Project Report

# Insurance Fraud Detection


*Project Report submitted on the fulfilment of the requirements of Post graduate Diploma in Big data Analytics*

Authors:

1. Gharge Rohit Vishwas 220960925006
2. Hajare Tanmay Yogesh 220960925007
3. Hanan P T 220960925008
4. Harshada Pardip Patil 220960925009
5. Jadhav Akshay Sanjay 220960925010
6. Jagdale Pankaj Kalidas 220960925011

Co-Ordinator: Prof. Roopa Panicker

STDC

CDAC Thiruvananthapuram

Trivandrum, Kerala 695581

# CONTENT

# 1. Abstract

Insurance fraud is a significant problem for the insurance industry, resulting in substantial financial losses. Fraudulent claims can take many forms, from exaggerating the value of a claim to filing a claim for a loss that did not occur. To combat insurance fraud, insurance companies are using advanced analytics and machine learning algorithms to identify fraudulent claims.

The insurance fraud detection problem is about building a model that can accurately detect fraudulent insurance claims based on historical data. The goal is to create a system that can flag suspicious claims and minimize losses caused by fraudulent activities.

The objective of this project is to develop a machine learning model for vehicle insurance fraud detection utilizing several classification techniques. The model was trained using Kaggle dataset. Using Information Gain model features were extracted. Matplotlib and Seaborn libraries in Python were used for visualization. The model was built utilizing 8 different classification models such as Decision Tree, Random Forest, Logistic Regression, Support Vector Machines, Naive Bayes, XGBoost, K-Nearest Neighbors and Artificial Neural Network.

Dataset: Insurance fraud detection dataset from kaggle

**Key Words:** Fraud detection, Machine Learning, Data Analytics, Insurance Company, fraudulent claims, Information Gain model, classification models.

**Methodology:**

1. Data collection
2. Data pre-processing
3. Feature engineering
4. Model selection
5. Model training
6. Model evaluation
7. Model prediction

# 2. Introduction

Insurance fraud is a significant problem for the insurance industry, leading to billions of dollars in losses each year. Machine learning has emerged as a powerful tool for detecting and preventing insurance fraud. In this context, insurance fraud detection involves developing and deploying machine learning models that can analyze large amounts of data and identify suspicious patterns or behaviors that suggest fraudulent activity.

The process of developing a machine learning model for insurance fraud detection typically involves several steps. First, historical data is collected and analyzed to identify patterns and trends associated with fraudulent claims. This data can include information about claimants, policyholders, and the types of claims being filed. Next, this data is used to train machine learning models that can identify these patterns and flag suspicious claims for further investigation.

There are several different types of machine learning models that can be used for insurance fraud detection, including supervised learning models, unsupervised learning models, and reinforcement learning models. Supervised learning models are trained on labeled data, which means that the data is already categorized as either fraudulent or legitimate. Unsupervised learning models, on the other hand, are trained on unlabeled data and must identify patterns and anomalies on their own. Reinforcement learning models use trial and error to learn which actions lead to the best outcomes, and can be used to develop strategies for preventing fraud.

Once a machine learning model has been trained, it can be deployed in a production environment where it can automatically analyze incoming claims and flag any that are suspicious. This can help insurance companies to detect and prevent fraud more quickly and efficiently than traditional manual methods. Overall, the use of machine learning for insurance fraud detection has the potential to save insurance companies millions of dollars each year while also improving the overall customer experience by reducing the number of fraudulent claims that are paid out.

# 3. Literature Survey

Abhijeet Urunkar et al [1] this paper discussed implement machine learning algorithms to build model to label and classify claim. Also, to study comparative study of all machine learning algorithms used for classification using confusion matrix in term soft accuracy, precision, recall etc,

Riya Roy [2] chose a sample of more than 500 data to compare the soft accuracy, precision, recall, and other metrics of all machine learning methods used for classification using confusion matrices. Following that, the data are separated into training and testing sets. here we can see, when compared to other algorithms, decision tree and random forest algorithms perform better than naive bayes.

