Vehicle Insurance Fraud claim detection Using Machine learning Algorithm

Aniket Guru
School of Computing
National College of Ireland
Dublin, Ireland
x21141797@student.ncirl.ie

Abstract

Five distinct models will be applied to three different datasets related to vehicle insurance claims classification using three machine learning algorithms: Linear Regression, Random Forest, and XGBoost [13]. Linear Regression and Random Forest have been used from two different libraries, namely scikit-learn and statsmodel modules, while both scikit-learn and H2O [12] libraries have been used for Random Forest. XGBoost is used from its own library. All models were hypertuned using grid search CV, and the best tuning parameters were selected to produce the best output. All models achieved an accuracy of over 90% on the training dataset. Random forest accuracy surpassed all other methods, especially XG Bosst, with an accuracy of more than 90%, making it an effective tool for identifying fraudulent claims.

Index Terms

RF, LR , CV, XG

I. Introduction

Machine learning is a branch of artificial intelligence (AI) and computer science [1] which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Machine learning is utilized in our day-to-day lives to address problems of all sizes, from the simplest to the most complex. For instance, applications like Google Assistant and self-driving cars leverage machine learning to accomplish their respective tasks.

Insurance provides financial protection in case of unforeseen events, such as accidents, medical emergencies, or property damage.

Vehicle insurance is a mandatory process using which a company accepts to provide an assurance of compensation for mentioned loss or damage to the vehicle in return for a payment of a specified premium. [2] Vehicle insurance fraud is being considered to be one of the major categories of fraud. Filing of false claims against a vehicle insurance policy is termed as vehicle insurance fraud.Over the past few years, there has been a rise in the number of cases where individuals make false claims in vehicle insurance. Making false claims is not only illegal, but it can also lead to the denial of the claim, policy cancellation, or even legal action.

In this project, we have implemented several machine learning techniques on past data where fraud has been found. We computed various evaluation metrics such as the confusion matrix [14], ROC,AUC ,accuracy, and F1 score to compare the performance of each model on each dataset. The goal was to determine which model is the best for identifying fraudulent claims in vehicle insurance.

A. Research question

a)Who are the target customers who are likely going to falsely claim the vehicle insurance?
b) what are the features that are more concerned with detecting that this claim could be fraud ?
c)What new features can we introduce to help identify if a claim is fraudulent?

B. Approach Summary

All three dataset belongs to Vehicle insurance data sourced from kaggel ,based on classification

a) Custom Functions: We have created several custom functions to optimize the code and simplify the process. For example, we have functions to drop a column, check for missing values, identify numerical and categorical columns, and check unique values to better understand the features. These functions help to reduce the complexity of the code and make it more efficient.//
b) Feature engineering: For all three datasets in the Vehicle insurance data sourced from Kaggle, feature encoding was applied to appropriate attributes using one-hot encoding and manual encoding to convert the data into numerical form. Some columns, such as policy numbers or IDs, were dropped during feature engineering because they were not as important as other features. The data was prepared for the next step using a proper approach.

c) Cleaning: For the data cleaning, I used the box plot IQR method to detect outliers wherever necessary. Mostly, I used the trim method by examining the box plot to achieve normally distributed histograms of the dependent feature and reduce skewness. In most cases, I was able to achieve this.

d) Transformation: In data-set 1 and 3, most of the features have a similar range of values, and there are no features with a large number of values. However, in dataset 2, which contains information about fraudulent claims on car physical damage, some columns showed high variability and skewness. To handle this, we used the MinMaxScaler function from the scikit-learn library to transform the data into a uniform scale and reduce the effects of outliers. This makes it easier to analyze the data and build accurate machine learning models.

e) Feature selection: During the machine learning process, feature selection plays a crucial role in determining the accuracy of the algorithm. To ensure the accuracy of the algorithm, we have carried out feature selection by analyzing the correlation between the feature columns and the target variable. The features that had a negligible correlation were dropped during the process. We also took care of multicollinearity using the Variance Inflation Factor (VIF) and multicollinearity plots. The columns causing multicollinearity were dropped to avoid any conflicts.

f) Model building and evaluation: For model building, I utilized two different libraries, namely scikit-learn and stats model, to implement linear classification techniques. I also evaluated the models with and without
hyperparameter tuning, and analyzed the performance using ROC, classification report [14], and confusion matrix.
Furthermore, I employed Random Forest algorithm using both scikit-learn and h2o libraries, and applied hyperparameter tuning with GridSearchCV to obtain optimal results. I evaluated the models using classification report, accuracy, ROC-AUC, and confusion matrix.
Lastly, I compared the performance of XGBoost [3]algorithm on both datasets by running it with hyperparameter tuning.

II. INITIAL LITERATURE REVIEW

Determining the fraud claim or false claim made by the customer is difficult task it is governed by multiple factor like type of accident , how frequent the customer is making claim and what all his past record in claim ,

In this research **R. Roy and K. T. George** focuses [4] on the use of machine learning techniques to detect fraudulent insurance claims. The authors discuss the challenges of detecting fraud in the insurance industry and propose the use of machine learning algorithms as a solution., including logistic regression, decision tree, and k-means clustering. The authors also discuss the importance of data pre-processing and feature selection for improving the accuracy of the models. The proposed framework is evaluated on a real-world insurance dataset and the results demonstrate the effectiveness of machine learning techniques in detecting fraudulent claims. The authors conclude that machine learning techniques can play a crucial role in reducing the financial losses caused by insurance fraud.
This paper **"Predicting Fraudulent Claims in Automobile Insurance"** [5] proposes a machine learning-based approach to detect fraudulent claims in automobile insurance. The authors used a dataset containing 1000 automobile insurance claims with 37 features, including claimant's age, vehicle age, accident type, and others. The authors applied several machine learning algorithms, such as decision tree, random forest, k-nearest neighbors, and support vector machine, and evaluated their performance using metrics such as accuracy, precision, recall, and F1 score. The results show that the random forest algorithm outperformed other algorithms, achieving an accuracy of 96.2% and F1 score of 0.961. The study concludes that the proposed machine learning approach can effectively detect fraudulent claims in automobile insurance.
IN this paper **"Research and application of random forest model in mining automobile insurance fraud,"** [6] describes the use of a random forest model for detecting automobile insurance fraud. The authors conducted experiments on real-world data and found that the random forest model achieved higher accuracy than other traditional machine learning models. They also proposed a feature selection method to identify the most important features for fraud detection. The study demonstrates the potential of using machine learning techniques for improving fraud detection in the insurance industry.
In this research **"Detecting insurance claims fraud using machine learning techniques"** [7]paper discusses the use of machine learning techniques to detect fraudulent insurance claims. The authors collected data from an insurance company with a total of 10,000 claims, out of which 356 were identified as fraudulent. The authors applied various classification algorithms such as Decision Tree, Logistic Regression, and Random Forest to the data and evaluated their performance using accuracy, precision, recall, and F1 score metrics. The Random Forest model outperformed the other models with an accuracy of 99.16%, a precision of 97.12%, a recall of 97.22%, and an F1 score of 97.17%. This study suggests that machine learning techniques, particularly the Random Forest algorithm, can effectively detect fraudulent insurance claims.

All of the above research studies suggest that selecting the most relevant features is crucial for achieving the best performance with machine learning models, particularly random forest and XGBoost.

III. METHODOLOGY

KDD (Knowledge Discovery in Databases) is a process that involves the extraction of useful, previously unknown, [7] and potentially valuable information from large datasets. The KDD process in data mining typically involves the following steps

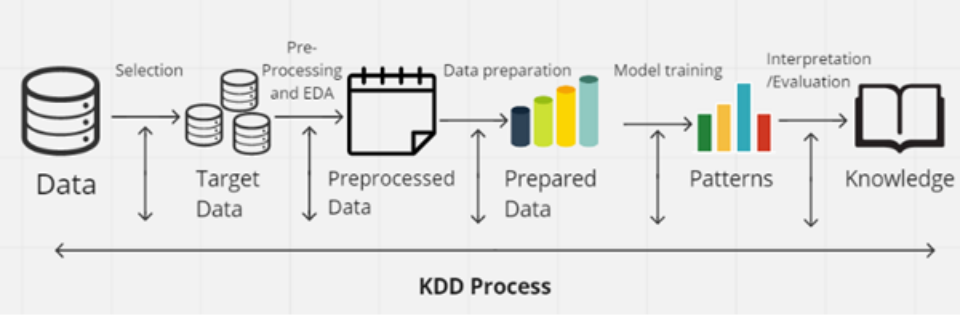


Fig. 1. KDD FLow Chart

Flowchart describes all the steps included in KDD.

