IJCSE International Journal of Computer Sciences and Engineering Open Access

Survey Paper Vol.-7, Issue-8, Aug 2019 E-ISSN: 2347-2693

Supervised Learning Techniques for Identifying Credit Fraud

Advait Maduskar[1*], Aniket Ladukar[2], Shubhankar Gore[3]

[1,2,3]Information Technology, Thakur College of Engineering and Technology, Mumbai University, Mumbai, India

Corresponding Author: advaitmaduskar.1@gmail.com, Tel.: +91-97738-12140

DOI: https://doi.org/10.26438/ijcse/v7i8.247250 | Available online at: www.ijcseonline.org

Accepted: 15/Aug/2019, Published: 31/Aug/2019

***Abstract*—** Credit fraud is a broad term associated with theft or fraudulent transactions that involve the usage of a credit card. The fraud detection systems today are only capable of preventing one-twelfth of one percent of all transactions processed, which still results in huge losses. To the human eye, fraudulent transactions are indistinguishable from real ones. However, there are underlying patterns common to these transactions that can be recognized by machine learning algorithms. In this paper, we have trained supervised learning models on a dataset containing more than 280,000 transactions. We go on to evaluate the performance of each of these models on the dataset in terms of accuracy and precision and compare them with each other. With this, we show that the Random Forest model shows promising results for identifying credit fraud when trained on a labelled dataset.

***Keywords*—** Machine Learning, Supervised Learning, Fraud Detection, Random Forest, Regression, Classifier

I. INTRODUCTION

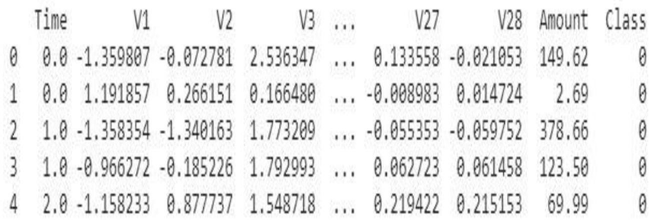
Credit card fraud is a wide-ranging term for theft and fraud committed using or involving a payment card, such as a credit card or debit card, as a fraudulent source of funds in a transaction. Machine Learning has permeated all walks of life and it is no surprise that credit card companies all over the world rely on it heavily to detect fraudulent transactions. [1]

With this project, we aim to find a suitable supervised learning model for identifying credit fraud. The algorithms we have used here are multiple linear regression, logistic regression, Naive Bayes classifier, Decision Tree classifier, XGBoost and Random Forest [2]. All these models are able to identify credit fraud with varying degrees of accuracy and precision. We tested these models on different splits of data and notice that the models generally give better performance when 70% of the dataset is used for training.

Through this paper, we have described the dataset that we used and its features, followed by the explanation of different supervised learning algorithms applied on the dataset. Having compiled the data regarding these algorithms' performances, we have presented a hypothesis that identifies ensemble modelling as the best algorithm for identifying credit fraud in this dataset.

II. DATASET USED

We have used the creditcard.csv dataset hosted on Kaggle. This dataset consists of 284,807 labelled transactions. Out of these, 492 transactions had been flagged as fraudulent. There are 30 independent variables which are all factors regarding a transaction and help determine its validity.



```
   Time        V1        V2        V3  ...       V27       V28  Amount  Class
0   0.0 -1.359807 -0.072781  2.536347  ...  0.133558 -0.021053  149.62      0
1   0.0  1.191857  0.266151  0.166480  ... -0.008983  0.014724    2.69      0
2   1.0 -1.358354 -1.340163  1.773209  ... -0.055353 -0.059752  378.66      0
3   1.0 -0.966272 -0.185226  1.792993  ...  0.062723  0.061458  123.50      0
4   2.0 -1.158233  0.877737  1.548718  ...  0.219422  0.215153   69.99      0
```

Figure 1: Snapshot of the dataset used

III. ALGORITHMS USED

Since the dataset used was a labelled dataset, the algorithms we used were all based on supervised learning techniques. We trained the algorithms on different proportions of training and testing sets and chose the "Class" label as the target variable for identifying fraud.

A. *Multiple Linear Regression*[3][4]

One of the models tested was multiple linear regression. There are 30 independent variables and 1 dependent variable. Due to the high dimensionality of data, separate variables were treated as multiple input features regressing over themselves to give a single output.

$$y_i = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n + \propto \qquad (1)$$

Comparing eq. (1) with our model, the target variable is “Class”, on the left hand side, while there are 30 input variables, on the right hand side. Since regression gives continuous values as output, we used a threshold of 0.5 to binarize the output in order to classify the transaction.

Despite to the binarization of continuous-valued outputs, we achieved a high accuracy of 99.88% with a relatively high precision of 83.89%

B. *Logistic Regression*[5]

Logistic Regression is used primarily when the dependent variable is binary in nature, as is in our case. A threshold of 0.5 is set for binarizing the continuous variable.

