International Journal of Communication and Information Technology

E-ISSN: 2707-6628
P-ISSN: 2707-661X
www.computersciencejournals.com/ijcit
IJCIT 2023; 4(1): 01-04
Received: 01-01-2023
Accepted: 05-02-2023

Saksham Shankar
Department of AIT-CSE, Chandigarh University, Kharar, Punjab, India

Corresponding Author:
Saksham Shankar
Department of AIT-CSE, Chandigarh University, Kharar, Punjab, India

Insurance fraud detection: Pre-emptive analysis and prevention

Saksham Shankar

DOI: https://doi.org/10.33545/2707661X.2023.v4.i1a.58

Abstract
Insurance companies have been facing the problem of false insurance claims, costing resources and money in detecting fraudulent claims among the many true ones. Many times, fraudulent claims are mistaken for genuine claims, and genuine ones mistaken for fraudulent claims. This paper aims to develop a Machine Learning algorithm to detect fraudulent claims and distinguish them with genuine claims. This project also aims to compare other algorithms and see which one is the most effective to deal with this growing problem that the insurance industry is facing.

Keywords: Machine learning, fraud, python

Introduction

Problem Definition

Fraud is defined as an act of intentional deceit to induce another person to part with some valuable possession. Insurance Fraud, therefore, is defined as an attempt to defraud the insurance process i.e. to fraudulently acquire extra compensation for any damages caused.

Problem Overview

The insurance industry has dealt with Insurance Fraud for a long time, and this type of fraud has cost a large amount of money. Current methods of detecting such fraudulent cases have been unsuccessful in having a high accuracy rate. This problem has gathered the attention of data analysts and machine learning experts to solve this problem. This attention has given rise to many proposals on how to deal with these fraudulent claims using Machine Learning, as manual searching and manually reviewing every claim is becoming increasingly difficult in the modern age, as more data needs to be collected and examined to find out the validity of even one claim.

In this research paper, we will be taking on the problem of Automobile Insurance Fraud, which is Insurance Fraud mostly done in the field of automobile accidents and incidents. The problem lies with detecting the valid cases.

Many Automobile Insurance Claims are based on fraudulent incidents, such as faked accidents, staged collisions etc.

Literature Review

Existing System

In the existing system, insurance fraud detection is a two- step process: first, we identify claims that are suspicious or have a higher probability of being fraudulent, and second, we use statistical analysis to identify suspicious claims. This is done by two methods: supervised or unsupervised.

In the supervised method, we analyze the records of both fraudulent and genuine claims, then perform statistical analysis to create a set of rules on which a claim can be judged as fraudulent or genuine. Its drawbacks are that it requires complete certainty on whether the analyzed claims are genuine or not. This also makes it so the rules are biased against one specific method of committing fraud without taking into account any new methods of committing fraud.

In the unsupervised methods, we detect abnormal claims, or claims that deviate from normal. This means that we do not rely on existing verdicts of a case's fraudulence, but we still make rules that determine whether a case can be fraudulent or not.

As we are relying on past frauds, this means that we will get a wide variety of methods to commit fraud, on which to base our rules, but again, the problem lies with the rules themselves.
The problem with such systems is that they do not have a high rate of success. Most claims identified as suspicious are completely natural, while some fraudulent claims slip through the cracks. In addition, both the methods do not say with any certainty whether a case is fraudulent or not. They only identify claims that should be inspected further, leading to waste of time and resources in investigating claims that may not even be fraudulent in the first place.

Proposed System
To propose an alternate to this flawed original system, we will first analyze existing proposals to supplement this system, namely the use of Logistic Regression, Random Forest Classifier, Decision Tree Classifier, K-Nearest Neighbors Classifier, Support Vector Machines Classifier, Extreme Gradient Boosting Classifier and Adaboost Classifier. Then, based on the results of that analysis, we will present a new solution.

Literature Review Summary
Abhijeet U, Amruta K, Rashmi B, Nandini ML, Fraud Detection and Analysis for Insurance Claim using Machine Learning.
Hritik K, Ranvir S, Fraud Claims Detection in Insurance Using Machine Learning.
Soham S, Insurance Fraud Detection using Machine Learning.
Srishti S, Anmol M, Farzil K, Ajay T, Kanak K, Insurance Fraud Identification using Computer Vision and IoT: A Study of Field Fires.
Rama DB. Insurance Claim Analysis Using Machine Learning Algorithms.
Najmeddine D. Extreme Gradient Boosting Machine Learning Algorithm for Safe Auto Insurance Operation.
Shimin L. An XG Boost Based System for Financial Fraud Detection.

Problem Formulation
The problem at hand is the issue of detecting fraudulent Automobile Insurance Claims.
Insurance Fraud is of two types: Hard Fraud and Soft Fraud. Hard Fraud refers to the deliberate planning of a fraudulent claim and invention of damage, loss of property or damage to a person, all to claim insurance money fraudulently. This type of fraud is the least common, because the second type of Soft Fraud is the most opportunistic type of Fraud. Soft Fraud is when an individual or party decides to misrepresent or exaggerate an existing claim to claim more money. This type of fraud is the most common because it is the most opportunistic type of fraud, as no person plans fraud from the start, and they usually take the opportunity to claim more money.
In the field of automobile insurance fraud, organized rings and groups dedicated to faking accidents to collect incorrectly assigned compensation are abundant. These rings have members within insurance claim adjusters as well as those people who can fast process fraudulent claims. However, there are also people who are opportunistic fraudsters, and misreport damages and claim more money than they are covered for in order to make a profit from their damage. Fraud committed by organized rings is called “hard fraud” and committed by opportunistic people is called “soft fraud”. At a glance, it is difficult to detect whether a fraud is of either category, so we will work on both.

Objective
The objective of this research paper is to analyze the efficiency of different Machine Learning algorithms in solving the current issue of Automobile Insurance Fraud Detection. We will be using methods such as Logistic Regression, Extreme Gradient Boosting (XGB), Support Vector Machine (SVM), K- Nearest Neighbors (KNN), Decision Tree Classifier, and Random Forest Classifier to analyze how we can effectively predict fraudulent cases, as well as use a Confusion Matrix to find out which method is the best in terms of predictions.

Methodologies
First, we will load the dataset into the algorithm. The algorithm is set such that we clean the dataset and visualize a few results first, to accurately determine what needs to be done exactly. Next, we prepare the dataset for machine learning models by preprocessing. After that, we will perform Linear Discriminant Analysis (LDA) on the dataset to standardize it for RF classifier algorithm.
The first step is to import and read the dataset. Once we read the dataset, we will first clean the dataset. This needs to be done as most datasets, including this one, may contain values with missing fields, incorrect fields, data that is incorrect or has been warped, and data that acts as an outlier and could skew the distribution and show incorrect results. These values spoil the dataset and allowed for inconsistencies and errors to emerge within the algorithm, therefore we will clean the dataset, first and foremost.
The next step is to actually look at the dataset and visualize it. This needs to be done as we cannot do any work on a dataset we do not know anything about, so we need to visualize how the dataset looks and the manner in which it is distributed among the different axes of influence i.e. the fields of the dataset. This will help us get a clearer picture of how the dataset is spread and gives us an idea of what we need to do next.
The next step is to accordingly encode any categorical values. In any dataset, it is often much easier to record categorical values in columns than numerical values, as not every parameter can be judged on a scale of numbers: some of these values need to be recorded in text form. However, a machine learning algorithm cannot work on text or categorical values. It requires numerical values as many of the algorithms require complex numerical calculations to be done. For this very reason we need to encode any categorical values into numerical values for the machine learning algorithm to be able to understand it properly.
In addition to this, we also need to do feature engineering i.e. selecting, adding and removing certain features based on redundancy or level of need in the final algorithm. This includes removal of unnecessary columns, removal of redundant columns, converting certain columns into an appropriate data type so that analysis can be done easily etc. [15].
Some columns may contain values that do not allow for correct encoding applications and obstruct the view of the actual correct values. These values need to be dropped as well.

For encoding purposes, we can use both Label and One Hot Encoding as appropriate; Label Encoding for when there is a certain hierarchy of categorical values, and One Hot Encoding when a hierarchical distribution needs to be avoided.

The next step involves preparations, if needed, for the dataset to be loaded in a sample algorithm. Let us take Random Forest for example. For a dataset to be passed to a Random Forest Classification algorithm, first we need to perform Linear Discriminant Analysis on that algorithm to prepare it for Random Forest.

