20BCD7094.docx

by

Submission date: 16-Apr-2023 09:49AM (UTC+0530)
Submission ID: 2065649175
File name: 20BCD7094.docx (1.35M)
Word count: 3168
Character count: 17344

VIT-AP UNIVERSITY

A Report on

Auto Insurance Fraud Detection

Submitted in partial fulfillment of the course

CSE 4029
Advanced-Data Analytics

Under the guidance of

Dr. Sagar Dhanraj Pandey

Submitted by
Rampam Greeshma Geethika – 20BCD7094
Puranam Srinija Pravallika – 20BCD7061

AUTO INSURANCE FRAUD DETECTION

Abstract:

Insurance fraud is a common phenomenon committed against insurance companies.

The insurance industry plays a crucial role in protecting individuals, businesses, and organizations from financial dangers. Additionally, several insurance companies support the expansion of various sectors. However, insurance fraud has emerged as a significant threat to the survival of the insurance industry. Insurers face difficulty in identifying fraudulent claims, which can range from staged accidents to exaggerated injuries or damages. Manual investigations and rules-based systems are the traditional methods of fraud detection. These methods take a long time and rarely catch sophisticated fraudsters.

In this project, we will deal with Auto Insurance. This kind of insurance is connected to the world economy and many citizens' daily life. As a result, a lot of insurance companies are trying to cut costs while maintaining their edge over competitors. In addition, insurance fraud accounts for a significant portion of insurance companies' expenses due to its long-term impact on pricing strategies and lower insurance company profits. We talk about how important it is to have high-quality data and how techniques like data cleaning and augmentation can make machine learning models more accurate. The need for interpretability and transparency in model predictions, the risk of bias in training data, and the possibility of adversarial attacks are just a few of the challenges and limitations of using machine learning to detect auto insurance fraud.

We train the dataset with a few machine learning algorithms, such as SVM, KNN, Decision Tree, and Random Forest, to determine the fraud's accuracy. This project's hybrid dataset is a combination of a Kaggle dataset and a personal dataset. Gradient Boost classifier, LightGBM classifier, and catBoost Classifier are utilized to improve accuracy.

Introduction:

Auto insurance fraud is a serious issue that costs industries millions of dollars annually. Due to the complexity of identifying fraudulent claims and the large amounts of data involved, detecting and preventing fraud is difficult. Insurers can now quickly and precisely identify fraudulent claims thanks to machine learning, a promising solution for auto insurance fraud detection.

An overview of machine learning-based strategies for identifying insurance fraud in auto insurance is provided in this abstract. We examine the different stages engaged in fostering an AI model for misrepresentation recognition, including information pre-processing, model determination, and assessment. Moreover, we feature the significance of information quality and make sense of how information cleaning and information expansion methods can work on the exactness of Machine Learning models.

The need for interpretability and transparency in model predictions, the possibility of bias in training data, and the possibility of adversarial attacks are just a few of the issues and limitations we discuss in our final section regarding the application of machine learning to the detection of auto insurance fraud. In the end, we come to the conclusion that methods based on machine learning have the potential to significantly enhance auto insurance fraud detection, aid insurers in reducing losses, and enhance customer trust.

Contribution:

- This paper measures the accuracy and precision of predicting fraud in Insurance companies through valid Machine learning methods.
- Classification techniques to classify the data and reduce the variance, such that the outliers are minimized.
- Boosting methodologies/algorithms like CAT Boost and ADA boost are used to quantitatively reduce the errors in Data we have as data after sampling is bound to be prone to outliers and high and extreme values.
- Usual methods of Random Forest for a better accurate prediction.

Literature review:

- **Nian, K., Zhang, H., Tayal, A., Coleman, T., & Li, Y. (2016). Auto insurance fraud detection using unsupervised spectral ranking for an anomaly.** ***The Journal of Finance and Data Science***
- **Wang, Y., & Xu, W. (2018). Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud.** ***Decision Support Systems***
- **Subudhi, S., & Panigrahi, S. (2020). Use of optimized Fuzzy C-Means clustering and supervised classifiers for automobile insurance fraud detection.**
- **Ngai, E., Hu, Y., Wong, Y., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of the literature.** ***Decision Support Systems***

PROPOSED METHODOLOGIES:

This is a hybrid dataset, with one of the halves from Kaggle and the others from small insurance agencies that gave us information about the same.