Hritik Kalra[3] The proposed work offers a system for automatically detecting fraud without human intervention that uses policy information as input to quickly anticipate whether a claim is legitimate or fraudulent. For making predictions, they combined the XGBClassifier and SVMClassifier models, greatly enhancing the accuracy and precision of the model.
J48 has the highest accuracy among the other classifiers, according to Dilkhaz Y. Mohammed's[4] assessment of five classifiers based on precision, recall, F-measure, and mcc. This does not necessarily imply that it does well in other tests.All of the findings show that the Random Forest classifier performs effectively. The overall effectiveness of Bayes Net is fairly comparable to the outcomes of Nave Bayes. All other classifiers are more accurate than a random tree.

This article by Shrutee Phadke [5] Before introducing the cleaned data to the machine learning method, clustering is implemented. The best fitting model is then chosen after performing hyper parameter adjustment to increase the model's effectiveness. Finally, the best fitted model determines whether the supplied assertions are false or true.
Rama Devi Burri et all [6] presented several machine learning techniques to analysis insurance claims efficiently. They also mentioned three ways to transform machine learning techniques into insurance industry. Additionally, they specified different challenges in implementing machine learning.

Shivani Waghade [7] reviewed frauds in healthcare insurance industry including types of healthcare frauds and types as well as sources of healthcare data. She also reviewed techniques for detecting frauds such as machine learning.

Sunita Mall et all [8] proposed a study to identify important triggers of fraud and to predict the fraudulent behavior of customers using those identified triggers. They used statistical techniques to identify and predict the triggers.

Najmeddine Dhieb et all [9] identifies fraudulent claims in the auto industry by using approaches based on an extreme gradient boosting algorithm known as XGBoost

# 4. Proposed System

The dataset used for the model building is InsuranceData.csv (Source:https://www.kaggle.com) which contains 1000 rows and 39 rows. The objective was to test different models on the dataset once it is obtained and cleaned.

## A. Dataset pre-processing

- The cells that are marked '?' in columns 'collision_type', 'proprty_damage' and police_report_available' were replaced by most frequently occurring values in these columns.
- The columns that were not needed for further procedures were dropped. They are - 'policy_number','policy_bind_date','policy_state','insured_zip','incident_location','incident_date','incident_state','incident_city','insured_hobbies','auto_make','auto_model','auto_year'.
- Since there were categorical variables that were necessary for the model building, they had to be converted into numerical variables. Firstly we manually mapped the variables before endonding them into numerical value in the following way:

| **Variables** | **Mapped Values** |
|---|---|
| 'policy_csl' | 100/300' : 1, '250/500' : 2.5 ,'500/1000':5 |
| 'insured_education_level' | 'JD' : 1, 'High School' : 2,'College':3,'Masters':4,'Associate':5,'MD':6,'PhD':7 |
| 'incident_severity | 'Trivial Damage' : 1, 'Minor Damage' : 2,'Major Damage':3,'Total Loss':4 |
| 'insured_sex' | 'FEMALE' : 0, 'MALE' : 1 |

| | |
|---|---|
| 'property_damage' | 'NO' : 0, 'YES' : 1 |
| 'police_report_available' | ['police_report_available'].map({'NO' : 0, 'YES' : 1 |
| 'fraud_reported' | 'N' : 0, 'Y' : 1 |

- Five columns ('insured_occupation', 'insured_relatioship', 'incident_type', 'collission_type" and 'authorities_contacted') were converted into numerical values using auto-encoding.
- After converting 12 categorical features, there were 33 features. Further, categorical(33) and numerical columns(15) combined into one dataframe.
- Features and target were separated in 'x' and 'y' variables. 'Fraud_reported" column was fixed as target and the remaining columns were used for features.