A.Vehicle Insurance Claim Fraud Detection (Dataset-1)

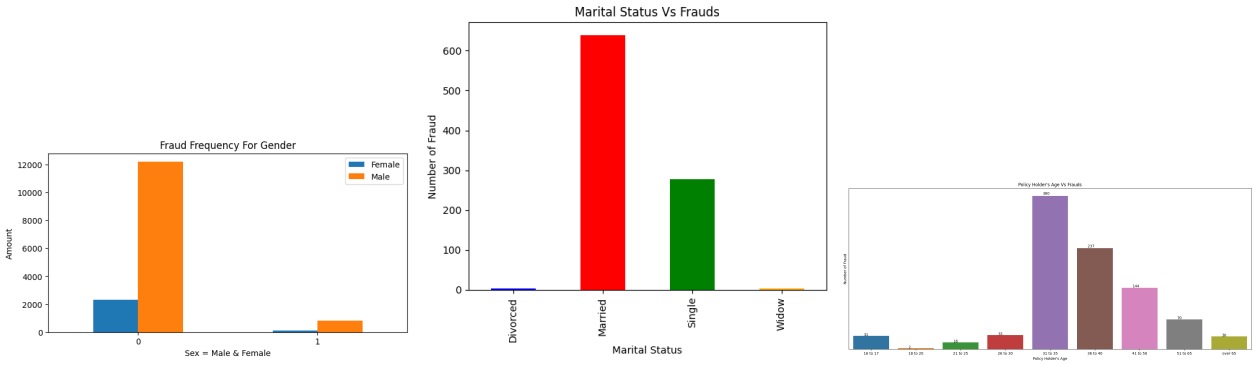
Step 1: Data Selection:

The data for this project was sourced [8] from kagele.com. The dataset contains information about vehicles such as their attributes, model, accident details, as well as policy details like policy type, tenure, etc. The aim of this project is to detect fraudulent claim applications, which is indicated by the target variable ”FraudFoundP”. The dataset has a total of 33 columns, comprising both independent and dependent features, and has 15,420 records. The reason for selecting this dataset is that it contains more than 33 columns, including numerical and categorical records, which can help in evaluating the performance of the model. And The dataset is described in below fig no.2

Feature Name	Description	Feature Name	Description
Month	Month in which the policy was taken	PolicyNumber	Unique identifier for the policy
WeekOfMonth	Week number of the month in which the policy was taken	RepNumber	Unique identifier for the sales rep.
DayOfWeek	Day of the week in which the policy was taken	Deductible	Amount of deduct on the policy
Make	Make of the insured vehicle	DriverRating	Rating of the driver
AccidentArea	Area where the accident occurred	Days_Policy_Accident	Number of days between policy and accident
DayOfWeekClaimed	Day of the week in which the claim was made	Days_Policy_Claim	Number of days between policy and claim
Month Claimed	Month in which the claim was made	PastNumberOfClaims	Number of claims in past
WeekOfMonthClaimed	Week of the month in which the claim was made	AgeOfVehicle	Age of the insured vehicle
Sex	Sex of the policyholder	AgeOfPolicyHolder	Age of the policyholder
Fault	Who was at fault for the accident	PoliceReportFiled	Whether a police report was filed or not
PolicyType	Type of policy	WitnessPresent	Whether a witness was present or not
VehicleCategory	Category of the insured vehicle	AgentType	Type of sales agent
Vehicle Price	Price of the insured vehicle	NumberOfSuppliments	Number of supplemental policies
Number of supplemental policies	Number of supplemental policies	AddressChange_Claim	Whether the address changed after the claim
NumberOfCars	Number of cars owned by the policyholder	BasePolicy	Base policy of the insured vehicle

Fig. 2. Crime Dataset

Step 2: Pre-processing and EDA: In pre- processing null Values are cheked in the data set there were no null value found . SO I checked the unique values of all categorical data and found thre are zero values involved.SO I found out in colum. The DayOfWeekClaimed, MonthClaimed and age have zero values in three values all of these column can not be zero . For MonthClaimed and DayOfWeekClaimed has a one single record which has zero value. For this we dropped the specific row And Age has 312 rows containing 0 as age. And for the ages we map the value using median of the age column. Since the policy number was merely an index number, it was dropped when the drop function was first used.



(a) Gender Vs. Claim (b) Married Status Vs. Fraud (c) Age group Vs. Fraud

Fig. 3. Fig No. 2

Eda: has been done to know the interesting knowledge about the data set. Some interesting facts are discovered in above fig no. 2. Few of the observation listed below:-
No of men are more than female on dataset men are quite likely to claim insurance than women.
Married individuals are more likely to claim fraud compared to others, such as single or widowed individuals.
The age group of 31 to 40, especially 31 to 35, had the highest number of fraud claims. Out of 923 fraud cases, 360 were claimed by this age group. **outliers** is being removed by using the box plot in fig no. 4 deductable and ages have the outliers so I have manually dropped the outliers. From Deductible based on the box plot in this column value more than 500 are being dropped total 311 records dropped. To make the numerical data normally distributed

Fig. 4. Box plot

Encoding:as our data set involves lots of details in the form of objects so for the utilization of those columns we have to use the encoder for that I have employed the label encoder from skit learn module
for further feature selection I have employed the co-liniarity of the independent variable to dependent variable and dropped the columns which was hardly related to the target variable. droped columns are (Month,WeekOfMonth,DayOfWeek,Days_Policy_Accident, MaritalStatus,RepNumber,DayOfWeekClaimed,MonthClaimed,WeekOfMonthClaimed,DriverRating)

Similar to co-linearity for multi co-linearity heat map is being drawn and dropped age and deductible columns are as both were highly correlated with other features.

Step 3: Data Preparation: In last out of 32 depended colums and I independed colum we have selected 18 as our final feature column and 1 as the target variable.

Out of 15108 records only 905 cases are being found as data set is extremely imbalance so Making it balance I have done the resampling using the resampling library from sk learn.units based on majority data which is fraud not detected I have equlise the data and made the data set size to 28406 which has equal no. of fraud detected to no fraoud .

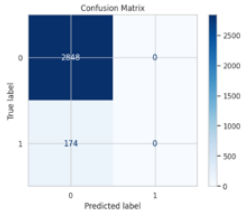
After this now we have two data set one which is imbalance aother one is balanced data set We applied test_train_split from sklearn to divide the data into 80:20 ration for traing and testing with random sample equals to 42.

Step 4: Model Training and Evaluation: Three Machine algorithum is applied on balanced and unbalanced data set are Linear regression, Random forest and Xg boost. Linear regresstion and random forest is applied with 2 library. The result of each algorithm are compared on unbalanced and banlanced data set .Hypertuning is applied with using grid search cv to get the best results.

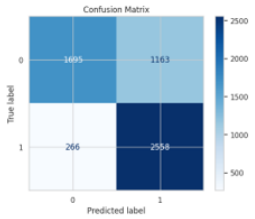
Confution matrix and Roc-auc curve is ploted and Classification reports for all the model to get the more understanding of the result

Model-1:we used linear classification from the Sklearn library

In case of unbalanced dataset. First, we built the model without tuning and achieved a testing accuracy of 0.94, with training accuracy also close to the testing accuracy. Next, we applied grid search CV to obtain the best parameters for the model configuration. Despite the model showing 0.94 accuracy on the test data, it failed to classify fraud cases ('1' class), although it successfully verified all cases of non-fraud.



LR	Parameters	0 "Non-Fraud"	1 "Fraud"
Unbalanced Data	Precision	0.93	0
	Recall	1	0
	f1-score	0.97	0
	Testing Accuracy	0.93	
	TrainingAccuracy	0.94	
	AUC	0.5	
	Cohen-kappa	0	



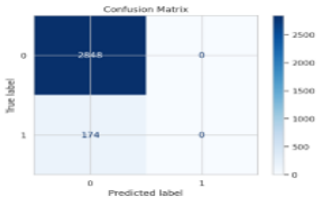
LR	Parameters	0 "Non-Fraud"	1 "Fraud"
balanced Data	Precision	0.86	0.59
	Recall	0.69	0.91
	f1-score	0.78	0.70
	Testing Accuracy	0.74	
	TrainingAccuracy	0.75	
	AUC	0.75	
	Cohen-kappa	0.49	

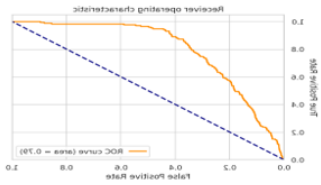
Fig. 5. Model-1 Summary

In the case of the balanced data model, without tuning, it achieved an accuracy of 0.74, which is not really good. However, after applying grid search CV to obtain the best model parameters, the accuracy improved to 0.75. In the case of the balanced data, it correctly identified 2568 fraud claims out of 2824. Additionally, the AUC is 0.75.

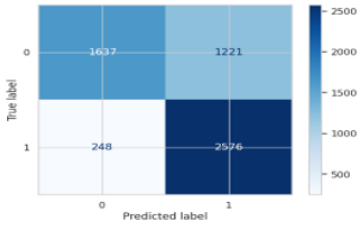
Model-2 We are using linear classification with a stats module on this model. fig no. shows the summary of the model

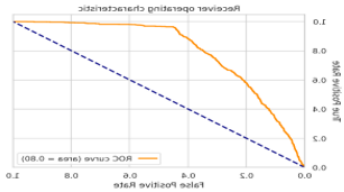
In the case of the unbalance data, the set model is not able to classify fraud class however it is performing with 0.94 accuracy in the case of classifying the non-fraud claims.





LR(Stats Module)	Parameters	0 "Non-Fraud"	1 "Fraud"
Unbalanced Data	Precision	0.93	0
	Recall	1	0
	f1-score	0.97	0
	Testing Accuracy	0.93	
	TrainingAccuracy	0.94	
	AUC	0.79	
	Cohen-kappa	0	





LR (Stats Module)	Parameters	0 "Non-Fraud"	1 "Fraud"
balanced Data	Precision	0.87	0.68
	Recall	0.57	0.91
	f1-score	0.69	0.78
	Testing Accuracy	0.74	
	TrainingAccuracy	0.74	
	AUC	0.75	
	Cohen-kappa	0.49	

Fig. 6. Model-2 Summary

In The Case of a Balanced data set, however, we are getting 0.74 as the testing accuracy but out of 2824 fraud claims it can classify correctly 2576, and in the case of the non-fraud cases it can classify 1627. And Also the Auc is also improved from the last model.