Data set snippets:

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	...	police_report_availabl
0	328	48	521585	17-10-2014	OH	250/500	1000	1406.91	0	466732	...	YE
1	52	36	526784	12-06-2012	OH	500/1000	1500	100.32	0	548793	...	YE
2	63	52	9546123	14-02-2010	IN	100/300	1200	520.40	500000	444952	...	YE
3	74	32	632178	23-06-2010	IL	100/300	1700	654.30	600000	479321	...	YE
4	37	44	954612	30-09-2012	IN	250/500	5000	889.30	0	258963	...	N

Fig(i)

	months_as_customer	age	policy_number	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	capital-gains	capital-loss	incident_hour_of_the_day	nu
count	1102.000000	1102.000000	1.102000e+03	1102.000000	1102.000000	1.102000e+03	1102.000000	1102.000000	1102.000000	1102.000000	
mean	202.401089	39.127042	5.572955e+05	1138.656987	1250.233185	1.108757e+06	500185.058076	25521.284936	-27177.613430	11.730490	
std	115.382965	9.211128	3.740603e+05	636.644539	257.858375	2.290792e+06	76023.814759	27989.965793	29708.453508	7.058916	
min	0.000000	19.000000	1.008040e+05	500.000000	100.320000	-1.000000e+06	135131.000000	0.000000	-347897.000000	0.000000	
25%	112.000000	32.000000	3.367500e+05	500.000000	1083.167500	0.000000e+00	447460.750000	0.000000	-51500.000000	6.000000	
50%	198.500000	38.500000	5.387105e+05	1000.000000	1257.020000	0.000000e+00	466346.500000	800.000000	-24300.000000	12.000000	
75%	275.000000	45.000000	7.697805e+05	2000.000000	1414.340000	0.000000e+00	603074.750000	51100.000000	0.000000	18.000000	
max	523.000000	65.000000	9.546123e+06	5400.000000	2047.590000	1.000000e+07	932184.000000	100500.000000	0.000000	42.000000	

Fig(ii)

Number of Instances: 1102
Number of Attributes: 40

Data Visualisation:

1. **Basic Visualization –**

Bar Plots - A graph type known as a bar chart uses rectangular bars to display data. Each bar's height or length reflects the value of the data it conveys. Since they are simple to interpret and use to compare various data points, bar charts are frequently employed in data visualization. For showing categorical data and spotting trends or patterns in the data, they are especially helpful.

Histograms - A histogram is a graphical representation of the distribution of a dataset. It displays the frequency of occurrences of values within a given range or bin.
In data visualization, histograms are important for quickly identifying patterns and trends in data and providing insights into the shape of the distribution.
They can be used to identify outliers and understand the spread and central tendency of a dataset.

Scatterplots - A scatterplot is a graphical representation of the relationship between two variables. It is an important tool for visualizing and analyzing the relationship between the dependent variable and the independent variable.
By using a scatterplot, researchers can quickly identify patterns, trends, and outliers in their data.

Line charts - An example of a graph is a line chart, which shows data as a series of points connected by a line. It is frequently used to demonstrate long-term trends in things like stock prices and climatic patterns. Line charts are crucial for data visualization because they can clearly express data changes and patterns, allowing users to quickly grasp trends and make defensible conclusions.

Pie Chart - Pie charts are commonly used to show data that is divided into a few distinct categories, such as market share, budget allocation, or survey responses with multiple-choice options. They can provide a visual representation of the relative proportions of different categories, making it easy to understand the distribution of data at a glance. However, pie charts may not be suitable for displaying data with too many categories or components, as it can result in a cluttered and hard-to-read chart. In such cases, other types of charts, such as bar charts or stacked bar charts, may be more appropriate.

2. **Decision Tree Classifier:** Firstly we have separated the data into training and testing sets to prepare it. We have utilized a target variable and a collection of input features, training the Decision Tree classifier on the training set. Further assessing the trained model's accuracy by contrasting the values it predicted with those found in the testing set and adjusting the model's parameters as appropriate to boost performance. Based on the values of the input features, the algorithm repeatedly divides the data into smaller and smaller subsets until each subset only contains examples of a single class. By doing this, it develops a structure like a tree that may be used to generate predictions about fresh data. The model's hyperparameters, such as the maximum depth of the tree, the bare minimum of samples needed to split a node, and the criterion used to assess the quality of a split, can be changed to increase the model's accuracy.

3. **SVM algorithm** is used, this is a type of algorithm that is a non-parametric clustering algorithm that does not assume the number or shape of clusters in data. (Support vector classifier). Basically gives out the best-fit hyperplane for our data.
   Prior to using SVM to forecast accuracy, we first prepare the data by choosing pertinent features and preprocessing it to guarantee that it is in a format that the algorithm can understand. The

SVM model was then trained using the training set, and its performance was assessed using the test set, which was divided up into a training set and a test set.
Here in our model, SVM uses a kernel function to transform the data into a higher-dimensional space if necessary in order to learn a decision boundary that divides the high-accuracy and low-accuracy classes. The SVM method seeks to identify the decision boundary that minimizes classification error while maximizing the margin between the two classes. We can use our now-ready SVM model to make predictions on new data once it has been trained by putting the input through the model and looking at the projected class label. We have compared the predicted labels to the actual labels in the test set and derived several performance metrics, including accuracy, precision, recall, and F1 score, to assess the SVM model's performance.

