ISSN 2348 - 9928
IJAICT Volume 8, Issue 01, January 2021 Doi:10.46532/ijaict-2020210101 Published on 05 (01) 2021

Insurance Fraud Detection Using Machine Learning

[1]Thanuj Kumar S, [2]Utsav Deep, [3]Syed Shoiab, [4]Syed Atif, [5]Tejas Bhatnagar and [6]T. Ramesh
[1,2,3,4,5,6] Department of Computer Science and Engineering, Presidency University, Bengaluru, India.
[1]thanujkumars98@gmail.com

ArticleInfo
International Journal of Advanced Information and Communication Technology
(https://www.ijaict.com/journals/ijaict/ijaict_home.html)
https://doi.org/10.46532/ijaict-2020210101
Received 10 Nov 2020; Revised form 31 Nov 2020; Accepted 22 Dec 2020; Available online 05 January 2021.

Abstract – Every year, the insurance industry losing billions of dollars due to fraud. The act when a person makes fake insurance claims to gain benefits, compensation & other advantages to which they are not entitled is known as Insurance Fraud. Nowadays insurance fraud detection is becoming a tedious problem for insurance companies to deal with as they need more investment and workforces to keep track of every transaction. In this paper, we are focusing on the major issue faced by insurance companies that is insurance fraud. we use the machine learning technique to detect insurance fraud based on the transactional data given by the insurance company. We build predictive models and compare their performance by calculation of confusion matrix then it is evaluated on various performance measuring parameters like accuracy, precision, recall, F1 score, and on AUC curve. SVM (Support Vector Machine) and XG Boost (Extreme Gradient Boosting) are the machine learning algorithms used. After model evaluation, we select the best model for prediction.

Keyword - Fraud Detection, Insurance Fraud, Machine Learning, Performance.

1. Introduction

The most significance of Machine Learning (ML) to use in Insurance Industry is to provide best possible predictions by building the model which processes the historical data of customer seamlessly to identify risk, claims and customer actions which further facilitates to make smart decisions and take appropriate actions. Insurance fraud affects enormously both financial aspects and everyday life. Frauds can reduce the trust in the company and degrades the growth of the company. This paper carries out a relative analysis of Insurance fraud detection using machine learning techniques to build machine learning models for prediction, which play a vital role in fraud detection, as it is implemented to extract and expose the hidden knowledge from exceptionally large data.

The rest of this paper is organized as follows. Section 2 briefly discusses the methods to be carried out. Section 3 Result analysis. Section 4 is the Conclusion; Section 5 discusses future enhancements.

2. Methods

Data Collection

Collecting data for training the machine learning models is the initial step in the machine learning pipeline. Data Collection is a method of gathering data from different sources to answer the relevant problem statement. The predictions made by models can only be as good as the data on which they have been trained. Some of the problems that can arise in data collection are unreliable data, Missing data, Imbalanced data. So we perform Data Preprocessing on the data we collected to incur these problems.

Data Preprocessing

Real-world-based raw data are likely to be unreliable as they are incomplete, inconsistent, and lacking in certain patterns of behavior. So, once we collect data, they are undergone pre-processing to cleanse the data and make data suitable for building ML models.

Pre-processing includes several techniques and actions:
Data cleaning: This action can be done manually or automated, to eliminate data that are incorrectly added or classified.
Data imputations: Most of the ML frameworks include techniques for balancing or filling the missing value with standard deviation, mean, and median.
Oversampling: Imbalanced or Biased datasets can be rectified by methods like repetition, and other over-sampling techniques and then added to the under-represented classes.
Data integration: Integrating multiple datasets to get a large dataset can be done to overcome incompleteness in a small dataset.
Data normalization: We reduce the data set size by reducing the order and magnitude by normalization as data size affects memory during the model training.

Exploratory Data Analysis

Exploratory Data Analysis, or EDA, is essentially a type of storytelling. It allows us to expose hidden insights and patterns, detect outliers and anomalies, test underlying

assumptions: determine optimal factor settings. Likely with visual graphs within data.

EDA is the initial step of the data modeling process. Data is collected and stored in a data repository. It could be a simple spreadsheet or complex database that comprises multiple spreadsheets in any format. Generally, the rows in a database are individual records while the columns are the various characteristics of each record. For analysis of the datasets, by the human eye (and brain) is nearly impossible because of vast data and information. That's the reason EDA comes into the picture. EDA uses various Data Visualization techniques to visually present the insights of data.