Once Linear Discriminant Analysis is performed, the next obvious step is to perform the actual algorithm on the data, that is, the Random Forest classification algorithm. First, we will need to perform a split on the data, to split it into a training set and a testing set of the data. We do so by segmenting the data into two segments: one holding 80% of the data, and one holding 20% of the data. The larger segment is called the training data, and the smaller segment is called the testing data. This is important so that we can train the Random Forest Classifier and test it on the testing set to get the accuracy [14].

In addition to splitting data into a training and testing set, we also need to separate data into the independent variables and the dependent variable, as analysis will be performed on the independent variables and on their ability to correctly predict the dependent variable.

To get a feel for how good the fit of this classifier is, we will need various performance metrics, such as Cohen-Kappa Score, Accuracy Score, Recall Score, F1 Score and averages. In addition, we can do a visualization of the results using a Confusion Matrix. A Confusion Matrix displays the number of correct predictions segmented by each class of prediction, and can help visualize the accuracy of a certain machine learning model [8].

Now that we have done a sample analysis on the dataset, we need to compare the various algorithms and their effectiveness on detecting fraudulent cases. For this, we will need to scale the data appropriate. The reason for that is, while in Random Forest and Decision Tree classifiers we do not need to scale data, other algorithms may be skewed due to uneven distribution of data. This is because Random Forest and Decision Tree rely on if-else condition based modeling to classify data points, whereas most other algorithms require on the numeric values for classification. For this reason, we need to scale the training data and testing data i.e. the training set of independent variables and the testing set of independent variables. It is important that we leave the dependent variable data untouched so that we can accurately predict values [9].

In the next step, we apply various other algorithms to this data and analyze their effectiveness. This can be done by fitting various models to the data and drawing a box-and-whisker plot to show the distribution of accuracies across multiple iterations of each model, the outlier accuracies of each model, the standard deviations and variances of accuracies of each model, and the mean accuracies of each model [10].

Once we have done this, we can attempt to create our own Machine Learning algorithms by combining other algorithms and perform similar analysis on them by drawing Confusion Matrices and obtaining their accuracies. This is all done in the interest of obtaining the best machine learning algorithm for finding out fraudulent cases among the entire data.

Experimental Setup

The experimental setup is set up in Jupyter Notebook. We import the modules Pandas, Numpy, Matplotlib.pyplot, seaborn, xgboost, and itertools, and sub modules such as Ensemble, Preprocessing, Model Selection, Metrics and Discriminant Analysis, Linear Model, Tree, Neighbors and SVM from scikit- learn module, and rcParams from PyLab. We use the insurance_claims.csv dataset to perform our analysis. This dataset has fields “months as customer”, “age”, “policy number”, “policy bind date”, “policy state”, “policy csi”, “policy deductable”, “policy annual premium”, “umbrella limit”, “insured zip”, “insured sex”, “education level”, “occupation”, “hobbies”, “relationship”, “capital gains”, “capital loss”, “incident date”, “accident type”, “severity”, “incident state”, “authorities contacted”, “incident city”, “incident location”, “vehicles involved”, “property damage”, “bodily damage”, “witnesses”, “police report availability”, “total claim”, “property claim”, “vehicle claim”, “make”, “model”, and “fraud detected”. This dataset has 1000 entries and 40 columns.

In according with our above discussed methodology, we visualize the total number of actual frauds and actual genuine cases, then group the entire distribution based on different parameters. We look for unnecessary columns and rows with null values, and remove them, as well as encoding the various columns with only one or two types of entries [11].

Next, using K-Fold Cross Validation, we perform LDA on the dataset to prepare it for the RF classifier model. After the prepared dataset is loaded into RF model, we will create a Confusion Matrix to visualize the True Positives, False Positives, False Negatives and True Negatives for the Fraud Reported values, to see whether the model was able to accurately predict the fraudulence of each case or not [12].

Next, we apply the various above discussed models to the dataset and obtain Object data type variables containing each Machine Learning model. Then, we visualize the results of all the selected models across multiple tests, to compare the mean accuracies, deviated accuracies and variated accuracies of each model, to find out the final conclusion.

For the next step, we combine the above models in various ways to attempt at obtaining better results on inputting the machine learning models [13].

Finally, we visualize these additional results and compare them to our existing results, to decide which machine learning model is the best at determining the number of fraudulent cases among a collection of various insurance claim cases.

Conclusion and Future Scope

We found that the distribution of fraudulent and actual cases is very skewed. (Fig 1)

We also visualized distribution of the data across many axes. (Figs 2 - 14).