4. **Random Forest Classifier** fits in the number of decision tree parameters at train time (Accuracy prediction)
   The Random Forest classifier initially divides the data into a training set and a testing set in order to forecast accuracy. The Random Forest model is constructed using the training set, and its performance is assessed using the testing set.
   A huge number of decision trees are constructed by the Random Forest algorithm during training, and each one is trained using a random fraction of the features and data examples in the training set. After that, the algorithm integrates all of the trees' predictions to produce a final forecast for each instance in the testing set.
   By comparing the predicted labels to the actual labels in the testing set, the accuracy of the predictions is determined to assess the Random Forest model's performance. This accuracy score can then be used to judge how well the model predicts brand-new, unobserved data.

5. **KNN** The dataset is divided into training and testing sets in order to employ KNN for accurate prediction. The accuracy of the model is then assessed on the testing set after the KNN algorithm has been trained on the training set. This is commonly done by calculating the percentage of correctly categorized occurrences or, in the case of regression tasks, the mean squared error. An essential KNN parameter is K's value. A higher value of K will produce a smoother decision border and may mitigate the effects of noise or outliers, but it may also impair accuracy. A lower number of K will provide a decision boundary that is more complex and may overfit the training set, but it may also produce better accuracy.

Boosting Algorithms:

Light GBM

Building decision trees in sequence while attempting to raise each tree's performance is the boosting approach used in Light GBM. The algorithm begins by building a straightforward decision tree, which is usually a shallow tree with a limited number of levels. The next tree is then constructed by concentrating on the occurrences that this tree incorrectly categorized. To attempt to correctly classify the cases that the prior tree incorrectly categorized, the following tree is constructed. Until a predetermined stopping criterion is met, such as a maximum number of trees or a minimal gain in performance, this procedure is repeated.

Gradient Boosting Algorithm:

The gradient boosting approach reduces the model's error over time by fitting a decision tree to the residuals of the prior tree. Until the desired accuracy is attained, this process is repeated a set number of times. The approach calculates the gradient of the loss function with respect to the previous tree's output in each iteration and utilizes this gradient to fit a new tree that accounts for the model's remaining mistake.
The fact that gradient boosting may be used to solve both regression and classification issues and can be applied to a variety of datasets is one of its main benefits.
Additionally, gradient boosting is known for its high accuracy and ability to avoid overfitting, which is a common problem in machine learning.

CAT Boosting Algorithm:

CatBoost builds decision trees from the input features and iteratively adds new trees to the model using a gradient-boosting technique. The algorithm recognizes the instances that the previous trees misclassified in each iteration and adds new trees that concentrate on these occurrences. Until a preset stopping criterion is satisfied (for example, the maximum number of iterations has been reached or the improvement in the model's performance is minimal), this process is repeated.
CatBoost's capability to handle categorical data directly, without the requirement for one-hot encoding or other preprocessing procedures, is one of its important strengths. This is accomplished using a method known as "ordered boosting," which computes the gradients for categorical variables using a permutation-based method.
This makes it possible for CatBoost to manage high-dimensional category features and record their interactions.

ADA Boosting Algorithm:

AdaBoost's strength is its ability to turn weak learners into strong learners, improving the performance of weak learners. Additionally, it can manage unbalanced datasets by giving the minority class higher weights. AdaBoost can be computationally expensive and susceptible to noise and outliers, though. In order to maximize performance, it is crucial to properly tweak the algorithm's hyperparameters.
It initializes the weight for each instance to be converting them over a range of equal weights. Then, it has the ability to work on a weak learner, which is basically a decision tree that is very simple and transit. After it is learned, the weights of the misclassified instances are increased, so the next weak learner focuses on the hardest to classify examples. Then the weights of the correctly classified instances are decreased. The final model is created by combining all of the weak learners, where the weights of each weak learner are determined by its accuracy.

XG Boost Algorithm

The approach is often used iteratively, with each iteration adding a new decision tree to the model, to use XGBoost in the boosting phase. A new decision tree is trained to concentrate on these examples after each iteration, and the weights of the misclassified instances are modified. The final boosted model is then produced by combining the results of all the decision trees.