| | months_as_customer | age | policy_number | policy_bind_date | policy_state | policy_csl | policy_deductable | policy_annual_premium | umbrella_limit | insured_zip |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 328 | 48 | 521585 | 10/17/2014 | OH | 250/500 | 1000 | 1406.91 | 0 | 466132 |
| **1** | 228 | 42 | 342868 | 6/27/2006 | IN | 250/500 | 2000 | 1197.22 | 5000000 | 468176 |
| **2** | 134 | 29 | 687698 | 9/6/2000 | OH | 100/300 | 2000 | 1413.14 | 5000000 | 430632 |
| **3** | 256 | 41 | 227811 | 5/25/1990 | IL | 250/500 | 2000 | 1415.74 | 6000000 | 608117 |
| **4** | 228 | 44 | 367455 | 6/6/2014 | IL | 500/1000 | 1000 | 1583.91 | 6000000 | 610706 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | .. |
| **995** | 3 | 38 | 941851 | 7/16/1991 | OH | 500/1000 | 1000 | 1310.80 | 0 | 431289 |
| **996** | 285 | 41 | 186934 | 1/5/2014 | IL | 100/300 | 1000 | 1436.79 | 0 | 608177 |
| **997** | 130 | 34 | 918516 | 2/17/2003 | OH | 250/500 | 500 | 1383.49 | 3000000 | 442797 |
| **998** | 458 | 62 | 533940 | 11/18/2011 | IL | 500/1000 | 2000 | 1356.92 | 5000000 | 441714 |
| **999** | 456 | 60 | 556080 | 11/11/1996 | OH | 250/500 | 1000 | 766.19 | 0 | 612260 |

1000 rows × 39 columns

```
In [5]:
# list of columns[12] not necessary for prediction
cols_to_drop=['policy_number','policy_bind_date','policy_state','insured_zip','incident_location','incident_date','incident_

In [6]:
# dropping the unnecessary columns
df.drop(columns=cols_to_drop,inplace=True)
```

```
[ ] df["collision_type"].unique()

    array(['Side Collision', '?', 'Rear Collision', 'Front Collision'],
          dtype=object)

[ ] df["policy_csl"].unique()

    array(['250/500', '100/300', '500/1000'], dtype=object)

[ ] df["insured_sex"].unique()

    array(['MALE', 'FEMALE'], dtype=object)

[ ] df["insured_education_level"].unique()

    array(['MD', 'PhD', 'Associate', 'Masters', 'High School', 'College',
           'JD'], dtype=object)
```

```
[ ] df["collision_type"].value_counts()["Side Collision"]

    276

[ ] df["collision_type"].value_counts()["Rear Collision"]

    292

[ ] df["collision_type"].value_counts()["Front Collision"]

    254

[ ] # Replace ? with null value
    df['collision_type'] = df['collision_type'].replace('?','Rear Collision')
```