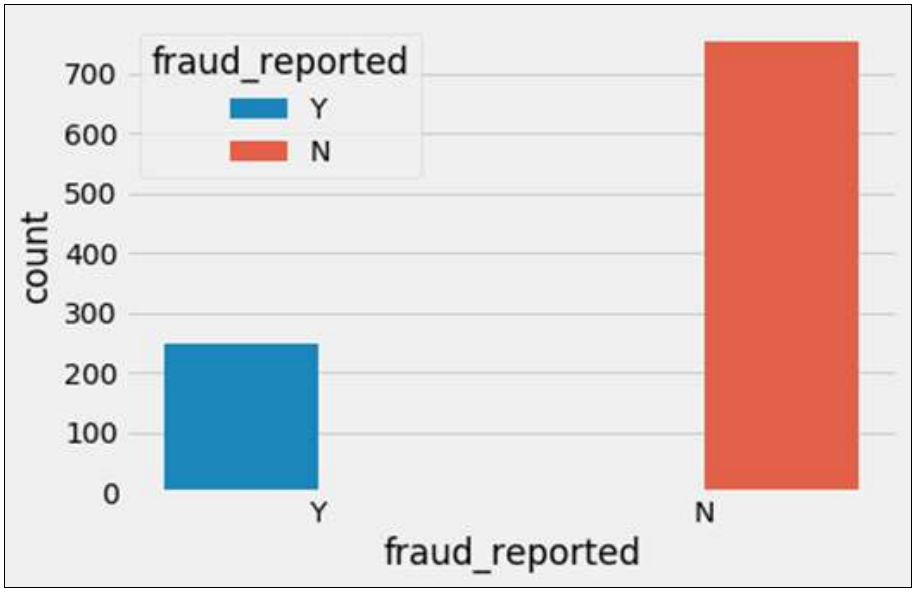

Fig 1: Distribution of cases

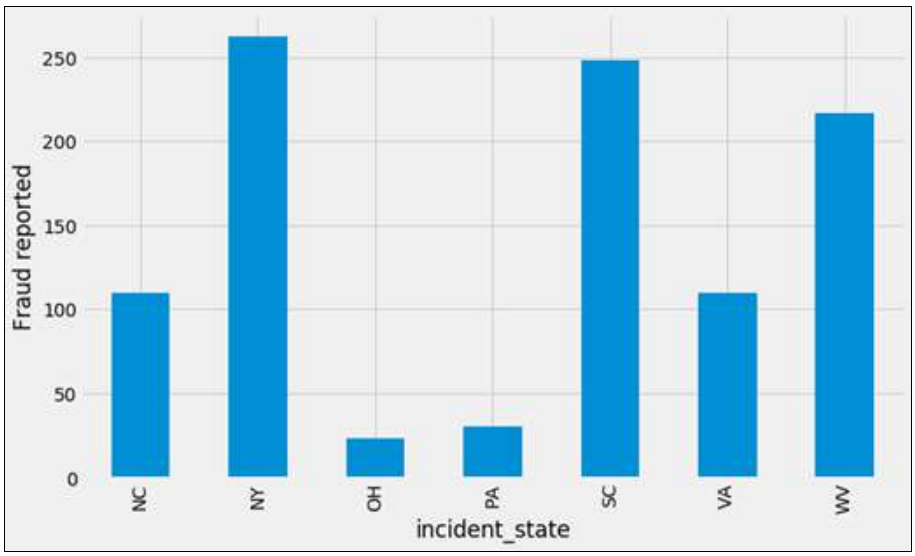

Fig 2: Distribution of data according to accident state

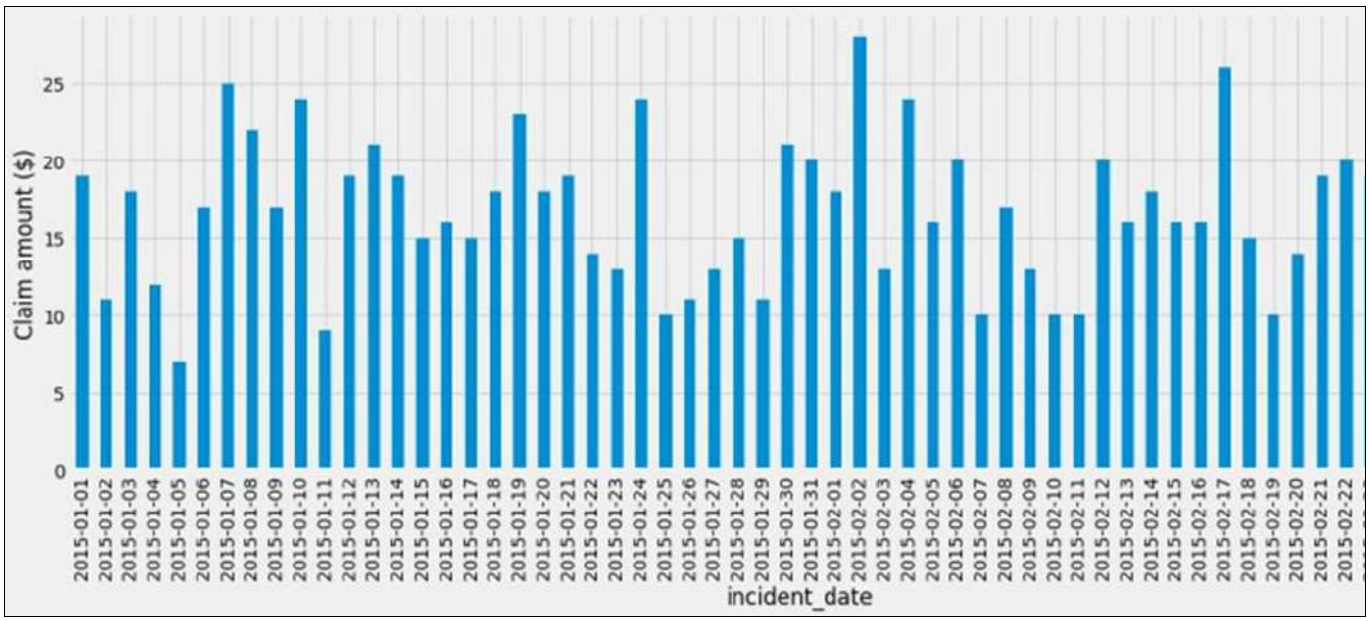

Fig 3: Distribution of data according to accident date

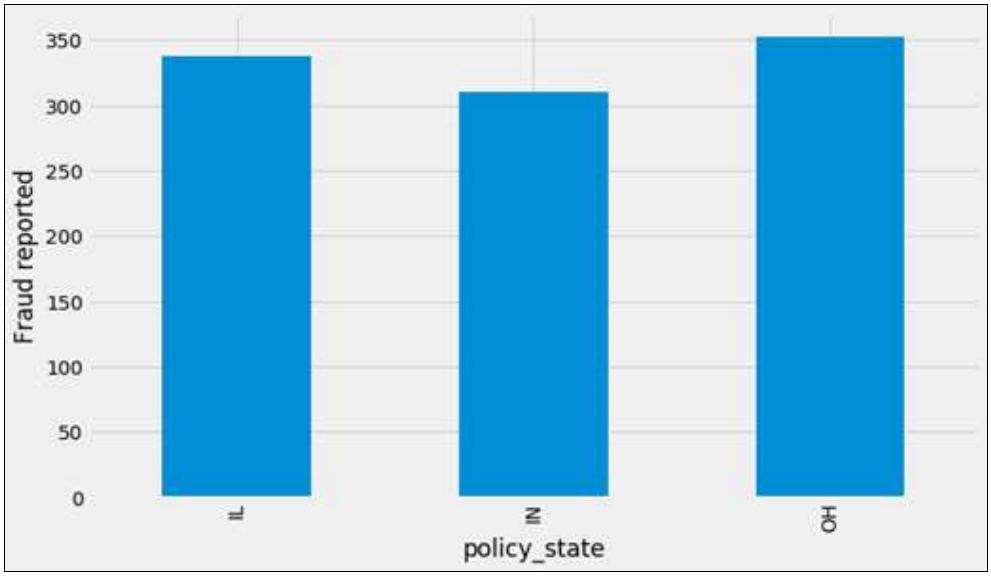

Fig 4: Distribution of frauds reported according to policy state

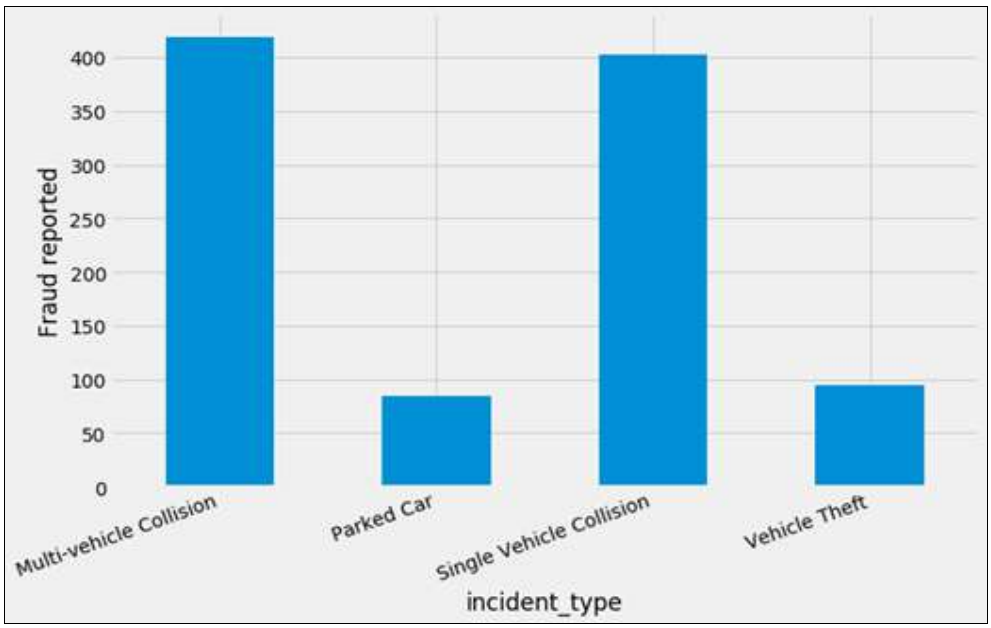

Fig 5: Distribution of data according to incident type)

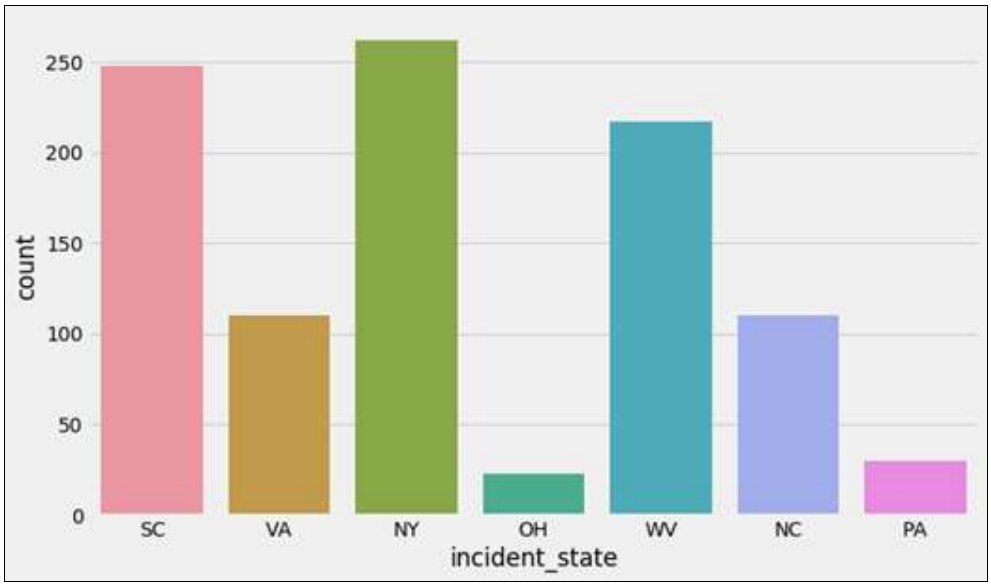

Fig 6: Distribution of all data according to incident state)