Model-3:

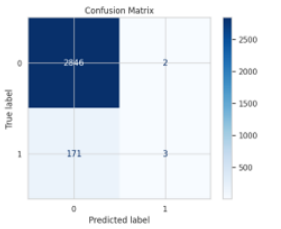
In model 3 I have applied Random forest model,Fig. no.7 represents the summary of the model

In the case of Balanced data set out of 174 fraud it is only able to classify correctly 3 .It is not able to classify the fraud class where as for non fraud claim it is working really well. However it obtained 94 percentage of accuracy. The reason of not able to classify the fraud class it might be the number of fraud claim is really less as compared to non froud class. Cohen- kappa score is also not desirable

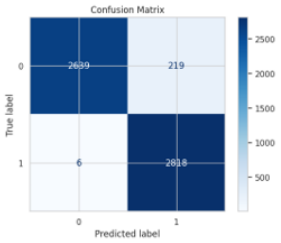
The model is performing exceptionally well on the balanced data set. It can correctly classify 2818 out of 2824 fraud claims and 2639 out of 2858 non-fraud claims. The training accuracy is also high at around 98 %. The AUC score of 0.96 is highly desirable, indicating the model's excellent performance in distinguishing between positive and negative classes. Additionally, the Cohen-kappa score is close to 1, indicating the model's overall effectiveness.

This model is perforerimg better than last two model in both cases.

Model-4: In this model Random forest classifier is applied using the H2o Library,Fig no. 8 contains the summary of model. In case of the Unbalanced data, the testing accuracy of the model is 0.94 which is really good. However, in the case of the fraud class, it is not classifying accurately, as only 66 cases are being correctly classified out of 174 fraud cases. On the other hand, for the non-fraud class, the model is able to correctly classify 2530 out of 2848 non-fraud claims. The Cohen kappa



RF (SK Learn)	Parameters	0 "Non-Fraud"	1 "Fraud"
Unbalanced Data	Precision	0.94	0.60
	Recall	1	0.02
	f1-score	0.97	0.03
	Testing Accuracy	0.94	
	TrainingAccuracy	0.94	
	AUC	0.51	
	Cohen-kappa	0.03	



RF (SK Learn)	Parameters	0 "Non-Fraud"	1 "Fraud"
balanced Data	Precision	1	0.93
	Recall	0.92	1
	f1-score	0.96	0.96
	Testing Accuracy	0.97	
	TrainingAccuracy	0.96	
	AUC	0.96	
	Cohen-kappa	0.92	

Fig. 7. Model-3 Summary

score is just 0.17, indicating poor agreement between the predicted and actual values. Furthermore, the F1 score in the case of the fraud class is only 0.24, which is not desirable and suggests that the model's precision and recall for fraud claims are not balanced.

In the case of the balanced dataset, the testing accuracy of the model is approximately 96%, which is excellent. Furthermore,

RF (H20)	Parameters	0 "Non-Fraud"	1 "Fraud"
balanced Data	Precision	0.98	0.93
	Recall	0.92	0.95
	f1-score	0.95	0.95
	Testing Accuracy	0.95	
	TrainingAccuracy	0.96	
	AUC	0.96	
	Cohen-kappa	0.98	

RF (H20)	Parameters	0 "Non-Fraud"	1 "Fraud"
Unbalanced Data	Precision	0.96	0.17
	Recall	0.89	0.38
	f1-score	0.92	0.24
	Testing Accuracy	0.94	
	TrainingAccuracy	0.97	
	AUC	0.51	
	Cohen-kappa	0.17	

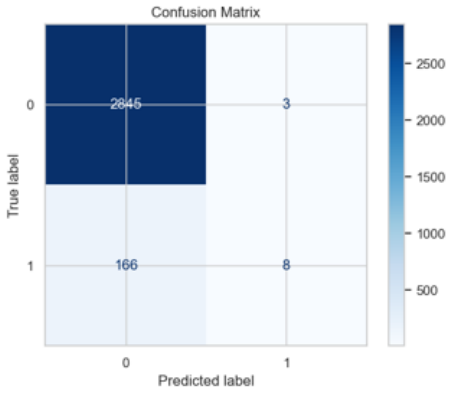
Fig. 8. Model-4 Summary

the F1 score is 0.95 for both classes, indicating good agreement between them. The model is able to correctly predict 2769 out of 2824 fraud cases and 2634 out of 2854 non-fraud cases. Additionally, the Cohen kappa score is close to 1, which is a highly desirable indicator of a good model performance.

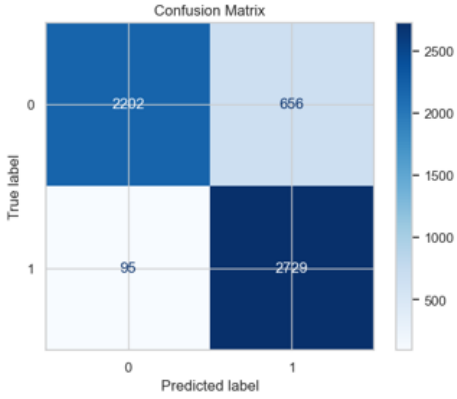
Model-5: In this model I have applied xg boost algorithm with hypertuning to obtain best results .Fig. 9 represents summary of model.

In a balanced dataset, this algorithm performs similarly to the random forest algorithm. However, while it achieves an accuracy of 94 percentage , it struggles to correctly classify the fraud class, correctly identifying only 8 out of 174 cases. In contrast, it performs well in classifying the non-fraud class. The Cohen kappa score of 0.08 indicates that the model is not performing well

in terms of agreement between its predictions and actual values.// In the case of the balanced dataset, the algorithm achieves



XG Boost	Parameters	0 "non-Fraud"	1 "Fraud"
Unbalanced Data(with Hyper tuning)	Precision	0.94	0.73
	Recall	1	0.05
	f1-score	0.09	0.97
	Testing Accuracy	0.94	
	TrainingAccuracy	0.94	
	AUC	0.81	
	Cohen-kappa	0.080	



XG Boost	Parameters	0 "non-Fraud"	1 "Fraud"
balanced Data (with Hyper tuning)	Precision	0.96	0.81
	Recall	0.77	0.97
	f1-score	0.85	0.88
	Testing Accuracy	0.88	
	TrainingAccuracy	0.87	
	AUC	0.94	
	Cohen-kappa	0.73	

Fig. 9. Model-5 Summary

an accuracy o Click to hide the PDF f 88 percentage, which is lower than the first dataset. However, the fact that the testing and training accuracy are approximately the same is a positive sign, indicating that the model is not overfitting. The AUC of 0.94 is high, indicating that the algorithm can distinguish between the two classes well. Additionally, it successfully identifies 2729 out of 2825 cases of fraud, which is impressive. The F1 score for both classes is around 0.86, indicating that the model is achieving a good balance between precision and recall and is not overfitting to one class.

Step 5: Interpretation of results: In the case of unbalanced data, models 3, 4, and 5 are showing similar results, and there is not a significant difference between them. However, none of these models are performing well, likely due to the highly imbalanced nature of the dataset, where the number of fraud cases is negligible compared to non-fraud case.

And In case of the balanced dataset random forrest and xg boost are providing the best results although the accuracy in the case of ranndom forrest is more but if we consider other paramert first one is model -4 rf with h20 module and second one is tunned xg boost and 3rd one is model 3 random forest with sk learn.

B. Fraudulent Claim on Cars Physical Damage(Dataset-2)

Step 1: Data Selection: Data is sourced from Kaggle named " Fraudulent Claim on Cars Physical Damage"containing more than 17000 records with 25 coulms containing the information of fraud claims has been processed by the policy holder.

fig no.10 contains the data set brief data set introduction.

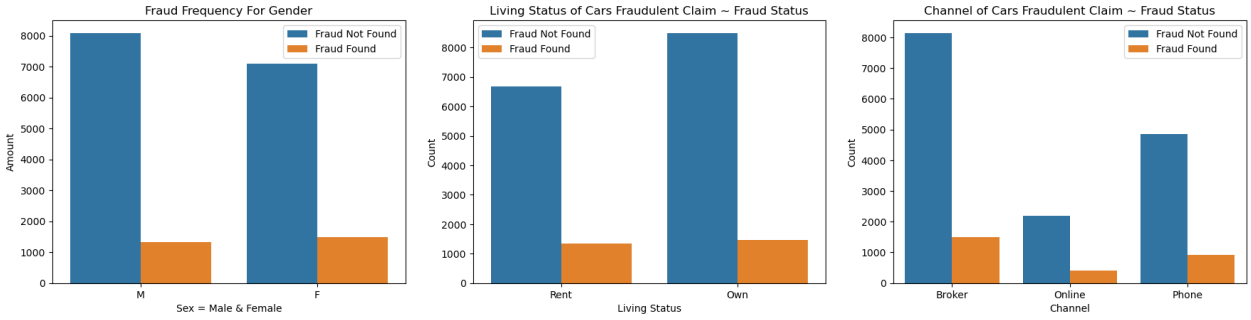
The reason for selecting this dataset is that it contains valuable features that can help us understand how to identify fraudulent claims before they are processed using machine learning. Additionally, the dataset contains both numerical and categorical features, which can help us understand which features are important for predicting whether a claim is a potential fraud or not.

