CREDIT CARD FRAUD DETECTION USING RANDOM FOREST CLASSIFIER

[1]M. SUSHMA, [2]INDU VADANA CIGA, [3]G.V. DEVAKI NANDAN, [4]A. SATISH KUMAR

Dept. of Information Technology, TKR College of Engineering and Technology, Hyderabad, Telangana

mandalreddysushma@tkrcet.com, indu.ciga@gmail.com, gvdnandan@gmail.com, satishk33108@gmail.com.

ABSTRACT

In huge organizations, transactions take place constantly. There are studies which show that fraudulent transactions take place quite often. This causes significant amount of damage to the customers and the organizations due to loss of trust. It is not sensible to investigate every transaction mainly because it is highly time consuming which leads to customers exasperated. In our project, we are focusing on credit card fraud detection in the real-world. There are already many approaches taken to reduce and detect the credit card fraud transactions. The results of these are not very accurate. Our approach in improving the accuracy is using popular machine learning algorithm that belongs to supervised learning technique. Based on Accuracy, specificity, sensitivity and precision of the techniques, its performance is evaluated.

1. INTRODUCTION

In the recent times, people are seeming to incline towards online money transfers. This statement holds true in many other scenarios, even communication is preferred online due to its ease. This change has a few limitations, the one with which we are more concerned is the fraud transactions which happen more frequently than we would think of. This spoils the relationship between the customer and the service provider. The hard-earned money of the people is being used by scammers. This has to be put to a stop. Credit card details of customers is being hacked and misused by these scammers. This may happen through the companies/organizations or due to poor security of the customers. To secure the customers personal information, organizations use multiple security programs in their source code but hackers are always finding new ways to get in. So, as a precaution we can observe patterns in the fraudulent and non-fraudulent transactions to alert the customers. This project aims to improve the system that observes these patterns. To achieve this, we have decided to use Random Forest algorithm that classifies the credit card dataset. The reason we have chosen Random Forest algorithm and not decision tree classifier is because Random Forest algorithm does not try to overfit to the trained dataset. This guarantees more accurate results.

2. LITERATURE SURVEY

Evaluating and Emerging Payment Card Fraud Challenges and Resolution (Pankaj

Richhariya and Prashant K Singh), 2014- This paper aims to speculate the solution of credit card fraud and various attributes of an effective payment through cards and its applied thoughts. It also reviews challenges associated with fraud transactions, metrics and mechanisms that can be used to resolve card fraud.

raining Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem (Zhi-Hua Zhou and Xu-Ying Liu), 2005 This paper examines the effect of sampling and threshold-moving in training cost-sensitive neural networks. This study suggests that some methods that have been believed to be effective in addressing the class imbalance problem may in fact only be effective in learning with imbalanced two-class data sets.

Learned lessons in credit card fraud detection fraud detection from a practitioner perspective (A.DalPozzolo, O.Caelen, A.LeBorgne and G.Bontempi), 2014 - This paper focuses on dealing with three crucial challenges from the practitioner's perspective: unbalanced Ness, nonstationary and assessment. This analysis is made possible by two real credit card datasets provided by our industrial partner.

Data mining for credit card fraud: A comparative study (Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel and J. Christopher Westland), 2011 - Random forest and support vector machines are two advanced data mining approaches. These two together are analyzed in this paper with the well-known logistic regression, in order to better detect and thus sway and execute credit card fraud. This paper is focused on data from an international credit card operation which is real-life data of transactions.

Adaptive Random Forests for Evolving Data Stream Classification (Heitor M. Gomes, Albert Bifet, Jesse Read, Jean Paul Barddal, Fabr´ıcio Enembreck, Bernhard Pfahringer, Geoff Holmes and Talel Abdessalem), 2017 - This paper presents the Adaptive Random Forest algorithm for classification of evolving data streams. In contrast to previous attempts of replicating Random Forests for data stream learning, Adaptive Random Forest includes an effective resampling method and adaptive operators that can cope with concept drifts of various types without including complex optimizations for various data sets.