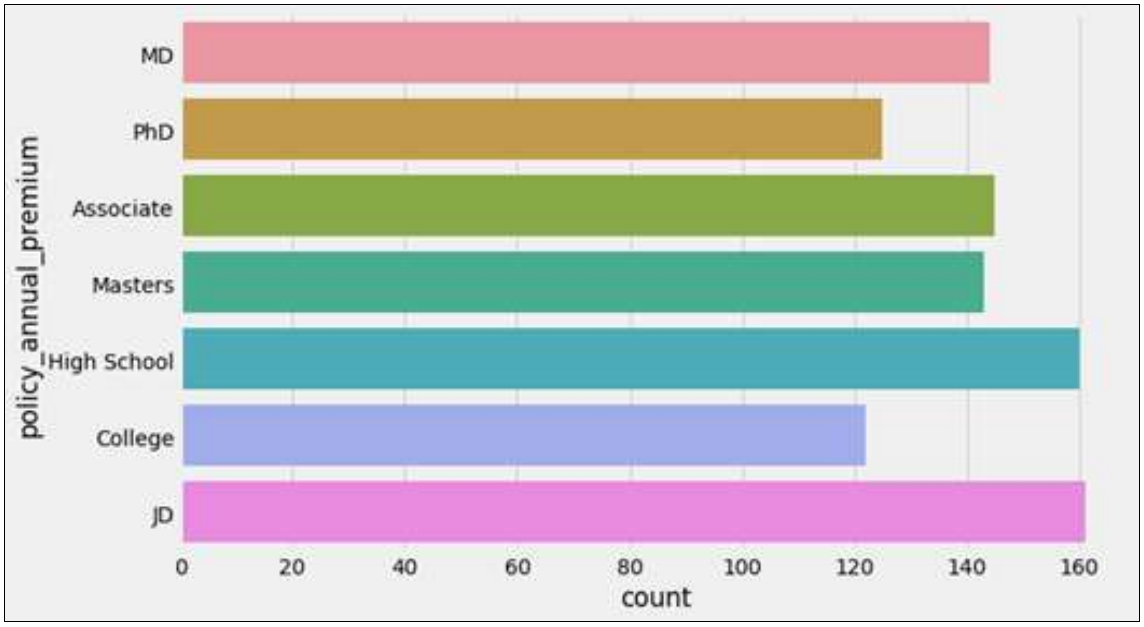

Fig 7: Distribution of data according to Education Level

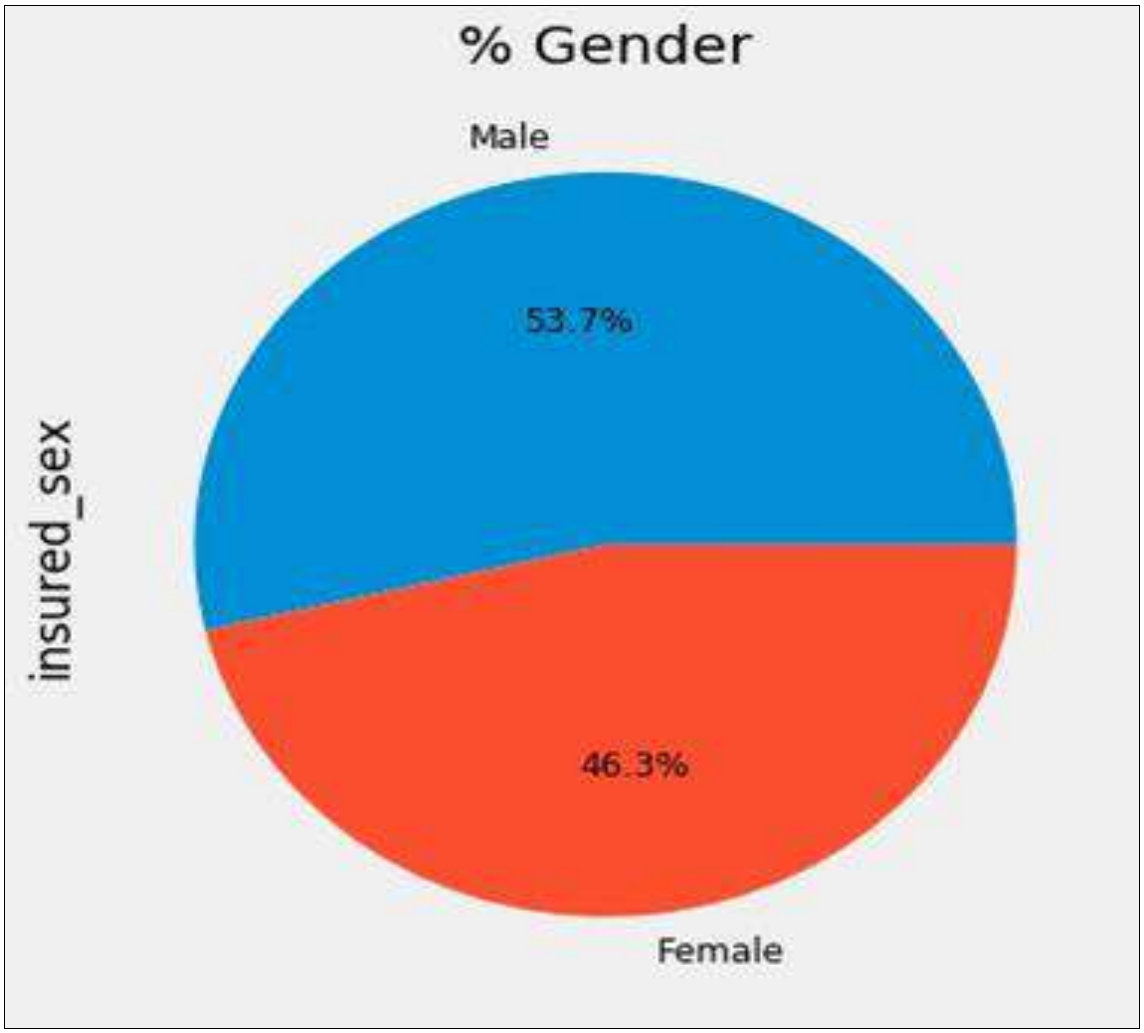

Fig 8: Distribution of data according to gender of driver

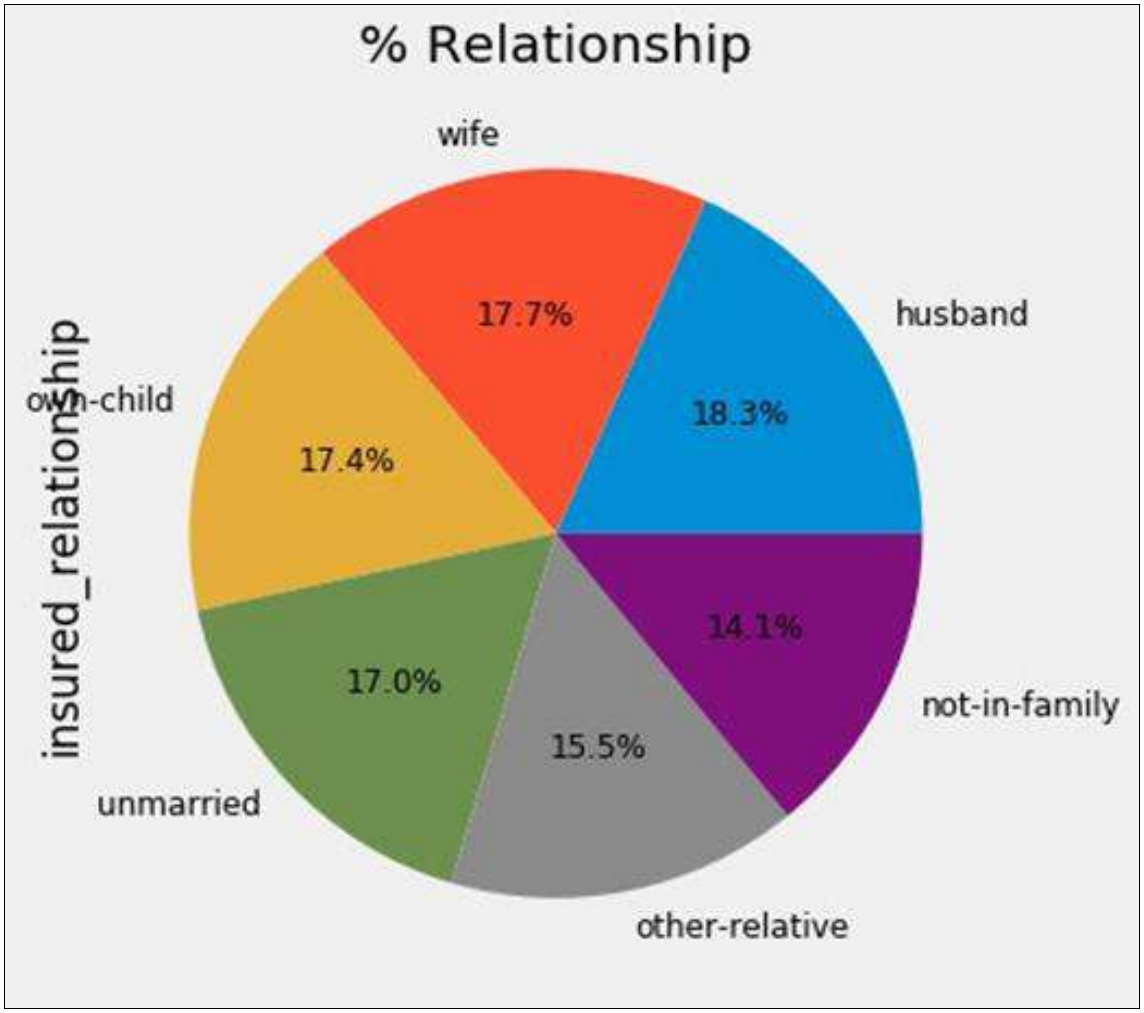

Fig 9: Distribution of data according to insured relation

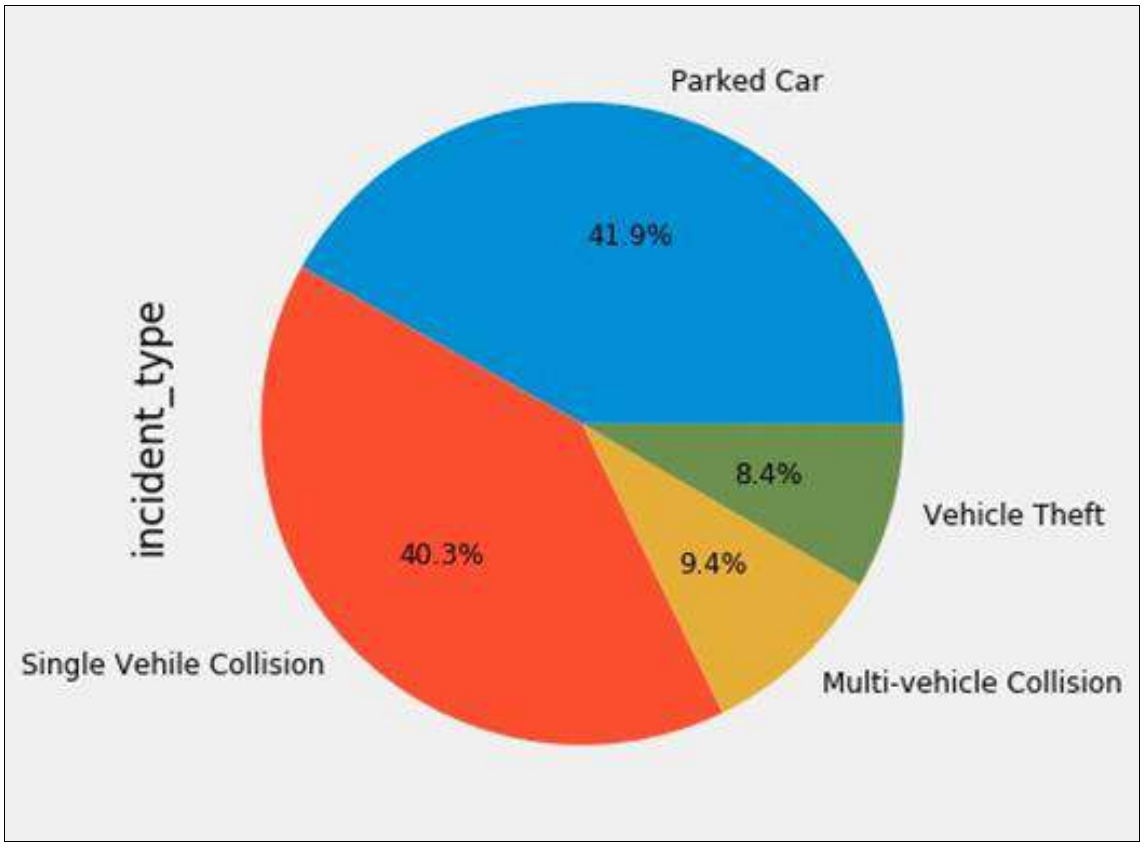

Fig 10: Distribution of data according to incident type

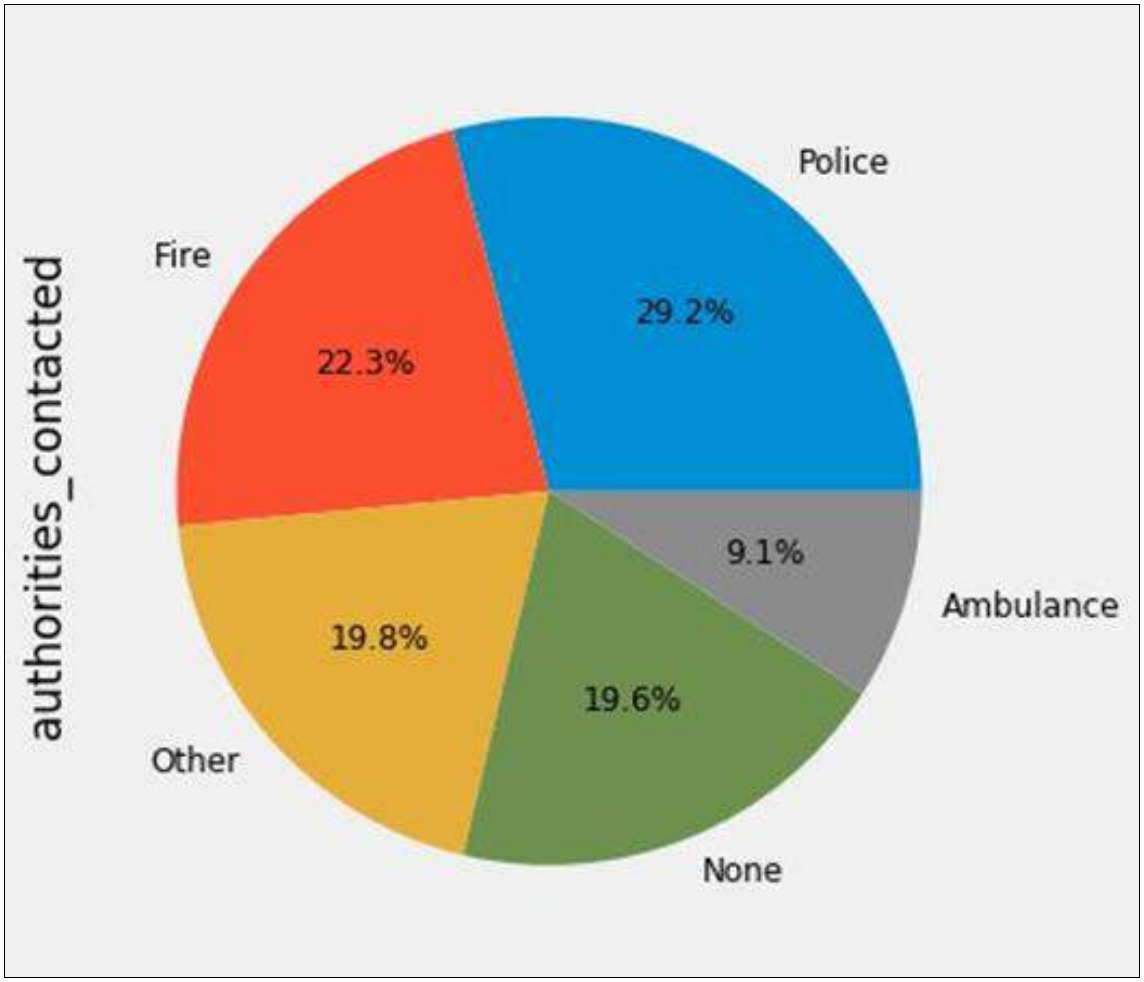

Fig 11: Distribution of data according to authorities contacted

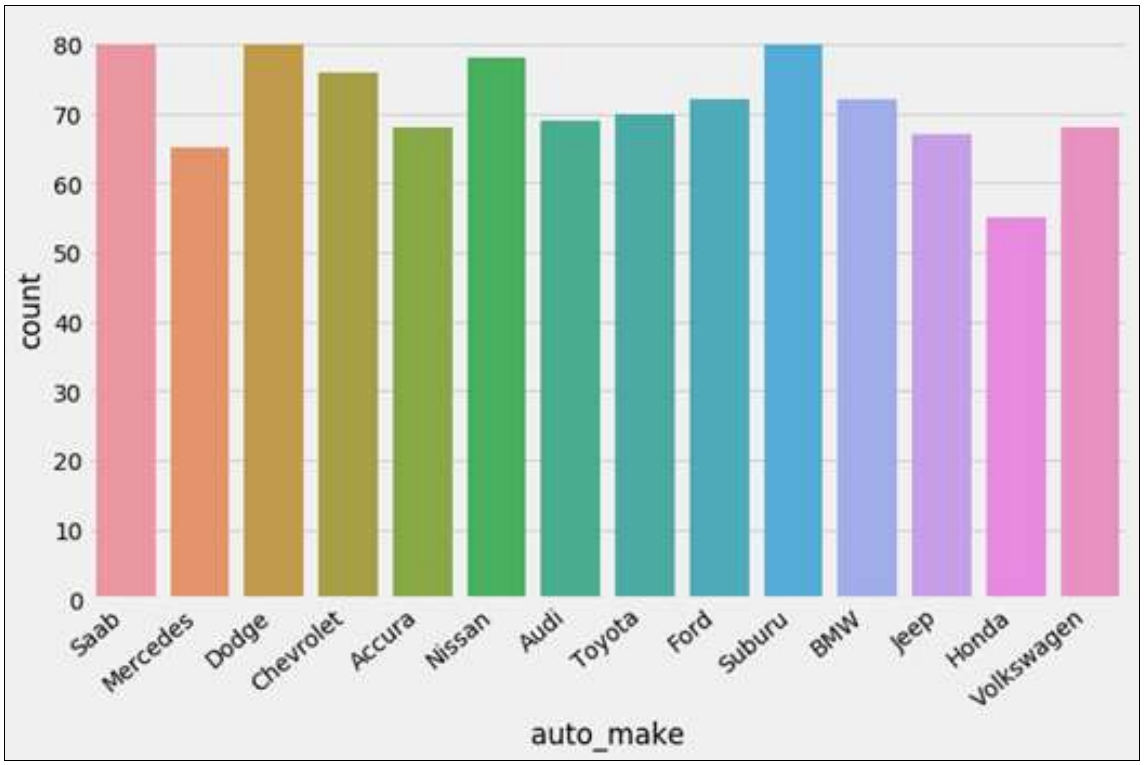

Fig 12: Distribution of data according to model of car

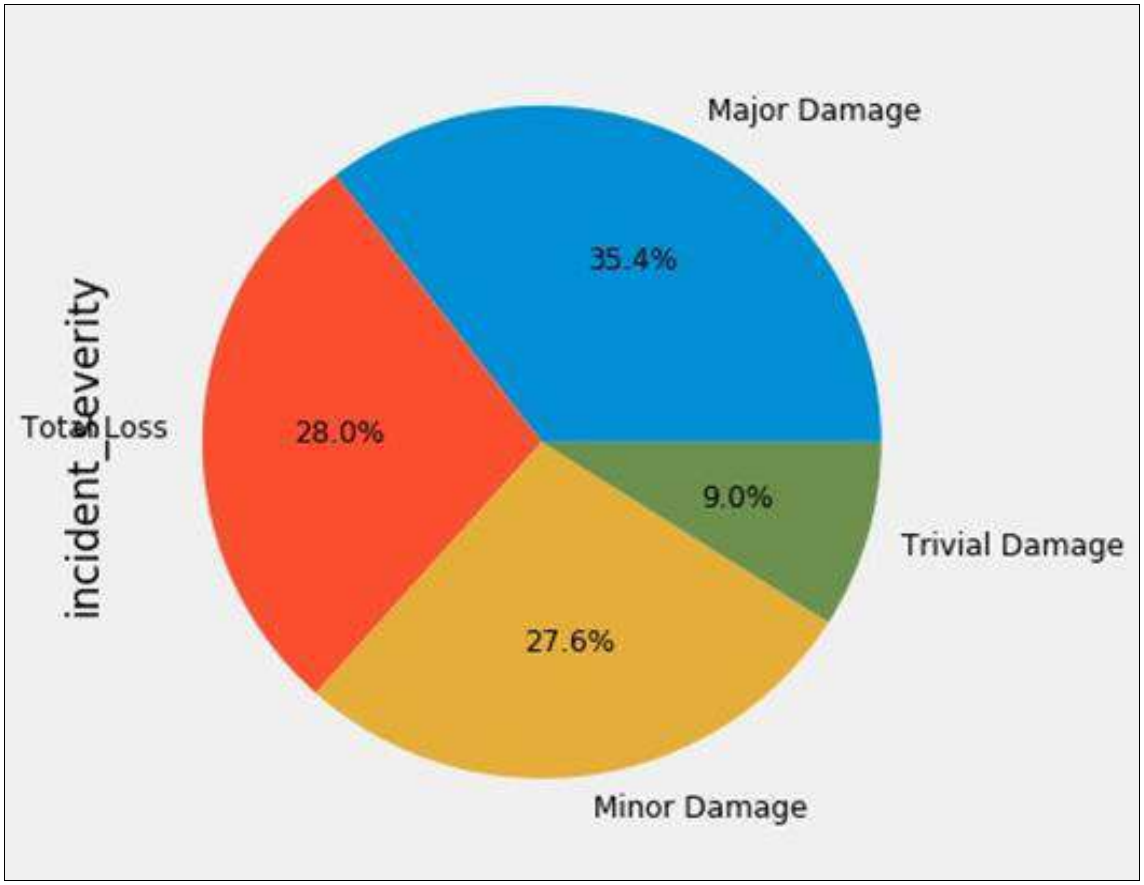

Fig 13: Distribution of data according to severity of incident

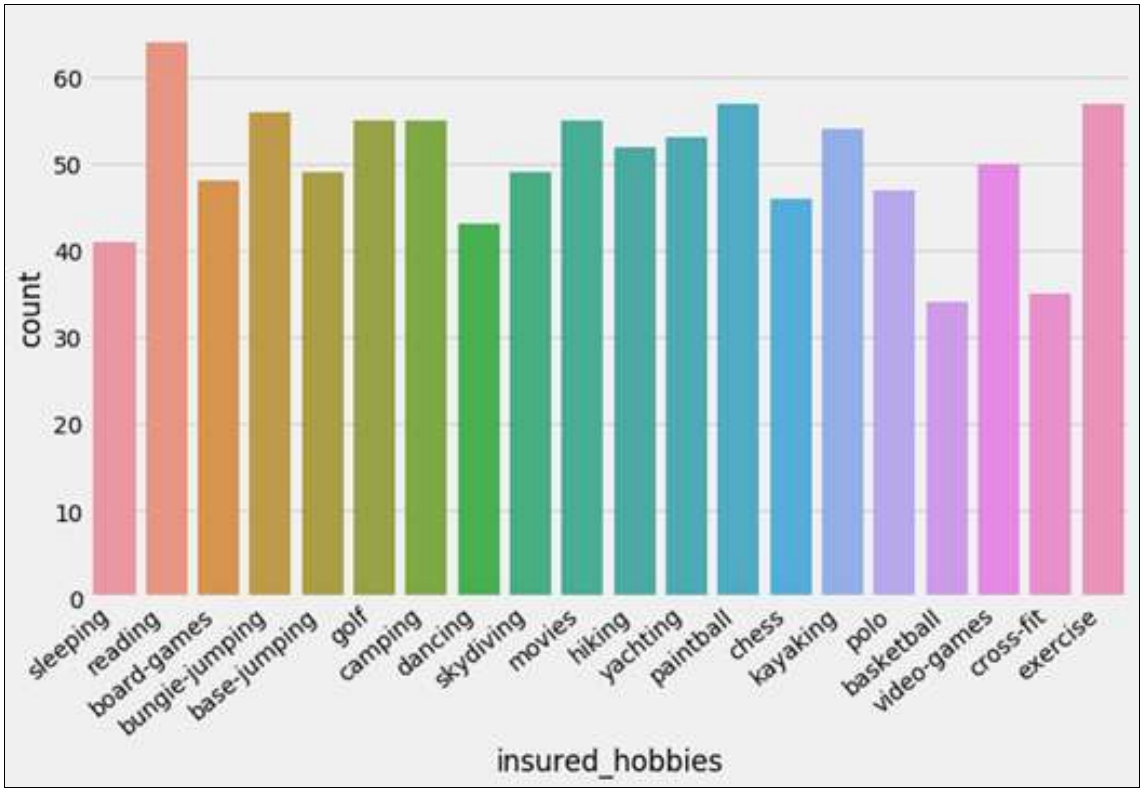

Fig 14: Distribution of data according to hobbies of insured person