Clustering

Clustering is the assignment of partitioning the population or data points into several groups with the end goal that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. In the data distribution, we try to apply linear regression (refer to Fig 1), then we calculate the sum of residues (R1) here residue is the subtraction of predicted value from measured value concerning the best line.

The next approach is we are going to cluster the groups and find the sum of residues individually (R2 and R3) for the best fit line (refer fig 1.0), when we do a summation of R2 + R3, we observe that R1 > (R2+R3). It means when we do a cluster and tries to fit individual models to those clusters, we get the model or models which perform better. Here we use K means clustering, this algorithm finds values between two points using the method of 'Distance Measure' to cluster them. Here distance measure is 'Euclidean Distance'. For calculating the K value, we use the elbow method or python library knee for getting the optimal number of clusters needed for our task. finally, parse all the clusters to look for the best ML algorithm to fit on those clusters.

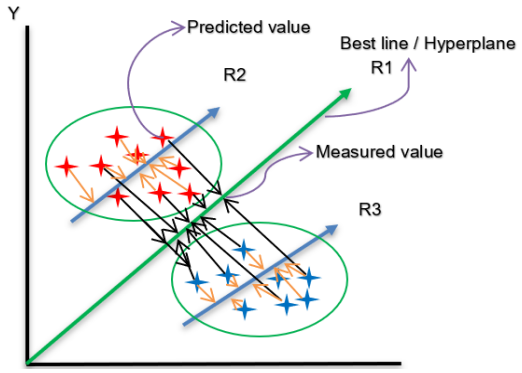

Fig 1. Clustering

Model Building

Before building a machine learning model, data is split into two parts called Training data and Testing data. For the purpose training of the model, we only expose the data which was for training and never allow testing data to be exposed. Once the model has undergone training using that data, we make use of the model to compute the predictions over the testing data, we will first define the independent variable and dependent variable X and y, respectively. We will now build the machine learning model using two different machine learning algorithms that are Support Vector Machine (SVM) and XG Boost (Extreme Gradient Boosting). Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for classification as well as regression problems. The main objective of this algorithm is to separate n-dimensional space into classes by best line so that we can insert the new data point in the appropriate class.

XGBoost stands for "Extreme Gradient Boosting". XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements Machine Learning algorithms under the Gradient Boosting framework. It provides a parallel tree boosting to solve many data science problems in a fast and accurate way. Now we will first import these and then will pass the training data to both models. After it gets trained, we will compute predictions over testing data.

Model Evaluation & Selection

Model Evaluation is a process of evaluating the models based on various performance measuring parameters. Evaluation of the model provides a clear picture of the model's efficiency and helps to select the best model for performing prediction. In this study, as we are using two algorithms, "SVM" and "XG Boost", so after building the models, these models are undergone a model evaluation phase. Scikit-learn python libraries are used extensively in this study for practical illustration. We use various performance measuring parameters like the Confusion matrix which further leads for calculations of Accuracy, Precision, Recall, and F1 Score. Also, we are using Area under the curve (AUC).

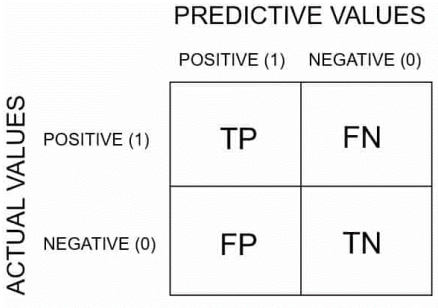

Fig 2. Confusion matrix

Fig 2 is the skeleton view of confusion matrix, row wise it's the actual values Column wise it's predicted values, with respect to positive and negative. (1,1) True Positive (TP), (1,0) False Negative(FN), (0,1) False Positive(FP), (0,0) True Negative(TN).