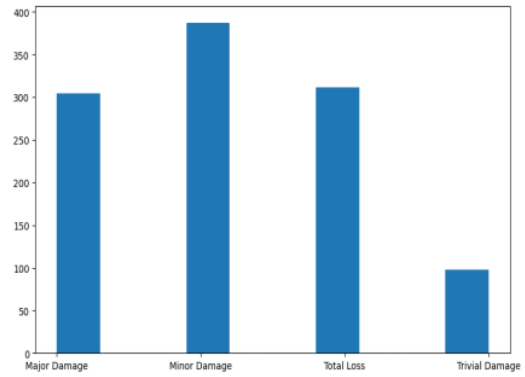
The regularisation term that is part of XGBoost's design helps to prevent overfitting, which is one of its main advantages. Complex models are penalised by this term, which aids in model simplification and prevents overfitting of the training set.

Proposed Methodology:

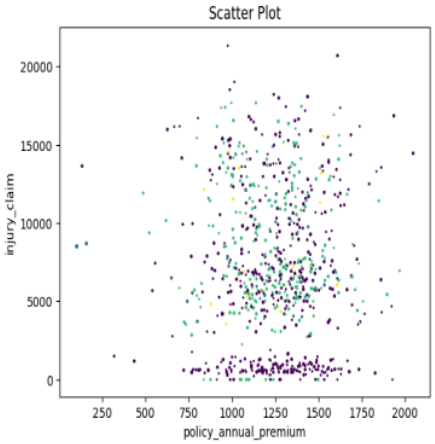
The proposed methodologies are a hybrid of ML and models of SVM. The dataset was first visualized and taken note of, the outliers and the relationship between variables of first-line consideration. Then, categorized methods were added as a boost to prevent errors in the predictive analysis.

Data Visualisation

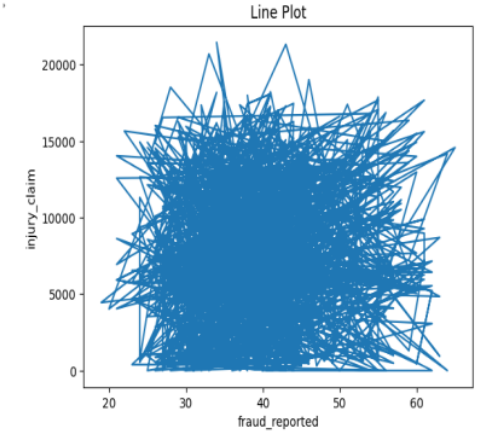
The variables are first grouped by the most effecting variable that contributes to the most fraud reported. Then we make the following visualizations to consider the relations and understand them more in a way explainable.



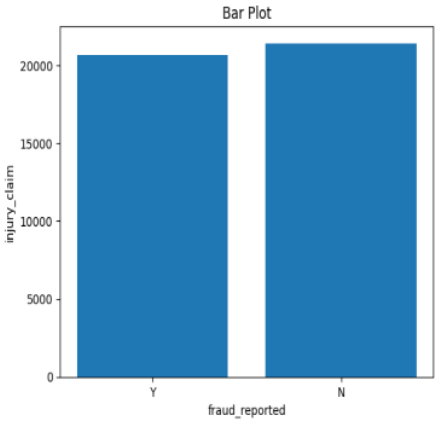
Fig(iii)



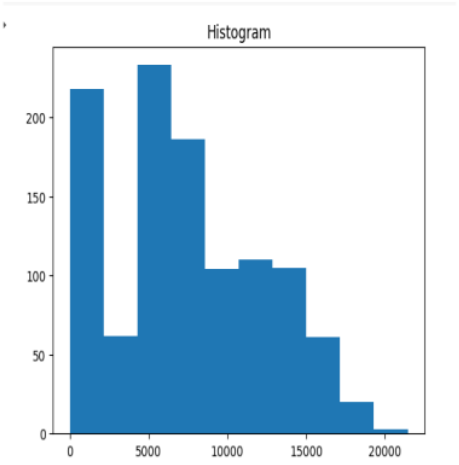
Fig(iv)



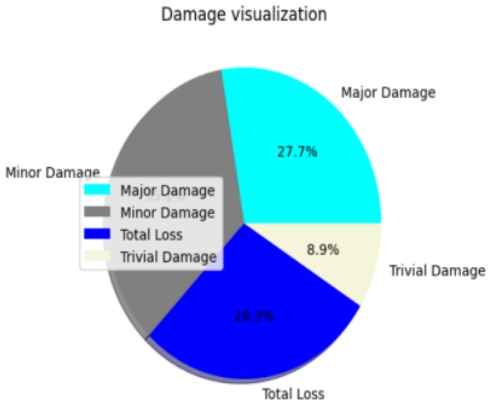
Fig(v)