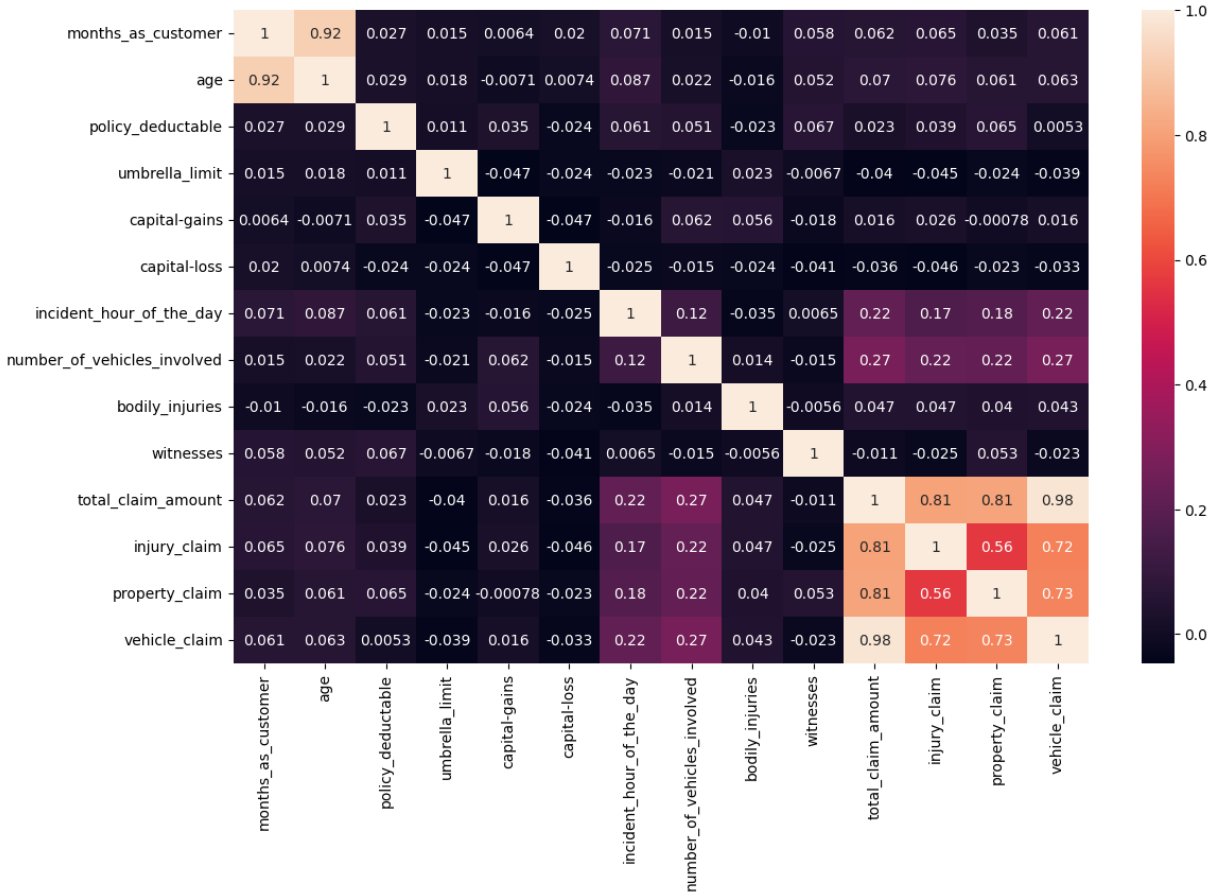
```
[ ] cat_df = df.select_dtypes(include=['object']).copy()

[ ] cat_df.columns

    Index(['policy_csl', 'insured_sex', 'insured_education_level',
           'insured_occupation', 'insured_relationship', 'incident_type',
           'collision_type', 'incident_severity', 'authorities_contacted',
           'property_damage', 'police_report_available', 'fraud_reported'],
          dtype='object')
```

```
[ ]  # combining the Numerical and categorical dataframes to get the final dataset
     final_df=pd.concat([num_df,cat_df], axis=1)
```

```
[ ]  num_df = df.select_dtypes(include=['int64']).copy()

[ ]  num_df.columns

     Index(['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'],
           dtype='object')
```

```
[ ]  # custom mapping for encoding [7]
     cat_df['policy_csl'] = cat_df['policy_csl'].map({'100/300' : 1, '250/500' : 2.5 ,'500/1000':5})
     cat_df['insured_education_level'] = cat_df['insured_education_level'].map({'JD' : 1, 'High School' : 2,'College':3,'Masters':4,'Associate':5,'MD':6,'PhD':7})
     cat_df['incident_severity'] = cat_df['incident_severity'].map({'Trivial Damage' : 1, 'Minor Damage' : 2,'Major Damage':3,'Total Loss':4})
     cat_df['insured_sex'] = cat_df['insured_sex'].map({'FEMALE' : 0, 'MALE' : 1})
     cat_df['property_damage'] = cat_df['property_damage'].map({'NO' : 0, 'YES' : 1})
     cat_df['police_report_available'] = cat_df['police_report_available'].map({'NO' : 0, 'YES' : 1})
     cat_df['fraud_reported'] = cat_df['fraud_reported'].map({'N' : 0, 'Y' : 1})

[ ]  # auto encoding of categorical variables [5]
     for col in cat_df.drop(columns=['policy_csl','insured_education_level','incident_severity','insured_sex','property_damage','police_report_available','fraud_repor
         cat_df= pd.get_dummies(cat_df, columns=[col], prefix = [col], drop_first=True)
```

```
[ ]  # separating the feature and target columns
     x=final_df.drop('fraud_reported',axis=1)
     y=final_df['fraud_reported']
```




A heatmap was used to find collinearity and multicollinearity between independent variables.