Step 2: Pre-processing and EDA: EDA (Exploratory Data Analysis) has been performed to gain insights into the data and

Column Name	Description	Column Name	Description
claim_number	unique identifier for each claim	age_of_vehicle	age of the vehicle at the time of the accident
marital_status	marital status of the driver	vehicle_price	price of the vehicle
annual_income	annual income of the driver	vehicle_weight	weight of the vehicle
high_education_ind	indicates if the driver has a higher education	fraud	binary variable indicating fraudulent or not
address_change_ind	indicates if the driver has changed address	living_status	living status of the driver
claim_date	date of the accident	claim_day_of_week	day of the week of the accident
policy_report_filed_ind	indicates if a policy report was filed	liab_prct	percentage of liability for the accident

Fig. 10. Crime Dataset

to better understand it.
In Fig no. 11 (a) shows that there are more men than women in the dataset, but there is no significant difference in the number of fraud claims between the genders.
Furthermore,Fig no. 11 (b) there is no significant difference in the number of fraud claims between homeowners and renters,as indicated by the corresponding figure.
In the Fig no. 11 (c)The broker channel is found to be associated with a higher number of fraud claims. This suggests that if the insurance is obtained through a broker, it is more likely to involve fraud.



(a) Gender Vs. Claim (b) Fraud frequency for Gender (c) Booking channel

Fig. 11. Dataset no.2 EDA

Outliers were detected using box plots. Fig. no.12 shows the box plot of the dataset, and outliers were found in vehicle price, age of driver, and annual income. A total of 188 records were dropped manually from the data frame to remove the outliers and reduce the skewness of the data.
Encoding is a necessary step in preparing the dataset for modeling, especially when it involves a mix of numerical and categorical data. Object data, which represents categorical variables, needs to be converted to integer format to enable the model to work with them. To achieve this, I first printed the unique values of all object categorical features, such as gender and claim day of the week, and manually encoded them. For the remaining features, I used the ordinal encoder from the sklearn library to encode them.
The FIg. 13 shows the **correlation** of the target variable with the dependent variables. If a feature is not correlated with the target variable, it will have no effect on the model. Therefore, we can drop columns that have negligible correlation values, such as vehicle price with a correlation value of -0.001, which is almost negligible.
Furthermore, a **multicollinearity** (Fig. 14)plot has been used to detect collinearity between the dependent variables. We dropped

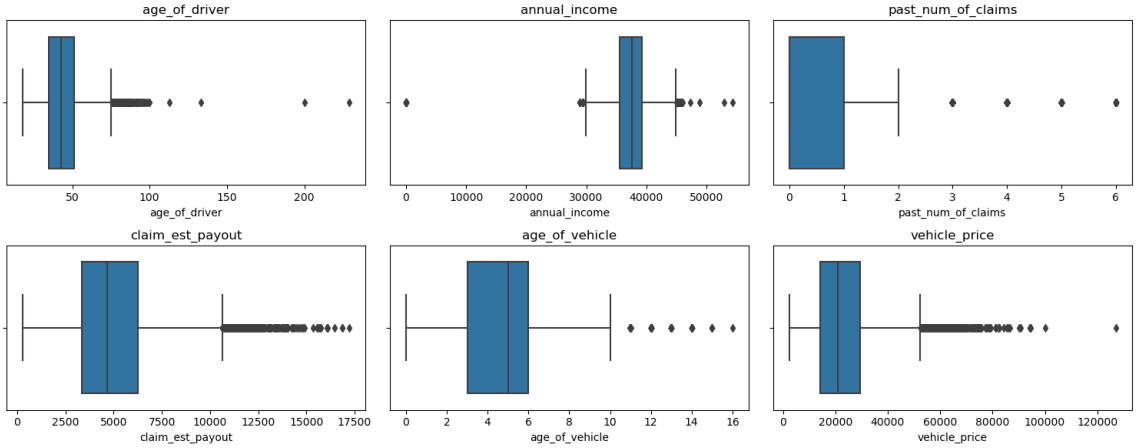


Fig. 12. BoxPlot for Dataset no.2

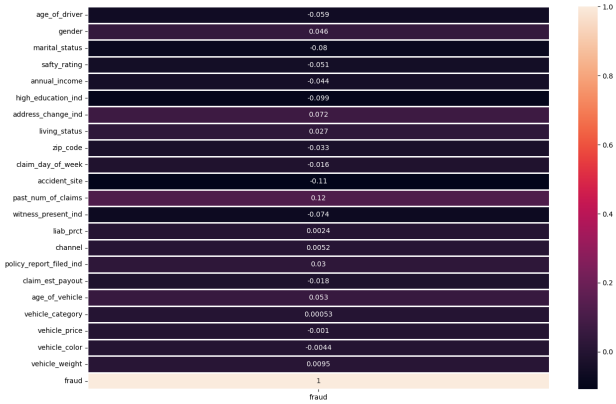


Fig. 13. Co-liniarity of the features with target variable

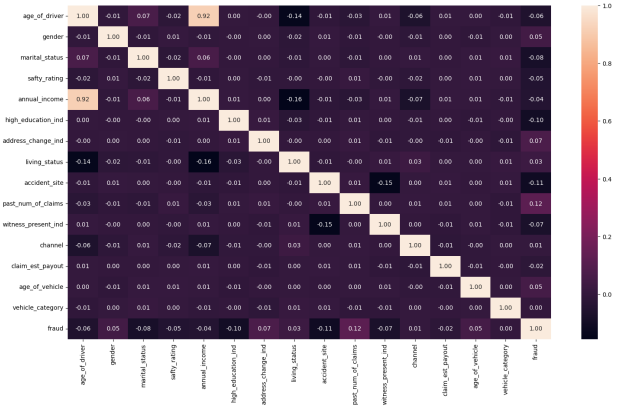


Fig. 14. Multicoliniarity of the dependent variables

the columns that have a correlation value of more than 0.6 in both positive and negative directions. From the plot, it is clear that annual income is highly correlated with age, so we decided to drop the column.

VIF(Variance inflation factor) was calculated to check for **Multicollinearity**, and it was found that the features ”annual income,” ”age of driver,” and ”safety rating” had VIF values greater than 10. To address this, the columns were normalized using the **Min-Max scaler** [11]from the sklearn module. This reduced the VIF value for ”age of driver” and ”claim est payout,” but ”safety rating” still had a VIF greater than 10. Therefore, ”safety rating” was dropped from the features. After dropping the ”safety rating” feature, all remaining features had a VIF value less than 10, which is desirable.

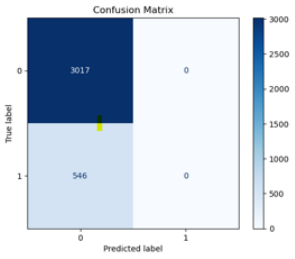
Step 3: Data Preparation: In the given dataset, there are only 2816 records of fraud cases out of 18000, making it imbalanced. To tackle this issue, we have applied random oversampling to increase the number of fraud cases and achieve an equal distribution of both outcomes, resulting in an enlarged dataset of 30034 rows.

At last , both the original dataset and the oversampled dataset have been divided into test and train sets in a 20:80 ratio, with a random state of 42 to ensure the data is shuffled.

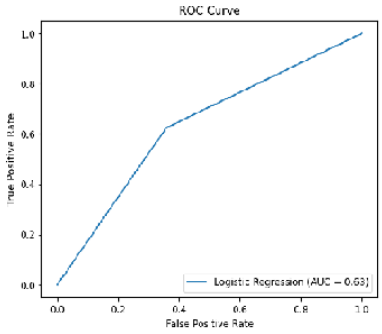
Model 1:- I have applied Linear regresstion on this model.

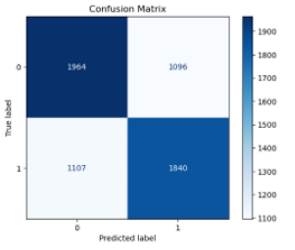
In case of unbalanced dataset it was unable to classify the fraud class ’1’ in the case of imbalanced data, despite achieving a testing and training accuracy of 0.84. Instead, it classified all data into the non-fraud class, resulting in a Cohen-Kappa score of 0, indicating poor model performance.

To address this issue, I used the oversampled dataset, resulting in a more balanced dataset with equal representation of both fraud and non-fraud cases. The testing and training accuracy for this dataset were around 64%, which is desirable, and the model was able to successfully classify 1840 out of 2947 fraud cases and 1964 out of 3060 non-fraud cases. The F1 score for both classes was close to 0.64, indicating a balance between the two classes. The Cohen-Kappa score was 0.26, which is not



LR (stats Model)	Parameters	0 "non-Fraud"	1 "Fraud"
Unbalanced Data (with Hyper tuning)	Precision	0.85	0
	Recall	1	0
	f1-score	0.92	0
	Testing Accuracy	0.84	
	TrainingAccuracy	0.84	
	AUC	0.5	
	Cohen-kappa	0	





LR (Stats Model)	Parameters	0 "non-Fraud"	1 "Fraud"
Oversampled (with Hyper tuning)	Precision	0.64	0.63
	Recall	0.64	0.62
	f1-score	0.64	0.63
	Testing Accuracy	0.64	
	TrainingAccuracy	0.63	
	AUC	0.63	
	Cohen-kappa	0.26	

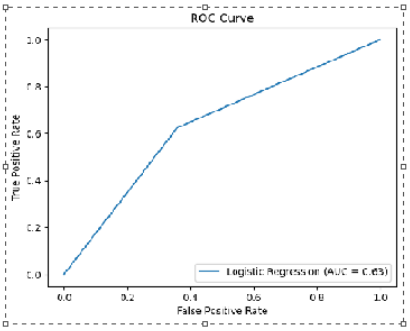


Fig. 15. Model 1 Detailed summary

bad, and the AUC was 0.63.