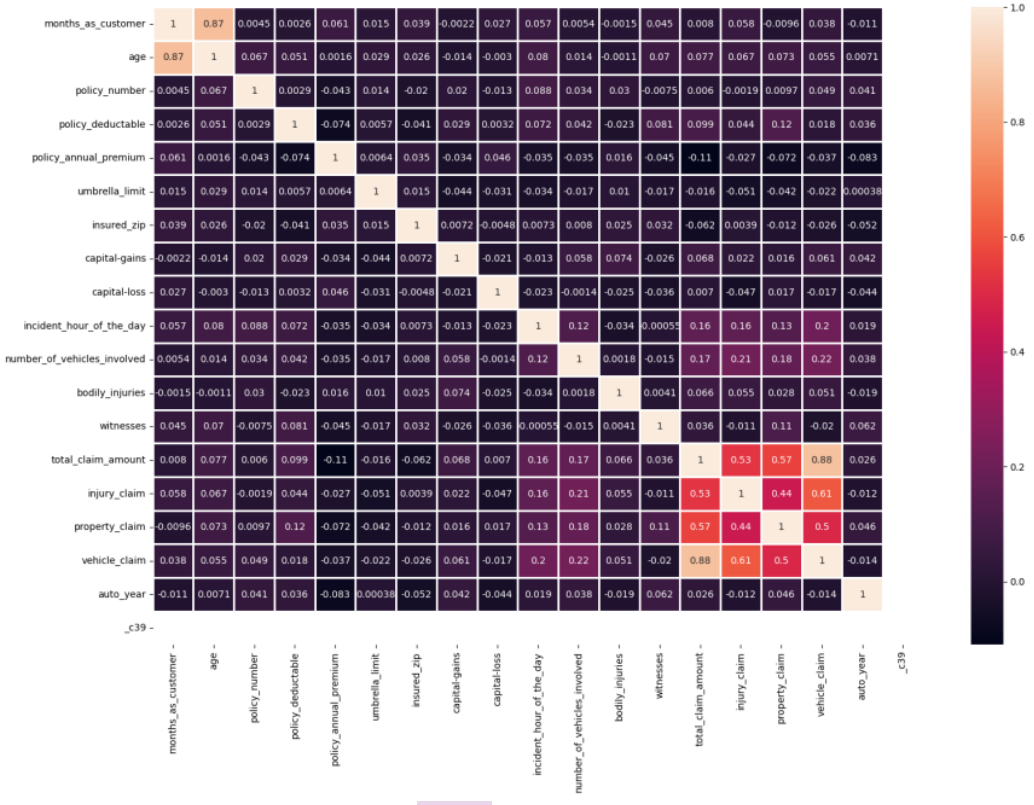
Fig(vi)



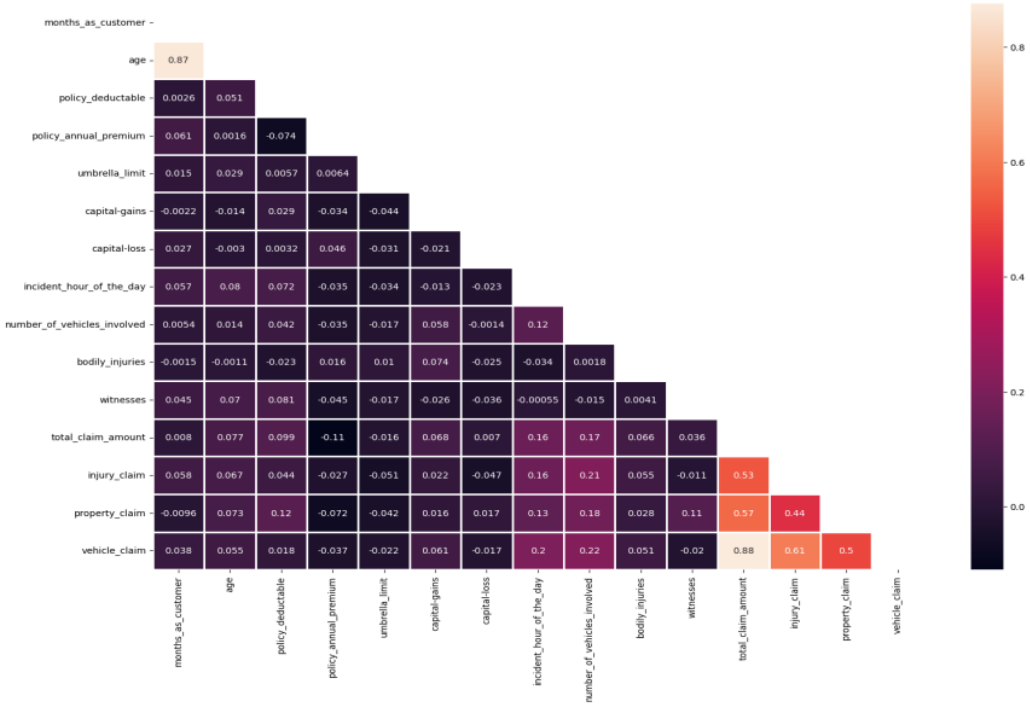
Fig(vii)