It was found that ‘age’ and ‘months_as_customer’ has high correlation (0.92) between them. Also ‘total_claim_amount’ had high correlation with ‘injury_claim (0.81)’, ‘property_claim’ (0.81) and ‘vehicle_claim’(0.98). Hence ‘age’ and ‘total_claim_amount” columns were dropped from the dataframe.

```
[ ]  # splitting the data into training and test set
     from sklearn.model_selection import train_test_split
     train_x,test_x,train_y,test_y=train_test_split(x,y, random_state=355 )
```

For model training the data was split into testing and training sets using sklearn library with random state =355 ( by default, data split was 75% for training and 25% for testing).

Data scaling is an important preprocessing step in machine learning that involves transforming input data to have a certain range or distribution. The goal of data scaling is to improve the performance of machine learning models by ensuring that features have a similar

scale and distribution.Data complexity was reduced by scaling data using StandardScaler.

```
# Scaling the numeric values in the dataset

from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
```

```
scaled_data=scaler.fit_transform(train_x)
scaled_train= pd.DataFrame(data=scaled_data, columns=train_x.columns,index=train_x.index)
scaled_train.shape
```

```
(750, 45)
```

## B. Feature Extraction:

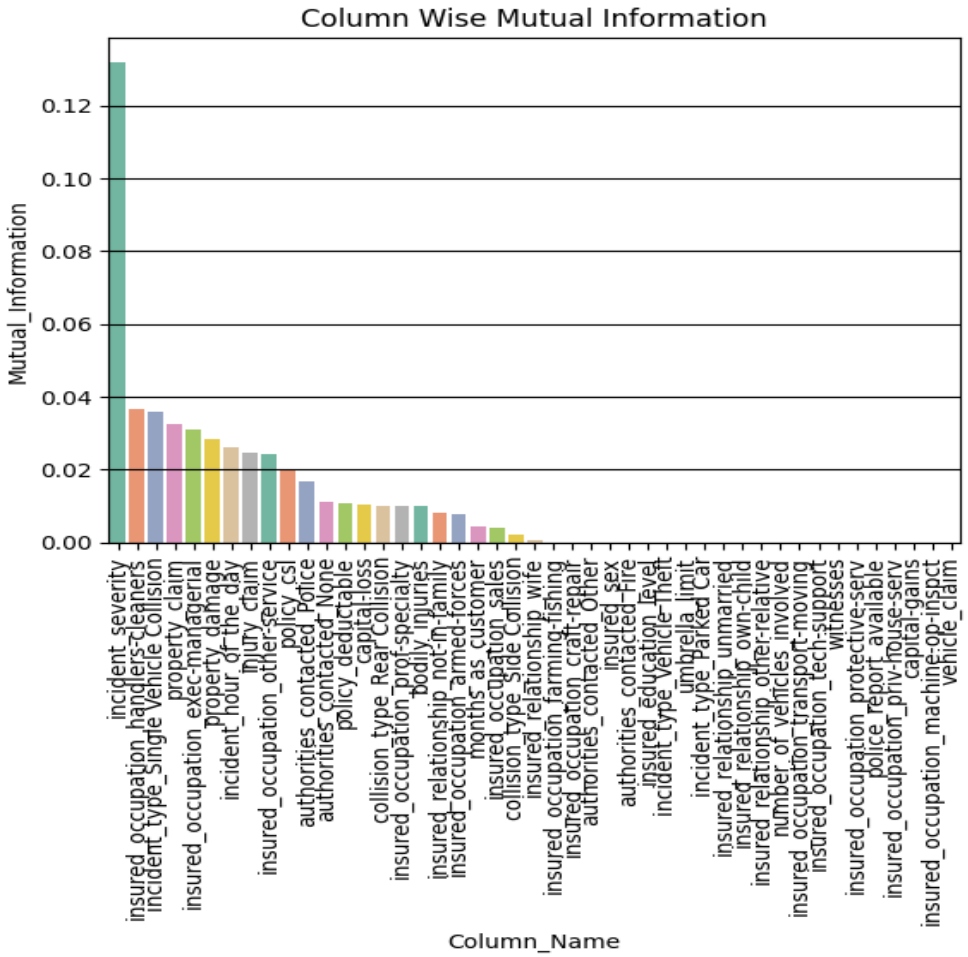
### Feature Selection:

For extracting the features, the Information Gain model was used. Information gain explains the amount of information gained about a random variable or signal from observing another random variable  A feature with a high information gain has a large impact on reducing the entropy of the target variable, and is therefore considered more useful for predicting the target variable. The result was plotted using matplotlib. Only 13 features that had high information gain were selected for further model building procedures. The 13 feature are - ‘'incident_severit','bodily_injuries','insured_occupation_sales','property_claim','insured_occupation_farming-fishing','injury_claim','insured_occupation_tech-support','policy_deductable','capital-loss','incident_type_Parked Car','insured_occupation_machine-op-inspct','collision_type_Rear Collision','insured_relationship_wife','insured_relationship_other-relative'.

```
from sklearn.feature_selection import mutual_info_classif
info_gain=mutual_info_classif(scaled_train,train_y)
info_gain
```

```
array([0.00432096, 0.01085626, 0.        , 0.        , 0.01031766,
       0.02614776, 0.        , 0.01007415, 0.        , 0.02459776,
       0.03255477, 0.        , 0.01980623, 0.        , 0.        ,
       0.13192187, 0.02854935, 0.        , 0.00790165, 0.        ,
       0.03114859, 0.        , 0.0365513 , 0.        , 0.02434167,
       0.        , 0.01016746, 0.        , 0.00411235, 0.        ,
       0.        , 0.00826459, 0.        , 0.        , 0.        ,
       0.00060779, 0.        , 0.03586287, 0.        , 0.01023752,
       0.00200664, 0.        , 0.01125722, 0.        , 0.01697193])
```

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.barplot(data=gain1,x="Column_Name",y="Mutual_Information",palette="Set2",
            order=gain1.sort_values("Mutual_Information",ascending = False).Column_Name)
plt.xticks(rotation=90)
plt.title("Column Wise Mutual Information")
plt.grid(color="black",axis="y")
plt.xticks(rotation=90)
plt.show()
```




```
[ ]  from sklearn.feature_selection import SelectKBest
     eleven_col=SelectKBest(mutual_info_classif,k=10)
     eleven_col.fit(scaled_train,train_y)
     train_x.columns[eleven_col.get_support()]
```

```
Index(['injury_claim', 'property_claim', 'policy_csl',
       'insured_education_level', 'incident_severity', 'property_damage',
       'insured_occupation_priv-house-serv',
       'insured_occupation_transport-moving', 'insured_relationship_wife',
       'collision_type_Rear Collision'],
      dtype='object')
```

## C. Models used

A total of eight different models were tried to find the model that best fits. They are - Classification using Decision Tree, Random Forest, Logistic Regression,Support Vector Machines, Naive Bayes, XGBoost, K-Nearest Neighbours and Artificial Neural Network.

1) Logistic Regression Classifier:

   Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

```
[ ]  from sklearn.linear_model import LogisticRegression
     model_Logi=LogisticRegression()
     model_Logi.fit(Scaled_fs_train_data,train_y)
     y_Predict_Logi=model_Logi.predict(Scaled_fs_test_data)
     y_Predict_Logi
```

2) Support Vector Classifier:

   The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane.

```
[ ]  # first using the Support vector classifier for model training
     from sklearn.svm import SVC
     sv_classifier=SVC()
```

```
     sv_classifier.fit(Scaled_fs_train_data, train_y)
     #y_pred=sv_classifier.predict([[0,0,6240,0,6240,0,2000,0,0,0,0,0,0]])
     y_pred=sv_classifier.predict(Scaled_fs_test_data)
     y_pred
```

3) K-Nearest Neighbours:

The K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.

```
[ ]  from sklearn.neighbors import KNeighborsClassifier
     classifier= KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2 )
     classifier.fit(Scaled_fs_train_data, train_y)
     #y_predict_knn=classifier.predict([[1,0,5720,0,11440,0,500,-24300,0,0,0,0,1]])
     y_predict_knn=classifier.predict(Scaled_fs_test_data)
     y_predict_knn
```

4) Naive Bayes:

The Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

```
[ ]  from sklearn.naive_bayes import GaussianNB
     classifier = GaussianNB()
     classifier.fit(Scaled_fs_train_data, train_y)
     y_predict_NB=classifier.predict(Scaled_fs_test_data)
```

5) Decision Tree Classifier:

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

For this dataset, ‘entropy’ was fixed as criterion and random state, 0.