From cleaning the dataset, we found that 'policy_bind_date', 'incident_date', 'incident_location' and 'insured_zip' columns were unnecessary, so they were dropped. However, 'auto_year' column had unnecessary values as well, but it is needed for analysis as it can tell the age of the car. So we performed feature engineering on that column to obtain the values i.e. finding the age of the vehicle of each entry, creating a column vehicle_age to store those values, then dropping the auto_year column with the other unnecessary ones.

Next, we found that collision_type, property_damage and police_report_available columns had many missing values. We need these columns for our analysis, so we analysed these columns to check the spread of the categorical variables.

Next, we performed both Label and One-Hot Encoding on the columns as required by specific columns according to their datatypes and type of variable. This made the dataset pliable to be used in a sensitive method such as LDA. We found that even without standardization, the data is 84.1% accurate, so we can use it for the RF classifier model. Feature Scaling is not required as RF classifier is a tree based model as convergence and numeric based issues that often make linear and logistic regression as well as neural networks unreliable, are not relevant issues here, therefore this is a good classifier to show initial accuracy of the models.

On performing analysis, the RF classifier's Confusion Matrix shows us that it has 28.5% error rate i.e. it misclassified 28.5% of the cases (Fig 15).

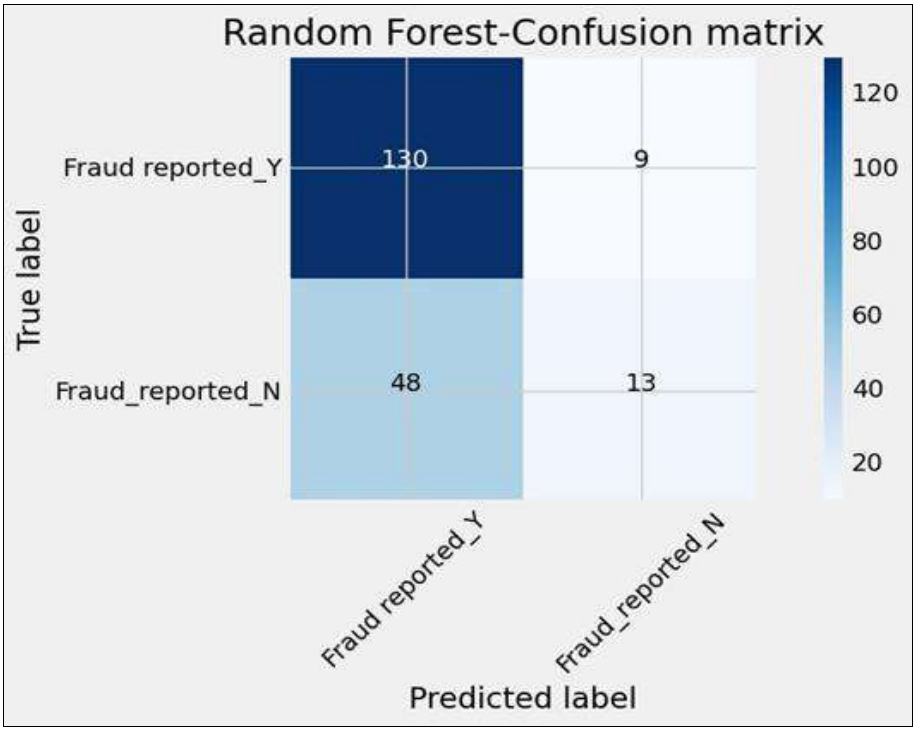

Fig 15: Random Forest Classifier Confusion Matrix