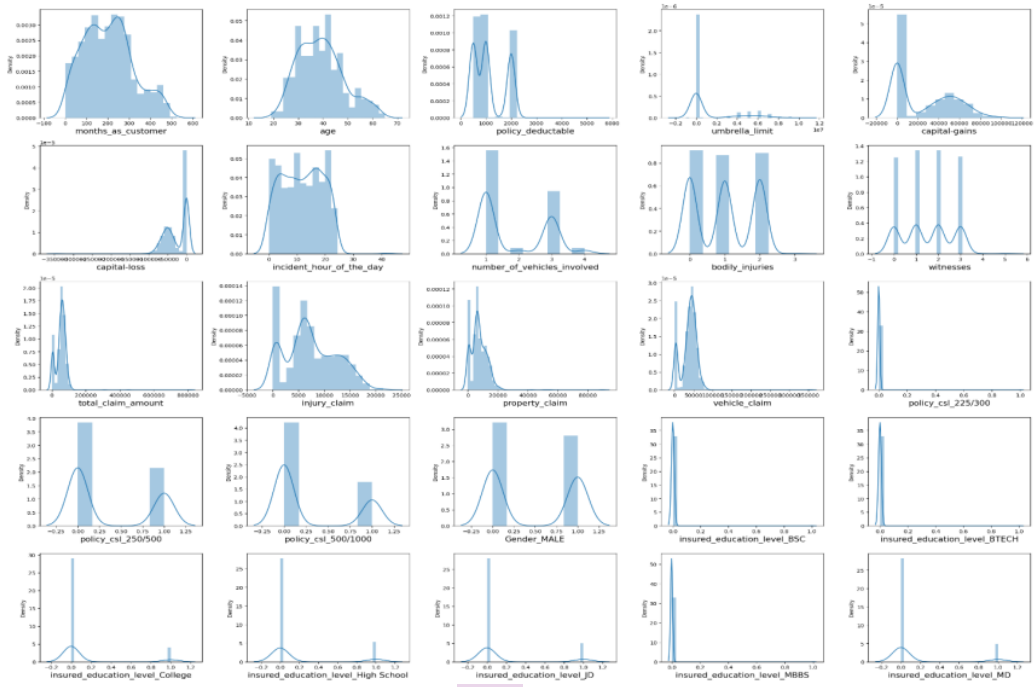
Fig(viii)



Fig(ix)


months_as_customer
age
policy_deductable
policy_annual_premium
umbrella_limit
capital-gains
capital-loss
incident_hour_of_the_day
number_of_vehicles_involved
bodily_injuries
witnesses
total_claim_amount
injury_claim
property_claim
vehicle_claim
0.8
0.6
0.4
0.2
0.0

Fig(x)


months_as_customer
age
policy_deductable
umbrella_limit
capital-gains
capital-loss
incident_hour_of_the_day
number_of_vehicles_involved
bodily_injuries
witnesses
total_claim_amount
injury_claim
property_claim
vehicle_claim
policy_csl_225/300
policy_csl_250/500
policy_csl_500/1000
Gender_MALE
insured_education_level_BSC
insured_education_level_BTECH
insured_education_level_College
insured_education_level_High School
insured_education_level_JD
insured_education_level_MBBS
insured_education_level_MD

Fig(xi)

Removing Outliers and Balancing Data:

Outliers are data points in a dataset that is significantly beyond the expected range of values. They may be the result of mistakes in measurement or data collection, or they may represent atypical or extreme findings that are not typical of the remainder of the data. As they can distort data and alter the model's overall performance, outliers can significantly affect statistical studies and machine learning models. In order to lessen the influence of these extreme results and increase the model's accuracy, outliers should be removed from the data.

Making sure that there are about equal numbers of occurrences of the target variable in each class or category is known as balancing the data. This is significant since a lot of machine learning algorithms are made to function best with balanced data. The model may get biassed in favour of the majority class when the data are unbalanced, which could negatively affect the performance of the minority classes. In order to ensure that the model is trained on a representative sample of the data and that all classes are given the same weight during the training phase, it can be helpful to balance the data.