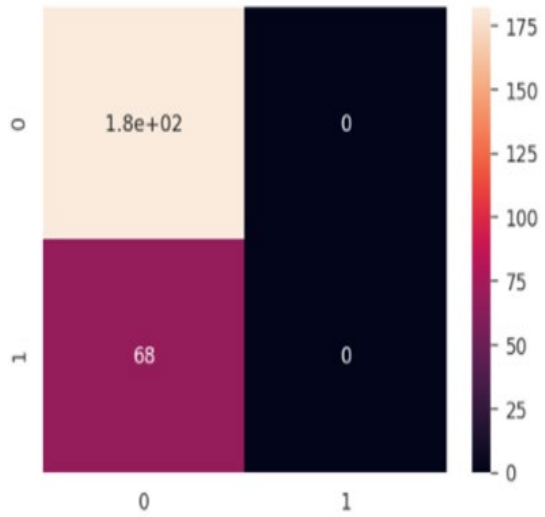

Fig 3. Confusion Matrix of SVM

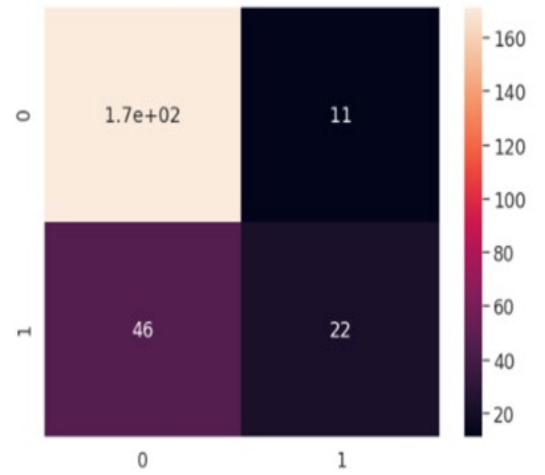

Fig 4. Confusion Matrix of XGB

We are using the confusion matrix python library to obtain the confusion matrices of both the algorithms, (Fig 2 and Fig 3). Based on the confusion matrices, we calculated the other parameters like accuracy, precision, recall, f1 score, and error rate. (Table 1.0) The accuracy of XG Boost (77.2%) is higher compared to SVM's (72.8%), though the Precision of SVM is 100% its Recall rate is 0.00% which shows an inconsistency, unlike XG Boost. F1 score is one of the great performance measurements in this XG Boost is shown more rate (85.7%) than SVM's (84.2%). The error rate of the SVM shows up to 27.2% and XG Boost is 22.8% which is less than SVM. In the above measurement, XG Boost seems to be more efficient than SVM.

Although evaluation is not yet finalized based on only these parameters because as we can see in the confusion matrix, data is not distributed uniformly which shows that our dataset is imbalanced therefore these above evaluation parameters are not enough to judge the model.

Table 1.0 Performance Measurement Table

Measure	Formula	SVM	XG Boost
Accuracy	$\frac{TP + TN}{TP + TN + FN + FP}$	0.728	0.772
Precision	$\frac{TP}{TP + FP}$	1.000	0.939
Recall	$\frac{TP}{TP + FN}$	0.000	0.323
F1-score	$\frac{2 * Precision * Recall}{(Precision + Recall)}$	0.842	0.857
Error rate	$\frac{FP + FN}{TP + TN + FN + FP}$	0.272	0.228

Therefore, we use parameters such as True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), False Negative Rate (FNR). (Table 1.1)

Table 1. 1 Performance Measurement Table

Measure	Formula	SVM	XG Boost
True Positive Rate	TP/P	1.000	0.939
True Negative Rate	TN/N	0.000	0.323
False Positive Rate	FP/P	0.370	0.252
False Negative Rate	FN/N	0.000	0.161

In Ideal scenarios, the True Positive Rate (TPR) and True Negative Rate (TNR) should be high, and False Positive Rate (FPR) and False Negative Rate (FNR) should be low. In table 1.1, we can see that the TPR of SVM is high (100%) than XG Boost (93.9%) but the TNR of SVM is 0.00% which is less than the XG boost is bearing 32.3%.

FPR of XG Boost (23.2%) is less (as recommended) than SVM (37.0%). But FNR of SVM (0.00%) is less than XG Boost (16.1%) in a small margin. In the above illustration, XG Boost seems to be efficient and consistent in all cases compared to SVM. We also using Area Under Curve (AUC), AUC represents the degree of separability. That means it tells that how much the model is capable of distinguishing the classes in our case it's fraud and Not fraud. Higher the AUC the better the model is at predicting fraud as fraud and Not fraud as Not fraud. An exceptional model has AUC near to 100%, and an ideal model has above 50%. But when the model is less than 50% or 50%, that means the model is incapable of distinguishing the classes

Table 1.2 Performance Measurement Table

Measure	Full form	SVM	XG Boost
AUC	Area under curve	0.500	0.631

In above table 1.2, we observe that the AUC rate of XG Boost is 63.1% which is ideal as it is greater than 50%, but whereas SVM is exactly 50%, which shows that in this case, SVM is incapable of distinguishing the classes, so SVM is not an ideal model to select.