Therefore, this instance of RF classifier has 71.5% accuracy, which is good but not good enough. So we imported the other classifiers and performed analysis based on those models as well, then visualized the results. (Fig 16).

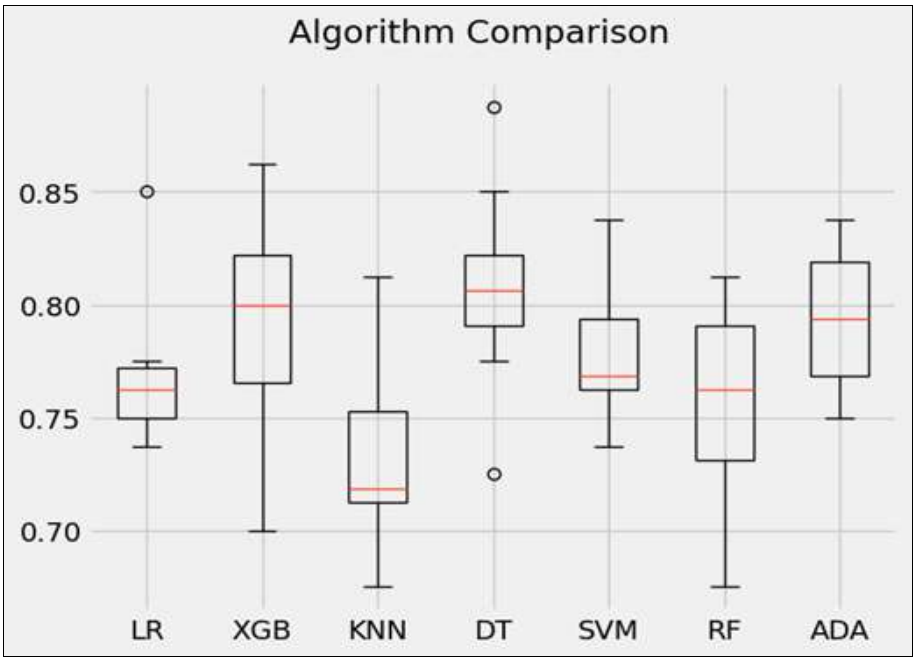

Fig 16: Comparison of Algorithms

Visualization shows us that DT stood at the highest mean accuracy, at 80.75%, followed closely by Adaboost algorithm at 79.37%, with RF classifier being the second worst at 75.75% and KNN algorithm being the worst at 73% accuracy.

To further test the models, we tested combinations of various models on the same dataset to see the various performance metrics. Of those models, the best combination was a combination of Logistic Regression and Decision Tree, achieved through applying Logistic Regression on Decision Tree nodes generated by the code. Doing this generated a model with an accuracy of 72%. (Fig 17, 18).