Fig. 15 provides the detailed summary of the both models

Model-2: In this we have implemented the Logistic regression using the Stats module

Fig. 16 contains detailed summary of the model

In the case of an imbalanced dataset, despite achieving an accuracy of 84%, the model was only able to classify 15 out of 546 fraud cases, while performing well in the non-fraud class. The F1 score for the fraud class was just 0.05, indicating that the model was unbalanced.

In case of using the oversampled dataset, although the testing and training accuracy was only 64 percentage, the model was able to achieve balance between both classes, classifying 1867 out of 2940 fraud cases and 1955 out of 3060 non-fraud cases. The Cohen-Kappa score was 0.27, indicating that the model's performance is better

Model-3: In this case we applied random forrest classifier algorithum using sk learn module.

Fig. 17 contains the detailed summary of the model In the case of the unbalanced dataset, despite achieving 85 percentage accuracy in training and testing, the model was only able to classify 5 out of 546 fraud cases, resulting in an F1 score of just 0.02. This indicates that the model is performing poorly due to the imbalance between both classes. However, the model was able to classify 3016 out of 3017 non-fraud cases accurately. The Cohen-Kappa score for this model was also low at 0.014, further highlighting the poor performance.

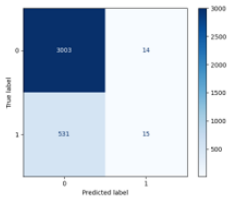
In contrast, when using the oversampled dataset, the model achieved nearly 100 percentage accuracy on the training data and 0.91 percentage accuracy on the testing data. The F1 score for both classes was equal, indicating a balance between them. Out of 2947 fraud cases, the model was able to classify 2931 correctly, and the Cohen-Kappa score was close to 1, indicating excellent model performance. Additionally, the AUC was 0.96, further highlighting the model's strong performance.

Model-4: In this model I have used random model classifier with h20 library.

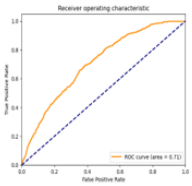
Fig. 18 contains the detailed summary of the model

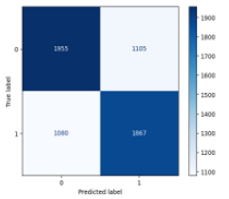
In the case of the unbalanced dataset, the model achieved training accuracy close to 99 percentage and testing accuracy close to 85%. Although the model was able to classify both classes, the f1 score for the fraud class was only 0.34 while for non-fraud claims it was 0.74, indicating some imbalance in the model's performance. However, it performed better than the previous three models. The model was able to classify 355 out of 546 fraud cases and 1818 out of 3047 non-fraud cases. The Cohen kappa score was 0.14, indicating poor performance in classifying fraud cases. The AUC was also 0.66.

In the case of the balanced dataset, the model achieved training and testing accuracy close to 99%. The f1 score of the model was 0.99 for both classes, indicating balance in classifying both classes. The model was able to classify 2886 out of 2947



LR (stats Model)	Parameters	0 "non-Fraud"	1 "Fraud"
Unbalanced Data (with Hyper tuning)	Precision	0.85	0.52
	Recall	1	0.03
	f1-score	0.92	0.05
	Testing Accuracy	0.84	
	TrainingAccuracy	0.84	
	AUC	0.71	
	Cohen-kappa	0.03	





LR (Stats Model)	Parameters	0 "non-Fraud"	1 "Fraud"
Oversampled (with Hyper tuning)	Precision	0.64	0.63
	Recall	0.64	0.62
	f1-score	0.64	0.63
	Testing Accuracy	0.64	
	TrainingAccuracy	0.63	
	AUC	0.69	
	Cohen-kappa	0.27	

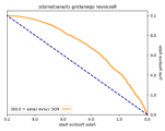
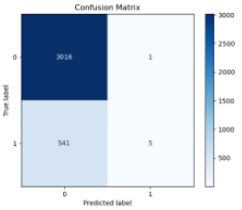
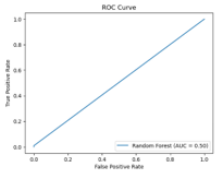
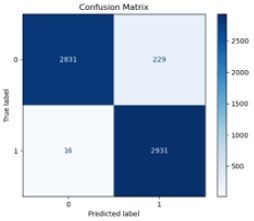


Fig. 16. Model 2 Detailed summary



RF (Sk learn)	Parameters	0 "non-Fraud"	1 "Fraud"
Unbalanced Data	Precision	0.85	0.83
	Recall	1	0.01
	f1-score	0.92	0.02
	Testing Accuracy	0.84	
	TrainingAccuracy	0.85	
	AUC	0.50	
	Cohen-kappa	0.014	





RF (SK learn)	Parameters	0 "non-Fraud"	1 "Fraud"
Oversampled	Precision	0.99	0.93
	Recall	0.93	0.99
	f1-score	0.96	0.96
	Testing Accuracy	0.95	
	TrainingAccuracy	1	
	AUC	0.96	
	Cohen-kappa	0.91	

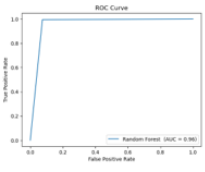
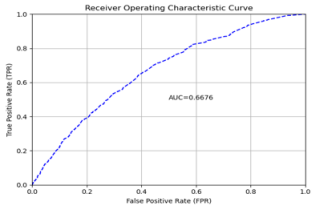


Fig. 17. Model 3 Detailed summary

RF (H2o)	Parameters	0 "non-Fraud"	1 "Fraud"
Unbalanced Data	Precision	0.90	0.23
	Recall	0.60	0.03
	f1-score	0.72	0.34
	Testing Accuracy	0.85	
	TrainingAccuracy	0.99	
	AUC	0.96	
	Cohen-kappa	0.6676	



RF (H2o)	Parameters	0 "non-Fraud"	1 "Fraud"
Oversampled Data	Precision	0.98	098
	Recall	0.99	0.99
	f1-score	0.99	0.99
	Testing Accuracy	0.98	
	TrainingAccuracy	0.99	
	AUC	0.99	
	Cohen-kappa	0.97	

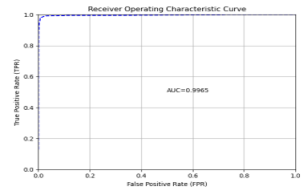


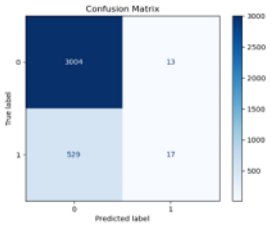
Fig. 18. Model 4 Detailed summary

fraud cases and 3038 out of 3060 non-fraud cases. The Cohen kappa score was also 0.97, indicating strong performance of the model. The AUC was close to 1.

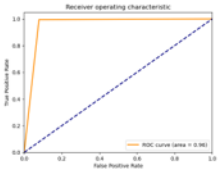
Model-5: In this data set , I have applied the tunned XG boost model with hypertunning using grid search cv // Fig. 19 contains the detailed summary of the model

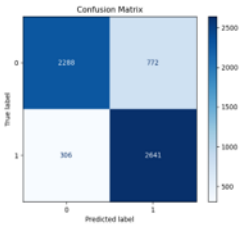
In the case of the unbalanced dataset, the model achieved training and testing accuracies close to 85%, but the poor F1 score of 0.06 for the fraud class indicated a significant performance imbalance between the two classes. The model was only able to classify 17 out of 546 fraud cases, while its performance on the non-fraud class was better. The Cohen kappa score was also poor at just 0.04.

In contrast, the oversampled dataset produced better results. The model achieved a training accuracy of 90 percentage and a



XG BOOST	Parameters	0 "non-Fraud"	1 "Fraud"
Unbalanced Data (with Hyper tuning)	Precision	0.85	0.57
	Recall	1	0.03
	f1-score	0.92	0.06
	Testing Accuracy	0.84	
	TrainingAccuracy	0.84	
	AUC	0.96	
	Cohen-kappa	0.04	





XG BOOST	Parameters	0 "non-Fraud"	1 "Fraud"
Oversampled (with Hyper tuning)	Precision	0.88	0.75
	Recall	0.75	0.90
	f1-score	0.81	0.83
	Testing Accuracy	0.90	
	TrainingAccuracy	0.82	
	AUC	0.82	
	Cohen-kappa	0.64	

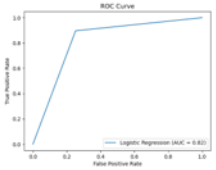


Fig. 19. Model 5 Detailed summary

testing accuracy of 82 percentage. The F1 scores for both classes were close to 0.81 and 0.83, respectively, indicating a better balance in the model's ability to classify both classes. The Cohen kappa score was also higher at 0.64, indicating a stronger performance. The model was able to classify 2641 fraud cases out of 2947 and 2288 non-fraud cases out of 3060. The AUC score of 0.82 indicated good overall performance of the model.

Step 5: Interpretation of results: Out of all the models, Model No. 4 seems to perform better than the others on the unbalanced dataset with an AUC of 0.96. However, due to the dataset being imbalanced, none of the models were able to effectively balance between the two classes.

On the oversampled dataset, Models No. 3, 4, and 5 are all performing well, with the random forest model performing better than the XG Boost model from both libraries.