3. PROPOSED SYSTEM

In proposed system, Random Forest algorithm is applied for classification of the credit card dataset. Overall, it is a collection of decision tree classifiers. Random Forest has an advantage over decision tree as it corrects the habit of overfitting to their training set. A subset of the training set is sampled randomly in order to train each individual tree and then build a decision tree. Each node then splits on a feature selected from a random subset of the full featured set. Even for large datasets with

many features and data instances, training is extremely fast in Random Forest and because each tree is trained independently of the others. Random Forest output depends on multiple decision trees which makes it unbiased. Therefore, the end results are more reliable. Minute change in the dataset will not affect the end result. Accuracy of this system is around 90%.

Modules

Data Collection

Data used in this paper is a set of product reviews collected from credit card transactions records. This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called *labelled data*.

Data Pre- processing

Organize your selected data by formatting, cleaning and sampling from it.

Three common data preprocessing steps are:

- *Cleaning*: Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonymized or removed from the data entirely.
- *Sampling*: There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

Feature Extraction

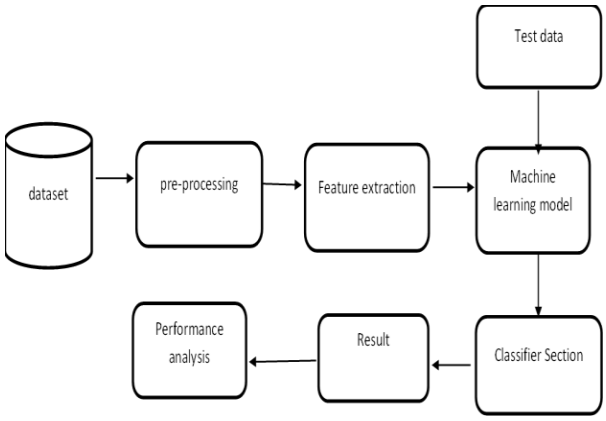
Next thing is to do Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally, our models are trained using Classifier algorithm. We use classify module on Natural Language Toolkit library on Python. We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen classifiers were Random Forest. These algorithms are very popular in text classification tasks.

Evaluation Model

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data

used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid over fitting, both methods use a test set (not seen by the model) to evaluate model performance. Performance of each classification model is estimated base on its averaged. The resull will be in the visualized form. Representation of classified data in the form of graphs. **Accuracy** is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