Misclassified samples: 69

	precision	recall	f1-score	supp
0	0.73	0.99	0.84	
1	0.00	0.00	0.00	
accuracy			0.72	
macro avg	0.36	0.49	0.42	
weighted avg	0.53	0.72	0.61	

Fig 17: Performance Metrics of Logistic Regression-Decision Tree Algorithm

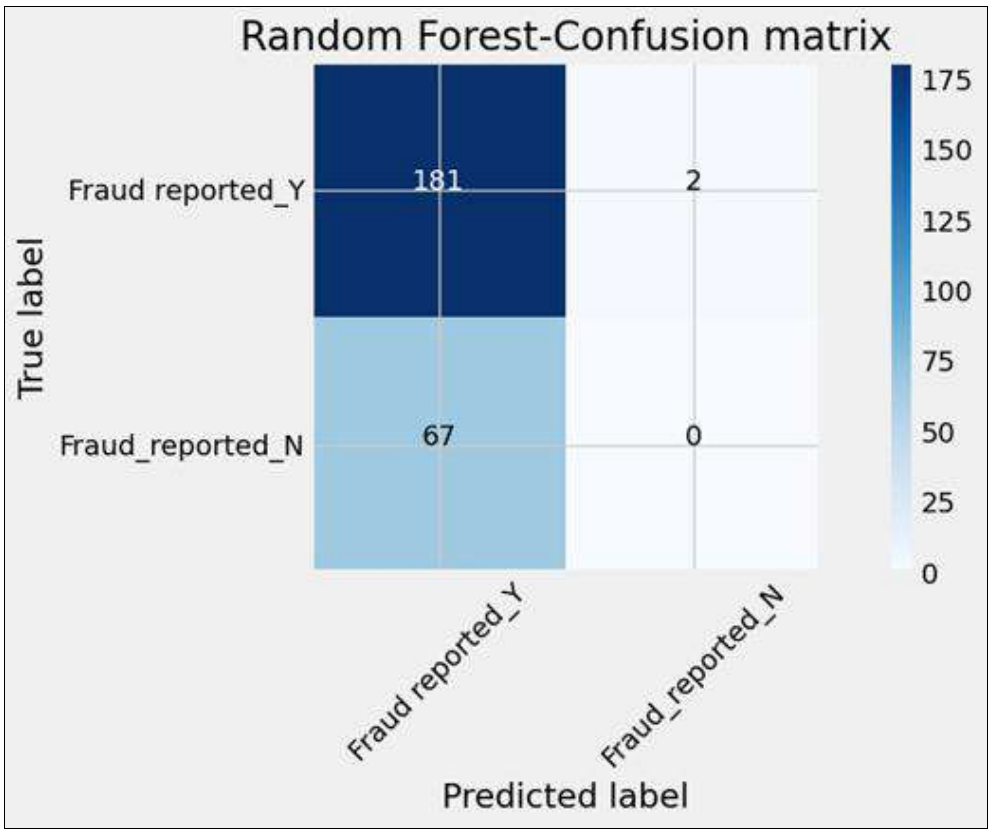

Fig 18: Confusion Matrix for the combined Logistic Regression-Decision Tree Model

In addition, we implemented an algorithm that takes the top three models i.e. DT classifier, Adaboost classifier and XG Boost classifier, then takes the majority results from all three of those classifiers and compares that combined result to the dataset. This algorithm gives an accuracy of 78.4% - already much better than the combination Logistic Regression and Decision Tree model. (Fig 19).

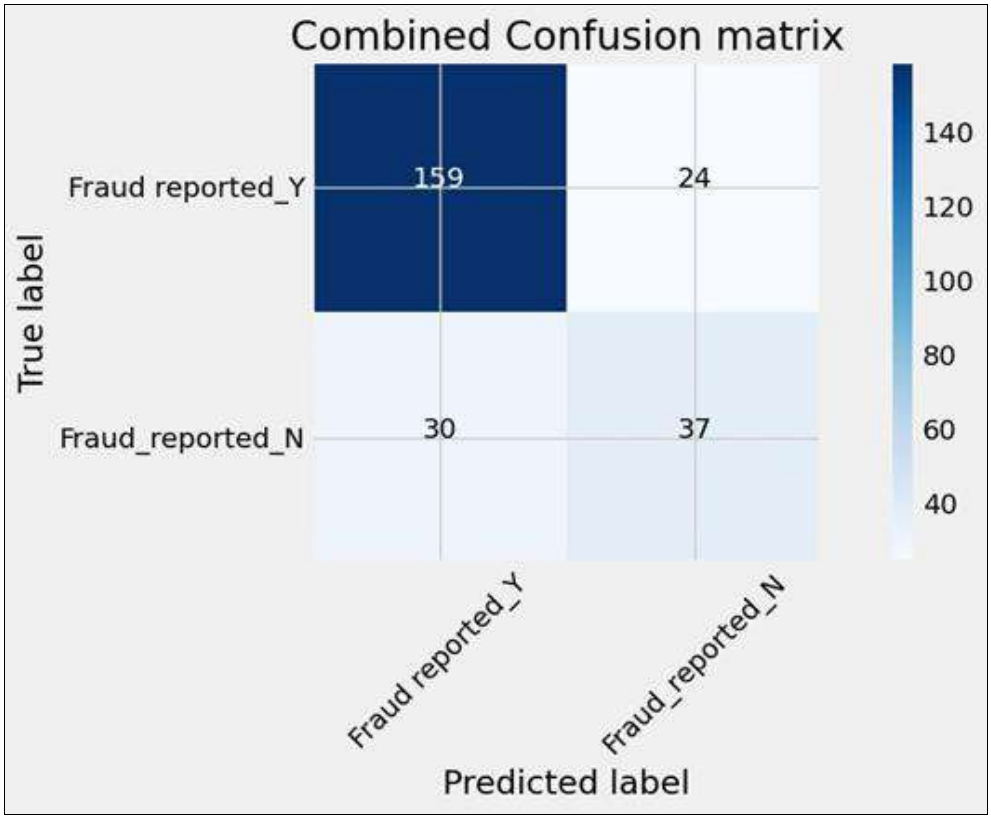

Fig 19: Confusion Matrix for the combined majority system

Therefore we can conclude that in such a situation where we require the identification of fraudulent insurance cases, an algorithm supported by Decision Trees is the best algorithm that can be used.

In the future, this project can be further refined by researching better methods for analysis of this type, and training the model more and more accurately to get the accuracy of the model as close to 100% as possible. However, realistically, 100% accuracy of the model is not possible. Therefore, we will get as close to 100% as possible.

References

1. Fraud Detection and Analysis for Insurance Claim using Machine Learning; c2022 Mar.
2. Fraud Claims Detection in Insurance Using Machine Learning; c2022.
3. Insurance Fraud Detection using Machine Learning; c2021.
4. Insurance Fraud Identification using Computer Vision and IoT: A Study of Field Fires; c2020.
5. Insurance Claim Analysis Using Machine Learning Algorithms; c2019.
6. Extreme Gradient Boosting Machine Learning Algorithm for Safe Auto Insurance Operation; c2019.
7. An XGBoost Based System for Financial Fraud Detection; c2020.
8. Rathore R. A Study on Application of Stochastic Queuing Models for Control of Congestion and Crowding. International Journal for Global Academic & Scientific Research. 2022;1(1):1-6. https://doi.org/10.55938/ijgasr.v1i1.6
9. Sharma V. A Study on Data Scaling Methods for Machine Learning. International Journal for Global Academic & Scientific Research. 2022;1(1):23-33. https://doi.org/10.55938/ijgasr.v1i1.4
10. Rathore R. A Review on Study of application of queueing models in Hospital sector. International Journal for Global Academic & Scientific Research. 2022;1(2):1-6. https://doi.org/10.55938/ijgasr.v1i2.11
11. Kaushik P. Role and Application of Artificial Intelligence in Business Analytics: A Critical Evaluation. International Journal for Global Academic & Scientific Research. 2022;1(3):01-11. https://doi.org/10.55938/ijgasr.v1i3.15
12. Kaushik P. Deep Learning and Machine Learning to Diagnose Melanoma; International Journal of Research in Science and Technology. 2023 Jan-Mar;13(1):58-72. DOI: http://doi.org/10.37648/ijrst.v13i01.008
13. Kaushik P. Enhanced Cloud Car Parking System Using ML and Advanced Neural Network; International Journal of Research in Science and Technology. 2023 Jan-Mar;13(1):73-86. DOI: http://doi.org/10.37648/ijrst.v13i01.009
14. Kaushik P. Artificial Intelligence Accelerated Transformation in the Healthcare Industry. Amity Journal of Professional Practices. 2023;3:1. https://doi.org/10.55054/ajpp.v3i01.630
15. Kaushik P. Congestion Articulation Control Using Machine Learning Technique. Amity Journal of Professional Practices. 2023;3:1. https://doi.org/10.55054/ajpp.v3i01.631
16. Rathore R. A Study of bed occupancy management in the healthcare system using the M/M/C Queue and Probability. International Journal for Global Academic & Scientific Research. 2023;2(1):01-09 https://doi.org/10.55938/ijgasr.v2i1.36