SVM classification:

```
Training accuracy of SVC : 0.761501210653753
Test accuracy of SVC : 0.7137681159420289
[[197   0]
 [ 79   0]]
              precision    recall  f1-score   support

           0       0.71      1.00      0.83       197
           1       0.00      0.00      0.00        79

    accuracy                           0.71       276
   macro avg       0.36      0.50      0.42       276
weighted avg       0.51      0.71      0.59       276
```

Fig(xii)

KNN:

```
Training accuracy of KNN is : 0.761501210653753
Test accuracy of KNN is : 0.7137681159420289
[[197   0]
 [ 79   0]]
              precision    recall  f1-score   support

           0       0.71      1.00      0.83       197
           1       0.00      0.00      0.00        79

    accuracy                           0.71       276
   macro avg       0.36      0.50      0.42       276
weighted avg       0.51      0.71      0.59       276
```

Fig(xiii)

Decision Tree Classifier:

```
Training accuracy of Decision Tree is : 0.8583535108958837
Test accuracy of Decision Tree is : 0.822463768115942
[[163  34]
 [ 15  64]]
              precision    recall  f1-score   support

           0       0.92      0.83      0.87       197
           1       0.65      0.81      0.72        79

    accuracy                           0.82       276
   macro avg       0.78      0.82      0.80       276
weighted avg       0.84      0.82      0.83       276
```

Fig(xiv)

Random Forest classifier:

```
Training accuracy of Random Forest is : 0.9346246973365617
Test accuracy of Random Forest is : 0.7463768115942029
[[196   1]
 [ 69  10]]
              precision    recall  f1-score   support

           0       0.74      0.99      0.85       197
           1       0.91      0.13      0.22        79

    accuracy                           0.75       276
   macro avg       0.82      0.56      0.54       276
weighted avg       0.79      0.75      0.67       276
```

Fig(xv)

Boosting Models:

Light gbm

```
Training accuracy of Light gbm is : 1.0
Test accuracy of Light gbm is : 0.8333333333333334
[[179  18]
 [ 28  51]]
              precision    recall  f1-score   support

           0       0.86      0.91      0.89       197
           1       0.74      0.65      0.69        79

    accuracy                           0.83       276
   macro avg       0.80      0.78      0.79       276
weighted avg       0.83      0.83      0.83       276
```

Fig(xvi)

Gradient Boosting Algorithm:

```
Training accuracy of Random Forest is : 0.9443099273607748
Test accuracy of Random Forest is : 0.8115942028985508
[[177  20]
 [ 32  47]]
              precision    recall  f1-score   support

           0       0.85      0.90      0.87       197
           1       0.70      0.59      0.64        79

    accuracy                           0.81       276
   macro avg       0.77      0.75      0.76       276
weighted avg       0.81      0.81      0.81       276
```

Fig(xvii)

CAT Boosting Algorithm:

```
Training accuracy of Catboost is : 0.9842615012106537
Test accuracy of Catboost is : 0.8043478260869565
[[178  19]
 [ 35  44]]
              precision    recall  f1-score   support

           0       0.84      0.90      0.87       197
           1       0.70      0.56      0.62        79

    accuracy                           0.80       276
   macro avg       0.77      0.73      0.74       276
weighted avg       0.80      0.80      0.80       276
```

Fig(xviii)

ADA Boosting Algorithm:

```
Training accuracy of Random Forest is : 0.8583535108958837
Test accuracy of Random Forest is : 0.8260869565217391
[[163  34]
 [ 14  65]]
              precision    recall  f1-score   support

           0       0.92      0.83      0.87       197
           1       0.66      0.82      0.73        79

    accuracy                           0.83       276
   macro avg       0.79      0.83      0.80       276
weighted avg       0.85      0.83      0.83       276
```

Fig(xix)

XG Boost Algorithm

```
Training accuracy of XGB is : 1.0
Test accuracy of XGB is : 0.822463768115942
[[181  16]
 [ 33  46]]
              precision    recall  f1-score   support

           0       0.85      0.92      0.88       197
           1       0.74      0.58      0.65        79

    accuracy                           0.82       276
   macro avg       0.79      0.75      0.77       276
weighted avg       0.82      0.82      0.82       276
```

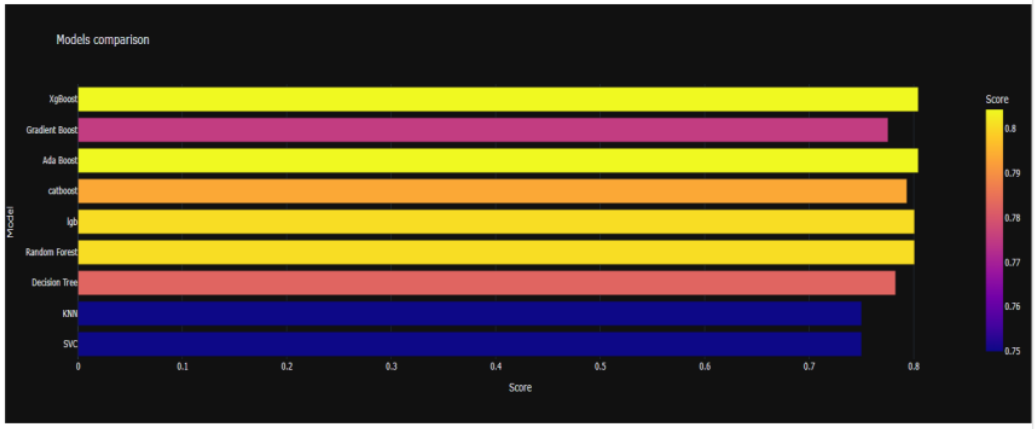
Fig(xx)