C. Car Insurance Fraud Claim(Dataset-3) This dataset was sourced from Kaggle.com and is called "Car Insurance Data." It was used to create a model for detecting vehicle insurance fraud, and consists of over 10,000 real records. The dataset has 18 dependent variables and 1 independent variable. The reason for choosing this dataset is that it contains a mix of numerical

Column Name	Description	Column Name	Description
AGE	Age of the policyholder	GENDER	Gender of the policyholder
DRIVING_EXPERIENCE	Years of driving experience of the policyholder	EDUCATION	Education level of the policyholder
INCOME	Annual income of the policyholder	CREDIT_SCORE	The credit score of the policyholder
VEHICLE_OWNERSHIP	Whether the policyholder owns the vehicle (1) or not (0)	VEHICLE_YEAR	Year of the vehicle
MARRIED	Whether the policyholder is married (1) or not (0)	POSTAL_CODE	Postal code of the policyholder
CHILDREN	Number of children the policyholder has	ANNUAL_MILEAGE	Estimated annual mileage of the vehicle
PAST_ACCIDENTS	Number of past accidents of the policyholder	DUIS	Number of driving under the influence (DUI) violations by the policyholder

Fig. 20. Dataset-3 Description

and categorical columns, which can help us to understand which attributes are most important in detecting potential fraud claims. By analyzing this dataset, we can gain insights into the patterns and behaviors of fraudulent claims, which can help insurance companies to better detect and prevent fraud. Fig. 20 shows the brief introduction of feature columns.

1.Data selection This dataset was sourced from Kaggle.com It was used to create a model for detecting vehicle insurance fraud, and consists of over 10,000 real records. The dataset has 18 dependent variables and 1 independent variable. The reason for choosing this dataset is that it contains a mix of numerical and categorical columns, which can help us to understand which attributes are most important in detecting potential fraud claims. which can help insurance companies to better detect and prevent fraud.

2.pre-processing and EDA: implemented few function to optimise the code well.Over 900 null values were found in the credit score and annual mileage columns. The credit score was found to be directly related to class, with the poverty class having lower scores. Credit scores were filled based on their respective groups. Mean values were used to map the missing annual mileage values. Duplicate values were dropped.

Outliers: we used the IQr method and plotted them using a box plot. Since the dataset had few rows, we only removed 300 extreme outliers from the annual mileage, speed violation, and DUI columns.

During the **exploratory data analysis**, some interesting findings were made. fig. 23 a) there were more female customers than male customers in the dataset, and males were found to be more likely to make insurance claims than females Figure 23 c) shows that customers with more driving experience are less likely to make insurance claims or commit fraud. Additionally, Fig 23b) shows the highest fraud is claimed by poverty class

For **encoding**, we used binary and label encoders from the sk learn library after looking at the unique values of the categorical columns.

To avoid **Multicollinearity** issues, we used a correlation plot to check the correlation between columns. We removed the least correlated and highly correlated column before building the model.

Step 3: Data Preparation: After cleaning the data, I noticed that the outcome was imbalanced, Since I had a small dataset, I split it into 80% for training and 20% for testing, with a random state of 42 for result reproducibility.

To balance the data, I used the random oversampling technique to increase the number of defaulters/claims count. This was

Column Name	Description	Column Name	Description
AGE	Age of the policyholder	GENDER	Gender of the policyholder
DRIVING_EXPERIENCE	Years of driving experience of the policyholder	EDUCATION	Education level of the policyholder
INCOME	Annual income of the policyholder	CREDIT_SCORE	The credit score of the policyholder
VEHICLE OWNERSHIP	Whether the policyholder owns the vehicle (1) or not (0)	VEHICLE YEAR	Year of the vehicle
MARRIED	Whether the policyholder is married (1) or not (0)	POSTAL_CODE	Postal code of the policyholder
CHILDREN	Number of children the policyholder has	ANNUAL_MILEAGE	Estimated annual mileage of the vehicle
PAST ACCIDENTS	Number of past accidents of the policyholder	DUIS	Number of driving under the influence (DUI) violations by the policyholder

Fig. 21. Dataset-3 Description

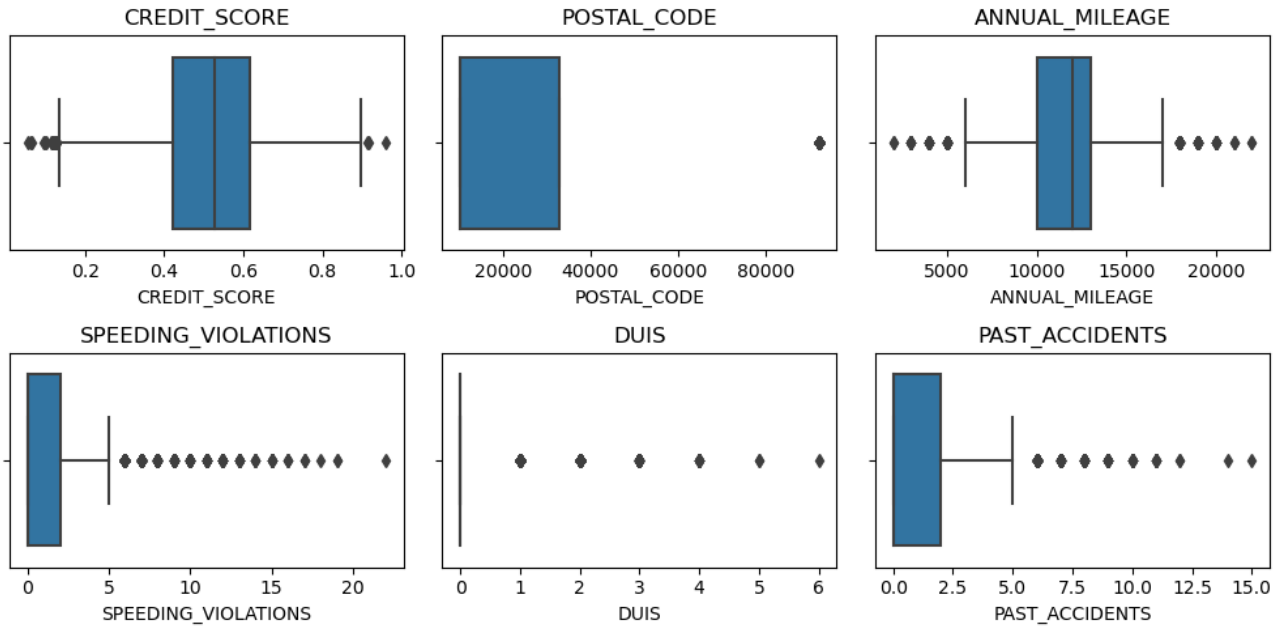


Fig. 22. Boxplot Dataset-3

necessary since I had less data, and the oversampling helped increase the number of claims from 2998 to 6679.

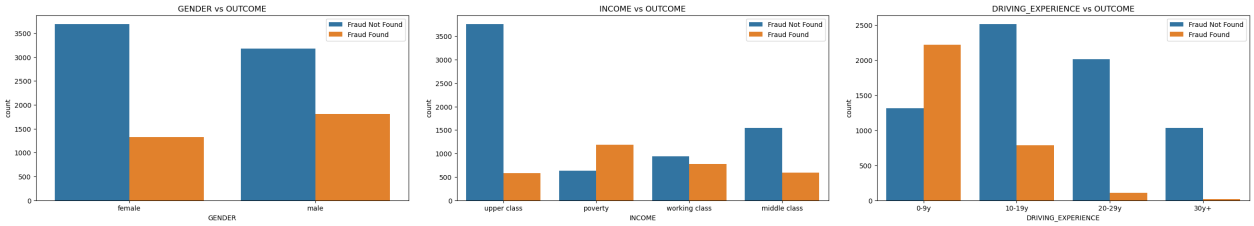
Step 4: Model Training and Evaluation: Created 5 different model from three different algorithms, namely Random Forest (RF), Logistic Regression (LR), and XGBoost. For LR and RF, I used different libraries, such as scikit-learn and stats model for LR and scikit-learn and H2O for RF.

Model-1: In this case linear regression algorithm applied using sk learn.Fig. 24 shows the detailed summary of the model Without oversampled data,the model achieved a testing accuracy of 84%, which was close to the training accuracy. It correctly classified 428 out of 603 fraud cases and 1216 out of 1335 non-fraud cases. However, the F1 score of both classes indicated an imbalanced performance of the model in identifying the two classes.

After applying oversampling to the data, the model's performance decreased slightly to 83%. However, the F1 score of both classes improved significantly, with a value of approximately 0.84, indicating better balance in the model's performance for both classes. Moreover, the Cohen's Kappa and AUC also improved. The model was able to correctly classify 1162 out of 1361 fraud cases.

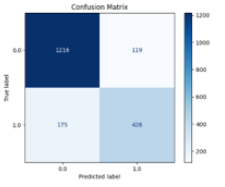
Model 2: In This model I have applied linear regression with stats module.Fig. 25 represents the detailed summary of the model.

The model performance in both cases is quite similar to Model-1, with only a slight improvement in the area under the curve.

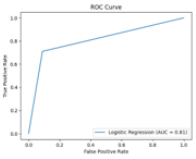


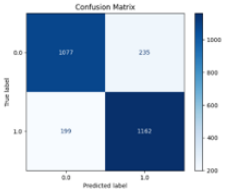
(a) Gender Vs. Claim (b) Fraud frequency for Gender (c) Driving Ex. vs age

Fig. 23. Dataset no.2 EDA



LR (SK Learn)	Parameters	0 "non-Fraud"	1 "Fraud"
Unbalanced Data (with Hyper tuning)	Precision	0.87	0.78
	Recall	0.91	0.71
	f1-score	0.89	0.74
	Testing Accuracy	0.84	
	TrainingAccuracy	0.84	
	AUC	0.81	
	Cohen-kappa	0.63	





LR (SK Learn)	Parameters	0 "non-Fraud"	1 "Fraud"
Oversampled (with Hyper tuning)	Precision	0.84	0.83
	Recall	0.82	0.83
	f1-score	0.83	0.84
	Testing Accuracy	0.81	
	TrainingAccuracy	0.83	
	AUC	0.84	
	Cohen-kappa	0.67	

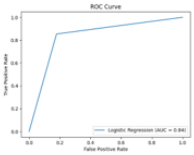


Fig. 24. Model 1 Detailed summary

This improvement indicates an increase in the overall performance of the model.