Due to the sigmoid function being used for predictive analyses in logistic regression, the output of logistic regression always lies between 0 and 1. The sigmoid function can be defined as follows:

$$sig(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

On applying logistic regression to the dataset, the results achieved were extremely promising, giving an average accuracy of 99.89% and due to nature of the sigmoid function, the precision achieved was relatively stable at 77.43%

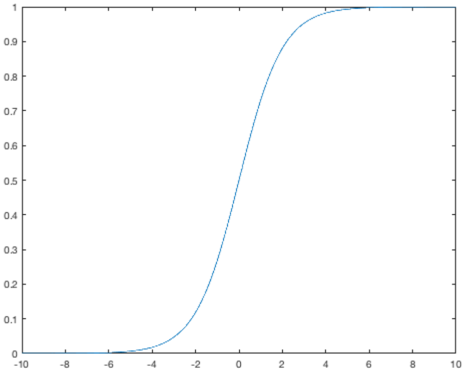


Figure 2: Sigmoid function used in logistic regression

C. *Decision Tree Classifier*[6][7]

A tree can be “learned” by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions.

Here, we use the decision tree for the purpose of classifying transactions as valid or fraudulent. Decision Trees can handle both numerical and categorical data and are suitable for handling highly dimensional data. After fitting the decision tree with our data, the leaf nodes acted as a binary classifier, where the node label "1" represented a fraudulent transaction and "0" denoted a valid one. Since our data has 30 input features, it is highly dimensional and so the decision tree algorithm performs relatively well.

Using decision tree over different splits of the entire dataset, we achieved an average accuracy of 99.89% giving rise to concerns over overfitting. This was confirmed by calculating the average precision which came out to be 54.54%. This can be attributed to the highly unbalanced nature of the dataset and the algorithm’s sensitivity to data.

D. *Naïve Bayes Classifier*[8][9]

Naïve Bayes Classifier is a probabilistic model that is derived from Bayesian statistics. It is based on apriori principle and works by assigning base probabilities to independent variables and using them to calculate conditional probabilities for a final outcome. Bayes theorem states that:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3}$$

Where P (A|B) is probability of event A occurring once B has already occurred.

When we trained this classifier on our data, the model grossly overfit due to both, the high dimensionality of data, as well as the size of the dataset. This was inferred as a result of our observations of the model’s high average accuracy of 99% while having extremely low mean precision of 8.02%

E. *XGBoost*[10][11][12]

XGBoost is based on optimizing a model’s computational performance using gradient boosting. Like random forest, it creates an ensemble of uncorrelated decision trees.

It is one of the better performing algorithms on the dataset since it uses bagging methods to optimize the results of multiple models. We achieve an average accuracy of 99.95% across multiple splits with a mean precision of 89.99%. This model is optimum for large datasets with high dimensionality since it is computationally efficient without trading off on prediction accuracy and precision.

F. *Random Forest*[13][14]

Random Forest algorithm results in the creation of multiple, uncorrelated decision trees. These trees work in coherence and use bagging as a means of creating an optimized model for classification.

When we trained the model using the dataset, with different train-test splits, we achieved an average accuracy of 99.95%, leading to concern about overfitting. However, bagging ensures that each of the trees is significantly different in structure to the others. This was supported by the average precision that was calculated to be 91.93%

IV. OBSERVATIONS AND ANALYSIS

After training the algorithms on the dataset across different splits of data, following results were achieved –

TABLE I: RESULTS ACHIEVED ON DATASET

Algorithm Used	Testing Set	Accuracy	Precision	Mean Squared Error
Decision Tree Classifier	20%	99.89%	57.65%	0.00109
	25%	99.86%	49.22%	0.00135
	30%	99.90%	57.81%	0.00103
	35%	99.89%	55.63%	0.00107
	40%	99.89%	52.38%	0.00109
Linear Regression	20%	99.86%	86.79%	0.00135
	25%	99.88%	82.61%	0.00125
	30%	99.88%	82.35%	0.00118
	35%	99.88%	83.16%	0.00116
	40%	99.88%	84.55%	0.00116
Logistic Regression	20%	99.89%	74.03%	0.00112
	25%	99.91%	80.90%	0.00091
	30%	99.90%	87.36%	0.00096
	35%	99.89%	72.52%	0.00111
	40%	99.89%	72.37%	0.00115
Naïve Bayes Classifier	20%	99.25%	10.22%	0.00746
	25%	99.08%	8.72%	0.00916
	30%	99.01%	7.88%	0.00989
	35%	98.88%	7.22%	0.01111
	40%	98.77%	6.08%	0.01231
Random Forest	20%	99.95%	91.11%	0.00047
	25%	99.95%	92.23%	0.00046
	30%	99.95%	94.26%	0.00046
	35%	99.95%	91.78%	0.00048
	40%	99.95%	90.29%	0.00051
XGBoost	20%	99.95%	89.83%	0.00049
	25%	99.95%	92.95%	0.00049
	30%	99.95%	91.66%	0.00044
	35%	99.95%	87.75%	0.00050
	40%	99.95%	87.75%	0.00051

The results achieved above denote that considering accuracy as well as precision, Random Forest algorithm performs the best on the dataset we have. This is due to the ensemble of uncorrelated trees and the use of bagging technique.