Results:

Comparing Accuracy for the Ultimate prediction model

Model	Accuracy
SVM	0.750
KNN	0.750
Decision Tree	0.782
Random Forest	0.800
Light GBM	0.800
Gradient Boosting	0.775
Cat Boosting	0.804
Ada Boosting	0.804
Xg Boosting	0.804

	Model	Score
6	Ada Boost	0.804348
8	XgBoost	0.804348
3	Random Forest	0.800725
4	lgb	0.800725
5	catboost	0.793478
2	Decision Tree	0.782609
7	Gradient Boost	0.775362
0	SVC	0.750000
1	KNN	0.750000



Future Scope:

Fraud is a major source of concern for organisations and financial institutions, resulting in considerable financial losses and reputational damage. Machine learning (ML) models have showed potential in detecting fraud, such as credit card fraud, insurance fraud, and identity theft. However, there is still potential for progress in this subject, and academics are looking for ways to improve these models' capabilities.

Incorporating more data sources is one potential area for improvement. While existing ML models focus on transactional data and user behaviour patterns to detect fraud, adding new data sources can provide additional insights into user behaviour. Social media activity and location data, for example, can be utilised to confirm user behaviour patterns and flag potential fraudulent activities.
Another area of investigation is the incorporation of more complex algorithms. As machine learning techniques progress, more advanced algorithms such as deep learning and reinforcement learning might be investigated for their potential in identifying fraud. Deep learning algorithms, for example, may automatically learn features from data, allowing them to find complicated patterns that regular ML models may struggle to detect.

Finally, the ethical implications of fraud detection models must be considered. It is critical, as with any ML application, to verify that models are fair and unbiased and that user privacy is safeguarded. Future studies can look into approaches to mitigate these difficulties, such as employing differential privacy or fairness requirements in model training.

Conclusion:

In conclusion, utilizing machine learning models to detect fraud is an efficient technique to tackle fraudulent actions in a variety of businesses. A machine learning model can learn to identify trends and anomalies that suggest fraudulent behavior by being trained on historical data. It is critical to have high-quality data and a clear understanding of the fraud situation at hand when developing an efficient fraud detection system. Furthermore, as new types of fraud occur, it is critical to continuously monitor and upgrade the system.

To summarise, fraud detection using ML models is an important topic with tremendous room for advancement. Researchers can continue to improve the accuracy and effectiveness of these models by incorporating more data sources, improving model explainability, improving real-time detection, introducing unsupervised learning, addressing the class imbalance, exploring more advanced algorithms, and considering ethical implications.

Overall, given the increasing frequency of fraud in various industries, the use of machine learning for fraud detection is becoming increasingly significant and necessary, addressing the one problem statement we focused on.

20BCD7094.docx

ORIGINALITY REPORT

12%	7%	4%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.inderscience.com Internet Source	1%
2	Submitted to Oklahoma City University Student Paper	1%
3	stax.strath.ac.uk Internet Source	1%
4	Rokach, . "Introduction to Ensemble Learning", Series in Machine Perception and Artificial Intelligence, 2009. Publication	1%
5	Submitted to National College of Ireland Student Paper	1%
6	Submitted to University College for the Creative Arts at Canterbury, Epsom, Farnham, Maidstone and Rochester Student Paper	1%
7	www.hindawi.com Internet Source	1%
8	Submitted to European University Student Paper	1%

9	llds.ling-phil.ox.ac.uk Internet Source	1 %
10	Submitted to Nanyang Technological University Student Paper	1 %
11	Submitted to University of Wales Institute, Cardiff Student Paper	1 %
12	Submitted to University of New Haven Student Paper	<1 %
13	Submitted to HELP UNIVERSITY Student Paper	<1 %
14	Submitted to University of Kent at Canterbury Student Paper	<1 %
15	hdl.handle.net Internet Source	<1 %
16	www.irjmets.com Internet Source	<1 %
17	Submitted to Liverpool John Moores University Student Paper	<1 %
18	Bouzgarne Itri, Youssfi Mohamed, Qbadou Mohammed, Bouattane Omar. "Performance comparative study of machine learning algorithms for automobile insurance fraud	<1 %

detection", 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), 2019
Publication

19	Galar, M., A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches", IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews), 2012. Publication	<1 %
20	www.soa.org Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		