Model 3: In this model I have applied random foreest classifier using sk learn module.Fig. 26 represents the detailed summary of the model.

The model performance without oversampling resulted in an accuracy of 83%, with 412 out of 603 fraud cases correctly classified. The F1 score showed a class imbalance, with a Kohen kappa score of 0.59 and AUC of 0.79.

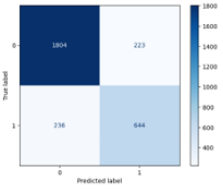
However, with oversampled data, the model's performance improved significantly, achieving a training accuracy of almost 99% and testing accuracy of 89%. The F1 score for both classes was around 90, indicating balance between the classes. The model was able to classify 1249 out of 1361 fraud claims and 1150 out of 1312 non-fraud claims, with a Kohen kappa score of 0.79 and AUC of 0.90.indicating a stronger agreement between the model's predictions and actual outcomes.

Model 4: In this I have applied the random forrest classifier with the help of h20 library .Fig. 27 represents the detailed summary of the model.

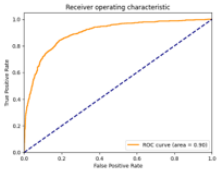
Without oversampled data, the model's performance was similar to Model 3, with 432 out of 603 fraud cases and 1148 out of 1335 non-fraud cases correctly classified. The F1 score was slightly better than the previous model.

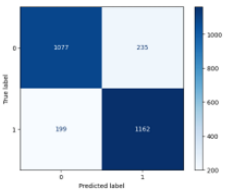
Fig. 27 represents the detailed summary of the model.

However, with oversampled data, the model's performance improved significantly. During training, the model achieved almost 100% accuracy, while during testing it achieved 90%, slightly better than Model 3. It was able to classify 1238 out of 1361 fraud claims and 1167 out of 1312 non-fraud cases, with an AUC of 0.94 and Kohen kappa of 0.79. These metrics indicate a strong performance in both identifying fraud cases and avoiding false positives, with a high level of agreement between the predicted and actual outcomes.



LR (Stats Module)	Parameters	0 "non-Fraud"	1 "Fraud"
Without Sampling Data	Precision	0.88	0.74
	Recall	0.89	0.73
	f1-score	0.89	0.74
	Testing Accuracy	0.84	
	TrainingAccuracy	0.84	
	AUC	0.90	
	Cohen-kappa	0.62	





LR (Stats Module)	Parameters	0 "non-Fraud"	1 "Fraud"
Oversampled	Precision	0.84	0.83
	Recall	0.82	0.85
	f1-score	0.83	0.84
	Testing Accuracy	0.81	
	TrainingAccuracy	0.83	
	AUC	0.91	
	Cohen-kappa	0.67	

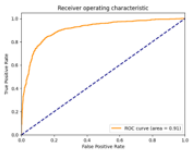
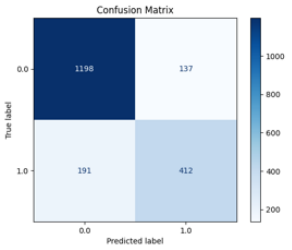
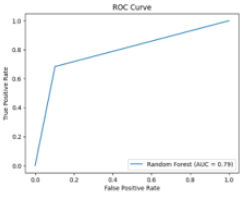
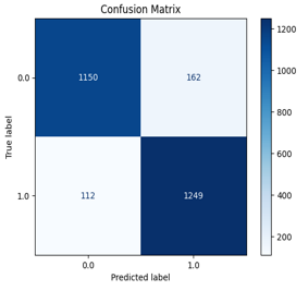


Fig. 25. Model 2 Detailed summary



RF (Sk learn)	Parameters	0 "non-Fraud"	1 "Fraud"
Unbalanced Data	Precision	0.86	0.75
	Recall	0.90	0.68
	f1-score	0.88	0.72
	Testing Accuracy	0.83	
	TrainingAccuracy	0.88	
	AUC	0.79	
	Cohen-kappa	0.59	





RF (SK learn)	Parameters	0 "non-Fraud"	1 "Fraud"
Oversampled	Precision	0.91	0.89
	Recall	0.88	0.92
	f1-score	0.89	0.90
	Testing Accuracy	0.89	
	TrainingAccuracy	0.99	
	AUC	0.90	
	Cohen-kappa	0.79	

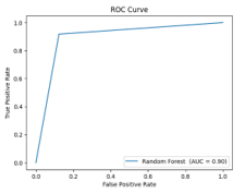
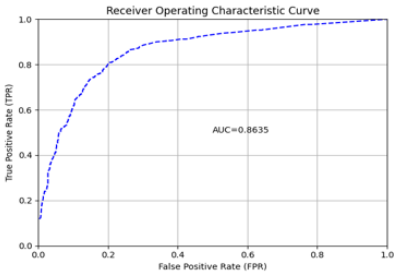


Fig. 26. Model 3 Detailed summary

RF (H2o)	Parameters	0 "non-Fraud"	1 "Fraud"
Unbalanced Data	Precision	0.87	0.70
	Recall	0.86	0.72
	f1-score	0.87	0.71
	Testing Accuracy	0.81	
	TrainingAccuracy	0.99	
	AUC	0.86	
	Cohen-kappa	0.57	



RF (H2o)	Parameters	0 "non-Fraud"	1 "Fraud"
Oversampled Data	Precision	0.90	098
	Recall	0.89	0.99
	f1-score	0.90	0.99
	Testing Accuracy	0.90	
	TrainingAccuracy	0.99	
	AUC	0.94	
	Cohen-kappa	0.79	

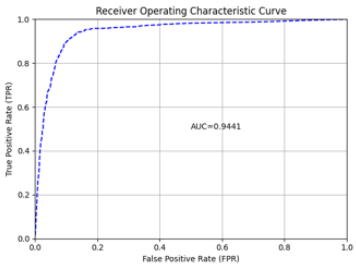


Fig. 27. Model 4 Detailed summary

Model 5(XG BOOST): Fig. 25 represents the detailed summary of the model.

XG BOOST	Parameters	0 "non-Fraud"	1 "Fraud"
Unbalanced Data (with Hyper tuning)	Precision	0.88	0.77
	Recall	0.90	0.72
	f1-score	0.89	0.74
	Testing Accuracy	0.84	
	TrainingAccuracy	0.84	
	AUC	0.90	
	Cohen-kappa	0.63	

XG BOOST	Parameters	0 "non-Fraud"	1 "Fraud"
Oversampled (with Hyper tuning)	Precision	0.88	0.86
	Recall	0.85	0.89
	f1-score	0.87	0.87
	Testing Accuracy	0.89	
	TrainingAccuracy	0.91	
	AUC	0.87	
	Cohen-kappa	0.73	

Fig. 28. Model 5 Detailed summary

When dealing with an imbalanced dataset, the model performance is similar to that of random forest.
However, with the oversampled dataset, the model's accuracy improved, achieving a training and testing accuracy of nearly 90%, indicating its improved performance.The F1 score for both classes was also around 0.87, indicating that the model was able to classify both classes effectively. Specifically, the model was able to classify 1205 out of 1361 fraud cases and 1120 out of 1312 non-fraud cases. The AUC was 0.8, indicating a good balance between the true positive and false positive rates, and the Kohen kappa score was 0.73, suggesting substantial agreement between the predicted and actual classifications.
Step 5: Interpretation of results:: Xg boost and Random forest both are providing best results.

IV. CONCLUSION AND FUTURE WORK

In **Data set 1**, due to highly imbalanced data, all algorithms did not perform well However, Random Forest with H2O framework performed well.
Fig. 29 (A) represents detailed summary of all ml algorithms In **balanced data**, three models produced superior results with an AUC greater than 0.9, except for models 1 and 2. The best model is Random Forest algorithm models 3 and 4, which had testing accuracies of 0.96 and 0.97, respectively. The difference in accuracy between the train and test sets was minimal, indicating low variance. Model 4 had a Cohen Kappa score of 0.98, which is close to one, and the best recall, accurately predicting 95 percent of default claimants and having the lowest false negative.