The Naïve Bayes classifier is the worst fit for the data giving an extremely low precision rate due to the dimensionality of the data. Similar in nature to Random Forest, Gradient Boosting algorithm also helps us achieve good results in terms of accuracy and precision.

V. CONCLUSION

Based on the testing we did on the dataset using different supervised learning techniques, we can state that ensemble learning models work better than standalone models when it comes to handling highly dimensional data. On the basis of our observation, an entropy driven model such as Random Forest or XGBoost outperforms probabilistic models such as a Bayesian classifier or simple logistic regression.

REFERENCES

[1] F. Misarwala, K. Mukadam, K. Bhowmick, "*Applications of Data Mining in Fraud Detection*", International Journal of Computer Sciences and Engineering, Vol.3, Issue.11, pp.45-53, 2015.

[2] D. Sathiya, S. V. Evangelin Sonia, *"A Comparative Study of Supervised Machine Learning Algorithm",* International Journal of Computer Sciences and Engineering, Vol.6, Issue.12, pp.875-878, 2018.

[3] Shen, A., Tong, R. and Deng, Y., 2007, June. Application of classification models on credit card fraud detection. In 2007 International conference on service systems and service management (pp. 1-4). IEEE.

[4] West, J. and Bhattacharya, M., 2016. Intelligent financial fraud detection: a comprehensive review. Computers & security, 57, pp.47-66.

[5] Chaudhary, K., Yadav, J. and Mallick, B., 2012. A review of fraud detection techniques: Credit card. International Journal of Computer Applications, 45(1), pp.39-44.

[6] Şahin, Y.G. and Duman, E., 2011. Detecting credit card fraud by decision trees and support vector machines.

[7] Sahin, Y., Bulkan, S. and Duman, E., 2013. A cost-sensitive decision tree approach for fraud detection. Expert Systems with Applications, 40(15), pp.5916-5923.

[8] Gadi, M.F.A., Wang, X. and do Lago, A.P., 2008, August. Credit card fraud detection with artificial immune system. In International Conference on Artificial Immune Systems (pp. 119-131). Springer, Berlin, Heidelberg.

[9] Raj, S.B.E. and Portia, A.A., 2011, March. Analysis on credit card fraud detection methods. In 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET) (pp. 152-156). IEEE.

[10] Zareapoor, M. and Shamsolmoali, P., 2015. Application of credit card fraud detection: Based on bagging ensemble classifier. Procedia computer science, 48(2015), pp.679-685.

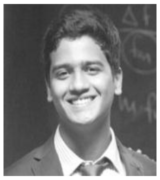
[11] Niimi, A., 2015, October. Deep learning for credit card data analysis. In 2015 World Congress on Internet Security (WorldCIS) (pp. 73-77). IEEE.

[12] Akila, S. and Reddy, U.S., 2017, November. Risk based bagged ensemble (RBE) for credit card fraud detection. In 2017 International Conference on Inventive Computing and Informatics (ICICI) (pp. 670-674). IEEE.

[13] Adewumi, A.O. and Akinyelu, A.A., 2017. A survey of machine-learning and nature-inspired based credit card fraud detection techniques. International Journal of System Assurance Engineering and Management, 8(2), pp.937-953.

[14] Tabassum, N. and Ahmed, T., 2016, March. A theoretical study on classifier ensemble methods and its applications. In 2016 3rd Internati onal Conference on Computing for Sustainable Global Development (INDIACom) (pp. 374-378). IEEE.

Authors Profile



Mr. Advait Maduskar is currently pursuing his Bachelor of Engineering in Information Technology at Thakur College of Engineering and Technology. He is in his senior year and is expected to graduate in May 2020. His research interests lie primarily in statistical machine learning and generative modelling, and has completed projects in these domains. He has been an active student member of the Association of Computing Machinery since 2017 and has conducted workshops for teaching junior students basics of machine learning.

Mr. Aniket Ladukar is a student at Mumbai University in the final year of his engineering studies in Information Technology. An expert at Object Oriented Programming, his interests lie in Statistics, Machine Learning and System Programming. He has completed projects in Machine Learning, IoT and Data Analytics. He has been a student member of ISTE and ACM since 2016 and 2017 respectively and has conducted technical seminars for junior students.

Mr. Shubhankar Gore is an undergraduate student pursuing his Bachelor’s degree in Information Technology from Mumbai University. He has completed projects in Deep Learning, Data Mining and IoT. He is primarily interested in Artificial Intelligence and Cognitive Science research and has been an active student member of ACM since 2017. He wishes to pursue higher studies in Applied Statistics and AI.