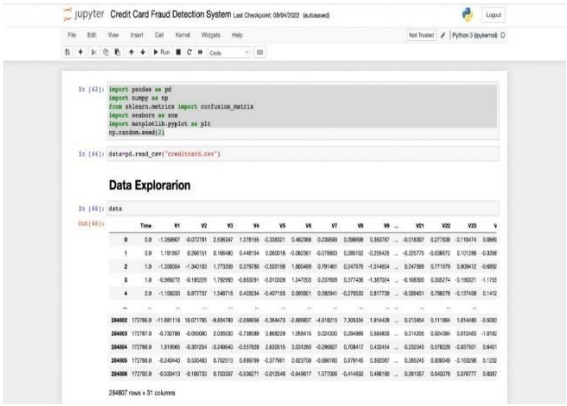
System Design

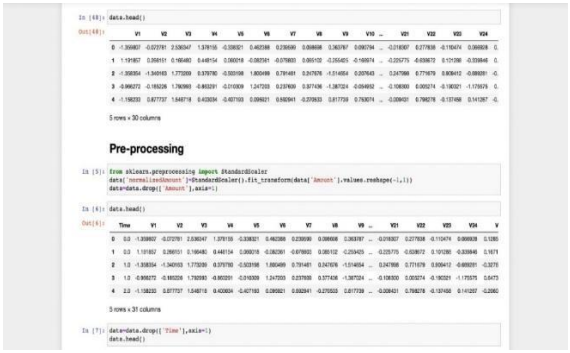


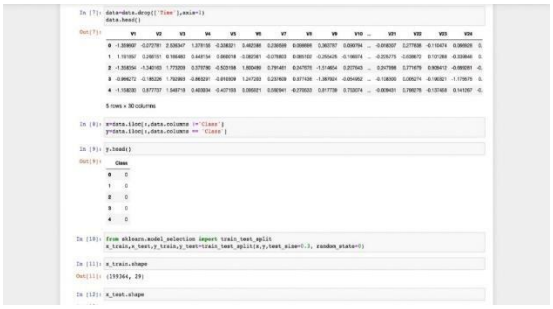
4. ALGORITHM

Decision Tree Analysis is a general, predictive modelling tool that has applications spanning a number of different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A decision tree is a tree-like graph with nodes representing the place where we pick an attribute and ask a question; edges represent the answers the to the question; and the leaves represent the actual output or class label. They are used in non-linear decision making with simple linear decision surface. Decision trees classify the examples by sorting them down the tree from the root to some leaf node, with the leaf node providing the classification to the example. Each node in the tree acts as a test case for some attribute, and each edge descending from that node corresponds to one of the possible answers to the test case. This process is recursive in nature and is repeated for every subtree rooted at the new nodes.

5.RESULTS AND DISCUSSION







SL #	TEST CASE NAME	DESCRIPTION	STEP NO	ACTION TO BE TAKEN (DESIGN STEPS)	EXPECTED (DESIGN STEP)	Test Execution Result (PASS/FAIL)
1	Excel Sheet verification	**Objective:** There should be an excel sheet. Any number of rows can be added to the sheet.	Step 1	Excel sheet should be available	Excel sheet is available	Pass
			Step 2	Excel sheet is created based on the template	The excel sheet should always be based on the template	Pass
			Step 3	Changed the name of excel sheet	Should not make any modification on the name of excel sheet	Fail
			Step 4	Added 10000 or above records	Can add any number of records	Pass

Test Case for Excel Sheet Verification

Here in machine learning we are dealing with dataset which is in excel sheet format so if any test case we need means we need to check excel file. Later on classification will work on the respective columns of dataset.

Test Case 1:

Random Forest Confusion Matrix,

CONCLUSION

The proposed paper evaluate that the Decision tree and support vector machine algorithm will perform better with a larger number of training data comparing to Adaboost classifier, but speed during testing and application will suffer. Application of more pre-processing techniques would also help. The SVM algorithm still suffers from the imbalanced dataset problem and requires more pre-processing to give better results at the results shown by SVM is great but it could have been better if more pre-processing have been done on the data.so, in proposed work we balanced the imbalanced data with up-sampling technique during pre-processing. We review the existing works on credit card fraud prediction in three different perspectives: datasets, methods, and

metrics. Firstly, we present the details about the availability of public datasets and what kinds of details are available in each dataset for predicting credit card fraud. Secondly, we compare and contrast the various predictive modeling methods that have been used in the literature for predicting, and then quantitatively compare their performances in terms of accuracy.

REFERENCES

1. *P. Richhariya and P. K. Singh, "Evaluating and emerging payment card fraud challenges and resolution," International Journal of Computer Applications, vol. 107, no. 14, pp. 5 – 10, 2014.*
2. *S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," Decision Support Systems, vol. 50, no. 3, pp. 602–613, 2011.*
3. *A.DalPozzolo,O.Caelen,Y.-A.LeBorgne, S.Waterschoot, and G.Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," Expert systems with applications, vol. 41, no. 10, pp. 4915– 4928, 2014.*
4. *C. Phua, D. Alahakoon, and V. Lee, "Minority report in fraud detection: classification of skewed data," ACM SIGKDD explorations newsletter, vol. 6, no. 1, pp. 50–59, 2004.*
5. *Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 1, pp. 63–77, 2006.*
6. *S. Ertekin, J. Huang, and C. L. Giles, "Active learning for class imbalance problem," The 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 823–824, 2007.*
7. *M.WasikowskiandX. -w.Chen,"Combating the small sample class imbalance problem using feature selection," IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1388–1400, 2010.*
8. *S. Wang and X. Yao, "Multiclass imbalance problems: Analysis and potential solutions," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 42, no. 4, pp. 1119–1130, 2012.*
9. *R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," Statistical science, pp. 235–249, 2002.*
10. *D. J. Weston, D. J. Hand, N. M. Adams, and C. Whitrow, "Plastic card fraud detection using peer group analysis," vol. 2, pp. 45–62, 2008.*
11. *E. Duman and M. H. Ozcelik, "Detecting credit card fraud by genetic algorithm and scatter search," Expert Systems with Applications, vol. 38, no. 10, pp. 13057–13063, 2011.*