification Error	.007	.014	.025	.002	.002	.0002
Sensitivity	.835	.940	.522	.865	.865	1.0
Specificity	.992	.985	.975	.997	.997	.9997
Fall Out	.007	.014	.024	.002	.002	.0002
Miss Rate	.164	.059	.477	.134	.134	0.0





Conclusion

This project did an empirical study and experiments on a binary classification problem , which was to identify Fraudulent transaction in a credit card data set. Five different classifiers were used which included- Naive Bayes, Logistic Regression, K Nearest Neighbor, Ensemble Classifier(Ensemble of Naive Bayes, Logistic Regression, KNN) , and finally a Deep Learning Architecture was evaluated. Accuracy of the Naive Bayes, Logistic Regression and KNN was good but the FallOut and Miss Rate , was very large, which highlights the weakness of these classifiers. Ensemble classifiers- Hard Voting and Soft Voting, performed comparatively better than individual classifiers. Finally a Deep learning based architecture was evaluated as it gave 0 Miss Rate on the test set. The Deep Learning architecture considerably outperformed all the other classifiers in all the evaluation criteria.

Some of the ideas for the future work of the project include - considering the problem as Unsupervised approach and not using the labels during training. Since the Fraudulent transactions are considerably small, we can use e.g Autoencoders to learn Non Fraudulent data by training only on Non Fraudulent data. During testing time , fraudulent transactions should have a high loss or reconstruction error. Similarly other unsupervised learning approaches can be evaluated.

The results obtained as a part of this project conclude that Machine Learning can greatly contribute to identify frauds in credit cards. Such approaches can similarly be applied to other industries where we need to capture anomalies and frauds.

Acknowledgement

I would like to thank Prof Sun Sun for giving us excellent understanding of Machine Learning , through solid theory and practical assignments. I would also thank TA's for help with assignments.

References

[1] A. Annie Portia S. Benson Edwin Raj. Analysis on credit card fraud detection methods. In *2011 International Conference on Computing Networking and Informatics (ICCNI)*, 2011.

[2] Samuel A. Oluwadare John O. Awoyemi, Adebayo O. Adetunmbi. Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 International Conference on Computing Networking and Informatics (ICCNI)*, 2017.

[3] Jyoti Gaikwad Amruta Deshmane Rinku Badgujar Snehal Patil, Harshada Somavanshi. Credit card fraud detection using decision tree induction algorithm. In *2015 International Journal of Computer Science and Mobile Computing*, 2015.

[4] Vijayshree et. all. Fraudulent detection in credit card system using svm and decision tree. In *International Journal of Science and Engineering Development Research*, 2016.

[5] Kaggle.com. *https://www.kaggle.com/mlg-ulb/creditcardfraud*. 2014.

[6] Edward A.Garcia Haibo He, Yang Bai and Shutao Li. *http://sci2s.ugr.es/keel/pdf/algorithm/congreso/2008-He-ieee.pdf*. 2008.

[7] *SKlearn ensemble Voting Classifier*.