Final Model Summary (With Balanced data set) Data set -1										
Algorithms/Evaluation	**LR (Sk Learn)**		**LR (Stats Module)**		**RF(Sk learn)**		**RF(H2o)**		**XG Boost (Tunned)**	
	Non-Fraud	Fraud	Non-Fraud	Fraud	Non-Fraud	Fraud	Non-Fraud	Fraud	Non-Fraud	Fraud
Precision	0.86	0.59	0.87	0.68	1	0.93	0.98	0.93	0.96	0.81
Recall	0.69	0.91	0.57	0.91	0.92	1	0.92	0.95	0.77	0.97
F1- Score	0.78	0.70	0.69	0.78	0.96	0.96	0.95	0.95	0.85	0.88
Train accuracy	0.75		0.74		0.96		0.96		0.87	
Testing Accuracy	0.74		0.74		0.97		0.95		0.88	
AUC	0.75		0.75		0.96		0.96		0.94	
Cohen Kappa Score	0.49		0.49		0.92		0.98		0.73	

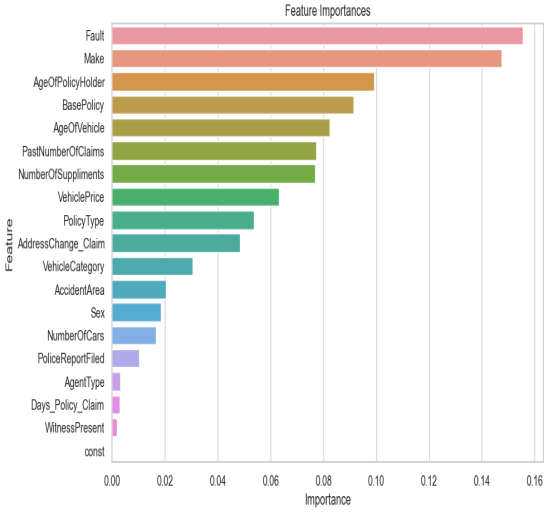


Fig. 29. A. Detailed Summary of All Model in data set 1, B. Importance of features on model .

Fig 29.(B) shows .According to the analysis of the training data set, the feature ”Fault” was found to be the most important feature, while the feature ”witness” was found to be the least important.

In case of **Dataset no.2**

From the table Shown in Fig. 30(A), it can be observed that the RF model with the H2O framework performed the best with a testing accuracy of 0.98 and an AUC of 0.99. It also has the highest precision, recall, and F1-score for the non-fraud class. The XG Boost model has the highest precision, recall, and F1-score for the fraud class. However, it has lower accuracy and AUC compared to the RF model.

It is also observed that the LR models have lower performance compared to the RF and XG Boost models

From the table, it can be observed that the RF model with the H2O framework performed the best with a testing accuracy of 0.98 and an AUC of 0.99. It also has the highest precision, recall, and F1-score for the non-fraud class. The XG Boost model has the highest precision, recall, and F1-score for the fraud class. However, it has lower accuracy and AUC compared to the RF model.

According to the analysis of the importance of fearures , ”claim_est_payout” was the most important feature and ”witness_present” was the least important.

Final Model Summary (With Balanced data set) Data Set -2										
Algorithms/Evaluation	**LR(Sk Learn)**		**LR(Stats Module)**		**RF(Sk learn)**		**RF(H2o)**		**XG Boost (Tunned)**	
	Non-Fraud	Fraud	Non-Fraud	Fraud	Non-Fraud	Fraud	Non-Fraud	Fraud	Non-Fraud	Fraud
Precision	0.64	0.63	0.64	0.63	0.99	0.93	0.98	0.98	0.88	0.75
Recall	0.64	0.62	0.64	0.62	0.93	0.99	0.99	0.99	0.75	0.90
F1- Score	0.64	0.63	0.64	0.63	0.96	0.96	0.99	0.99	0.81	0.83
Train accuracy	0.63		0.64		1		0.99		0.82	
Testing Accuracy	0.64		0.63		0.95		0.98		0.90	
AUC	0.63		0.68		0.96		0.99		0.82	
Cohen Kappa Score	0.26		0.27		0.91		0.97		0.64	

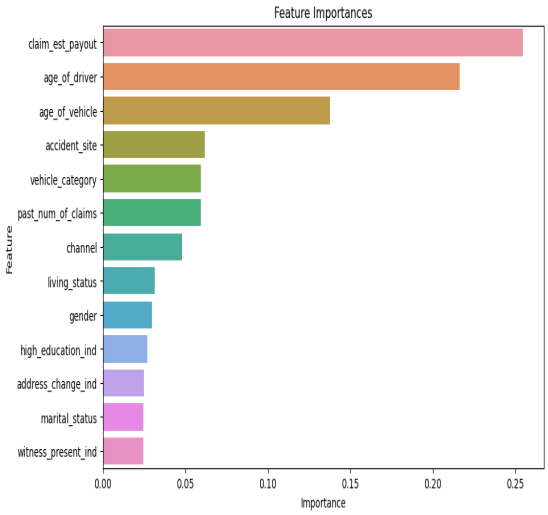


Fig. 30. A. Detailed Summary of All Model in data set 2, B. Importance of features .

The final model summary for **Data Set 3**, which used a balanced data set, shows that the Random Forest algorithm with

Final Model Summary (With Balanced data set) Data Set -3										
Algorithms/Evaluation	**LR (Sk Learn)**		**LR (Stats Module)**		**RF(Sk learn)**		**RF(H2o)**		**XG Boost (Tunned)**	
	Non-Fraud	Fraud	Non-Fraud	Fraud	Non-Fraud	Fraud	Non-Fraud	Fraud	Non-Fraud	Fraud
Precision	0.84	0.83	0.84	0.83	0.91	0.89	0.90	0.98	0.88	0.86
Recall	0.82	0.83	0.82	0.85	0.88	0.92	0.89	0.99	0.85	0.89
F1- Score	0.83	0.84	0.83	0.84	0.89	0.90	0.90	0.99	0.87	0.87
Train accuracy	0.83		0.83		0.99		0.99		0.89	
Testing Accuracy	0.81		0.81		0.89		0.90		0.91	
AUC	0.84		0.91		0.90		0.94		0.87	
Cohen Kappa Score	0.67		0.67		0.79		0.79		0.73	

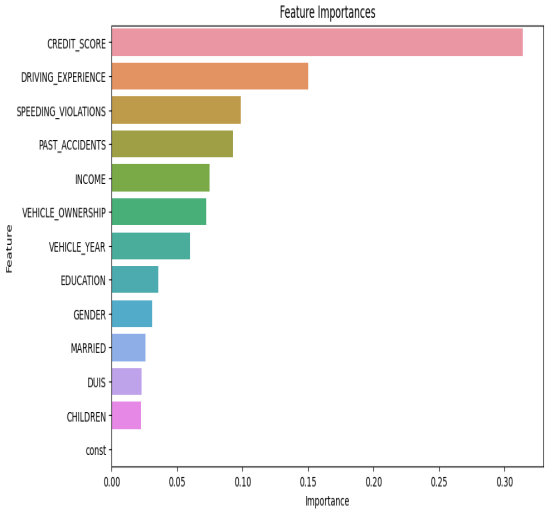


Fig. 31. A. Detailed Summary of All Model in data set 3, B. Importance of features .

H2O framework had the best performance with a precision of 0.90 and recall of 0.89. The least performing algorithm was Logistic Regression using Sk Learn with a precision of 0.84 and recall of 0.82. The most important evaluation metric for this data set was the Cohen Kappa Score, which had values ranging from 0.67 to 0.79.
According to the analysis of the importance of features, ”Credit score” most important, ”channel” least important feature

The features column could have been categorized based on their accessory list rating for more effective results. Additionally, using different methods for cleaning and filling missing values could have improved the accuracy. To achieve better results in the classification dataset, various sampling techniques could be employed. However, since only an imbalanced dataset was used, the evaluation was limited to varying tuning settings.

REFERENCES

[1] BY- IBM “What is machine learning?”Available:”https://www.ibm.com/topics/machine-learning
[2] N. S.Patil, S. Kamanavalli, S. Hiregoudar, S. Jadhav, S. Kanakraddi and N. D. Hiremath, ”Vehicle Insurance Fraud Detection System Using Robotic Process Automation and Machine Learning,” 2021 International Conference on Intelligent Technologies (CONIT), Hubli, India, 2021, pp. 1-5, doi: 10.1109/CONIT51480.2021.9498507.
[3] M. Hanafy en R. Ming, “Classification of the Insureds Using Integrated Machine Learning Algorithms: A Comparative Study”, Applied Artificial Intelligence, bll 1–32, 2022.
[4] R. Roy and K. T. George, ”Detecting insurance claims fraud using machine learning techniques,” 2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT), Kollam, India, 2017, pp. 1-6, doi: 10.1109/ICCPCT.2017.8074258..
[5] G. Kowshalya and M. Nandhini, ”Predicting Fraudulent Claims in Automobile Insurance,” 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2018, pp. 1338-1343, doi: 10.1109/ICICCT.2018.8473034.
[6] Y. Li, C. Yan, W. Liu and M. Li, ”Research and application of random forest model in mining automobile insurance fraud,” 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, China, 2016, pp. 1756-1761, doi: 10.1109/FSKD.2016.7603443.
[7] R. Roy and K. T. George, ”Detecting insurance claims fraud using machine learning techniques,” 2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT), Kollam, India, 2017, pp. 1-6, doi: 10.1109/ICCPCT.2017.8074258.
[8] S. Bansal, ”Vehicle Claim Fraud Detection,” Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/shivamb/vehicle-claim-fraud-detection
[9] S. Ramireddy, ”Fraudulent Claim on Cars - Physical Damage,” Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/surekharamireddy/fraudulent-claim-on-cars-physical-damage/code?select=training+data.
[10] S. Saha, ”Car Insurance Data,” Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/sagnik1511/car-insurance-data.
[11] .”scikit-learn. ”MinMaxScaler.” scikit-learn, 2021. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html.
[12] ”H2O.ai. ”H2O Distributed Random Forest (DRF).” H2O.ai, n.d. [Online]. Available: https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/drf.html.
[13] ”XGBoost Documentation.” XGBoost, n.d. [Online]. Available: https://xgboost.readthedocs.io/en/stable/.
[14] ”scikit-learn. ”sklearn.metrics.classification_report.” scikit-learn, 2021. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_