By the process of evaluating, we yield a significant outcome on selecting the best model among SVM and XG Boost, considering the evaluation results we are selecting XG Boost which is shown more efficient than SVM.

Model Deployment

The concept of deployment in Machine learning refers to a model application for predicting by use of new data. Model building is not the end of the project, the whole idea of building the model is to discover hidden knowledge, the knowledge extracted needs to be presented and organized in a way that the customer can easily use it. Based on the specified requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable Machine learning process. In general, end-user or customers are the ones who use the application, so it has to be easy to operate and user-friendly.

For example, in our study Insurance companies may want to deploy a trained model or set of predictive models to quickly identify transactions, which have a high probability of being fraudulent. However, even if the analyst will not carry out the deployment effort the customer needs to understand upfront what actions will need to be carried out to make use of the created models. In this study, we designed the web application connected to our backend predictive ML models and deployed it in the cloud platform. We are using Flask which is a micro web framework used for web application development. And for the Cloud platform, we chose the Heroku cloud platform that allows deploying, management, and scaling of applications very quickly.

3. Discussion of Results

Here, we present the experimental comparison between the performance score of two algorithms that are used to build ML models to perform prediction on our dataset. Using evaluation technique, we found that the accuracy of XG Boost (77.2%) is higher compared to SVM's (72.8%) though we can't judge a model just by accuracy as it showed an imbalanced distribution of data in the confusion matrix, we performed various model evaluating parameters which briefly explained in Section 2.6. In addition to it, we also used Area Under Curve (AUC), AUC represents the degree of separability between classes. In which, the AUC rate of XG Boost is 63.1% which is ideal as it is greater than 50%, but whereas SVM is exactly 50%. With the evaluation process result, we select the best model which addresses our problem statement and efficient enough to perform the prediction. In the model selection phase of this study, we have chosen XG Boost over SVM based on the performance in the evaluation phase. And we have deployed our model as a Web application using the Flask web framework in the Heroku cloud platform.

4. Conclusion

In this study, we imported the relevant dataset which correlated to the problem statement then we used Machine learning techniques like Support Vector Machine (SVM) and XG Boost to build the predictive models. These models were undergone a model evaluation process to estimate accuracy, precision, recall, f1 score, error rate, and also AUC rate. By comparing the models on basis of the result of model evaluation, XG Boost outperformed SVM. Then deployment of the selected model is done by wrapping it as a web application by flask web framework, then later application is pushed to the cloud using Heroku cloud platform. Now this proposed system is ready to predict whether the claimed insurance is "Fraud "or "Not Fraud".

5. Future Enhancement

Enhancements are a never-ending process in technology as there is always a window for innovations whatsoever, so some of the enhancements of this Fraud detection system are, the system should be able to process the custom dataset uploaded at the front-end phase of the system. The system should be able to present the entire customer transaction history with visual graphs and remarks, When the system detects the fraud, it should automatically generate the complete report and send a notification to the In-charge immediately, followed by it should withhold the further transaction of that customer until the case is resolved.

References

[1]. S. Subudhi and S. Panigrahi, "Detection of Automobile Insurance Fraud Using Feature Selection and Data Mining Techniques," International Journal of Rough Sets and Data Analysis, vol. 5, no. 3, pp. 1–20, Jul. 2018.

[2]. M. Nur Prasasti, A. Dhini, and E. Laoh, "Automobile Insurance Fraud Detection using Supervised Classifiers," 2020 International Workshop on Big Data and Information Security (IWBIS), Oct. 2020.

[3]. X. Liu, J.-B. Yang, D.-L. Xu, K. Derrick, C. Stubbs, and M. Stockdale, "Automobile Insurance Fraud Detection using the Evidential Reasoning Approach and Data-Driven Inferential Modelling," 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Jul. 2020.

[4]. C. Yan and Y. Li, "The Identification Algorithm and Model Construction of Automobile Insurance Fraud Based on Data Mining," 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), Sep. 2015.

[5]. Bodaghi and B. Teimourpour, "Automobile Insurance Fraud Detection Using Social Network Analysis," Lecture Notes in Social Networks, pp. 11–16, 2018.

[6]. M. Kirlidog and C. Asuk, "A Fraud Detection Approach with Data Mining in Health Insurance," Procedia - Social and Behavioral Sciences, vol. 62, pp. 989–994, Oct. 2012.

[7]. V. Jain, "Perspective analysis of telecommunication fraud detection using data stream analytics and neural network classification based data mining," International Journal of Information Technology, vol. 9, no. 3, pp. 303–310, Aug. 2017.