```
[ ]  from sklearn.tree import DecisionTreeClassifier
     DT_classifier= DecisionTreeClassifier(criterion='entropy', random_state=0)
     DT_classifier.fit(Scaled_fs_train_data, train_y)
     y_predict_DT=DT_classifier.predict(Scaled_fs_test_data)
```

6) Random Forest Classifier:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Number of estimators in our model was fixed as 100.

```
[ ]  from sklearn.ensemble import RandomForestClassifier
     model_rf=RandomForestClassifier(n_estimators=100)
     model_rf.fit(Scaled_fs_train_data,train_y)
     y_pred_RF=model_rf.predict(Scaled_fs_test_data)
```

7) XGBoost Classifier:

XGBoost is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost stands for “Extreme Gradient Boosting”. It is efficient at handling missing values and has built-in support for parallel processing, making it possible to train models on large datasets in a reasonable amount of time.

```
[ ]  from xgboost import XGBClassifier

[ ]  xgb=XGBClassifier()

[ ]  y_pred = xgb.fit(Scaled_fs_train_data, train_y).predict(Scaled_fs_test_data)
```

8) Artificial Neural Network:

Artificial neural networks are relatively crude electronic networks of neurons based on the neural structure of the brain. They process records one at a time, and learn by comparing their classification of the record (i.e., largely arbitrary) with the known actual classification of the record. The errors from the initial classification of the first record is fed back into the network, and used to modify the network's algorithm for further iterations.

```
import numpy as np
import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

model_ann=Sequential()
model_ann.add(Dense(13,input_dim=13,activation="sigmoid"))
model_ann.add(Dense(13,activation="relu"))
model_ann.add(Dense(13,activation="relu"))
model_ann.add(Dense(13,activation="relu"))
model_ann.add(Dense(1,activation="sigmoid"))

model_ann.compile(optimizer="sgd",loss="binary_crossentropy",metrics=["accuracy"])

model_ann.fit(Scaled_fs_train_data,train_y,epochs=10,verbose=1)
```

# 5. Discussion and Results

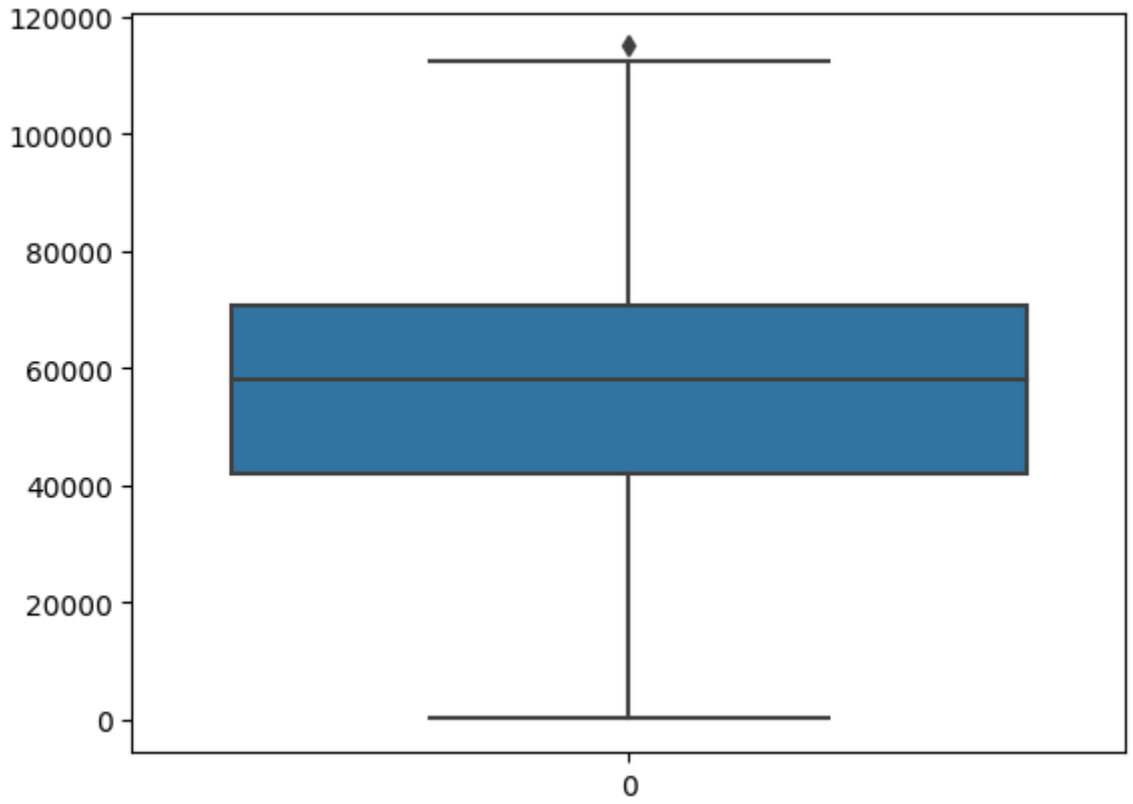
## A. Prediction Output:

```
[ ] # we perform 8 classification algorithm in that we got good accurancy for Logistics Regression, SVM, and ANN
    #here we use svm for  predication
    y_pred=sv_classifier.predict([[0,0,6240,0,6240,0,2000,0,0,0,0,0,0]])
    y_pred

    C:\Program Files\Python310\lib\site-packages\sklearn\base.py:409: UserWarning: X does not have valid feature names, but SVC was fitted with feature names
      warnings.warn(
    array([0], dtype=int64)
```

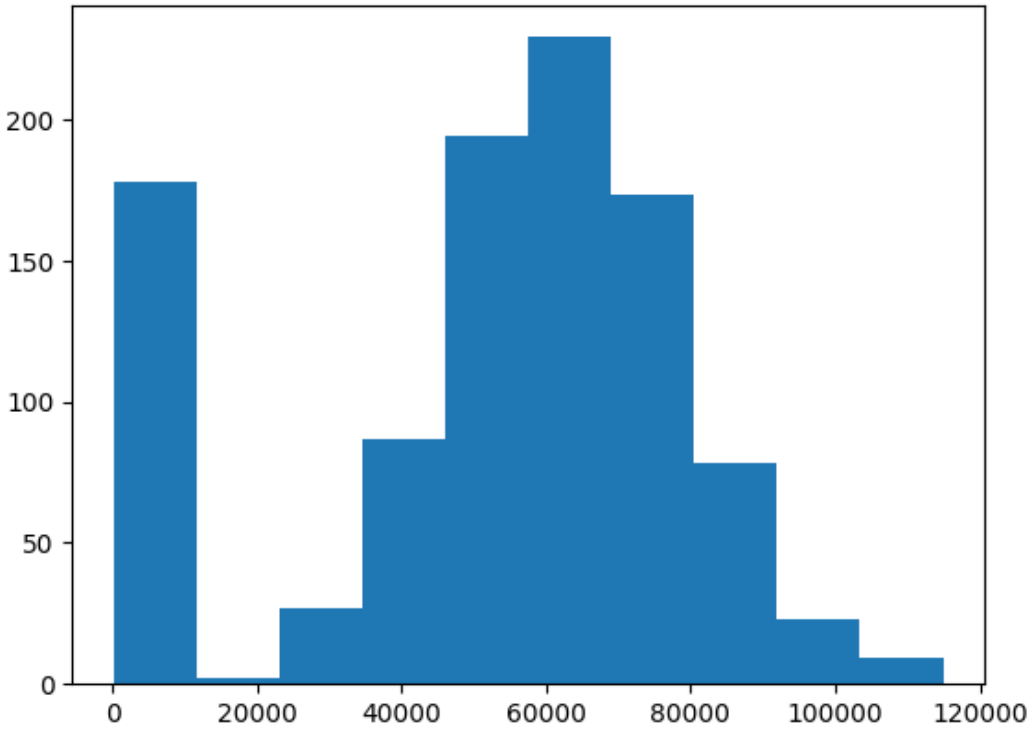
## B. Evaluation measures used

## Data Visualization:

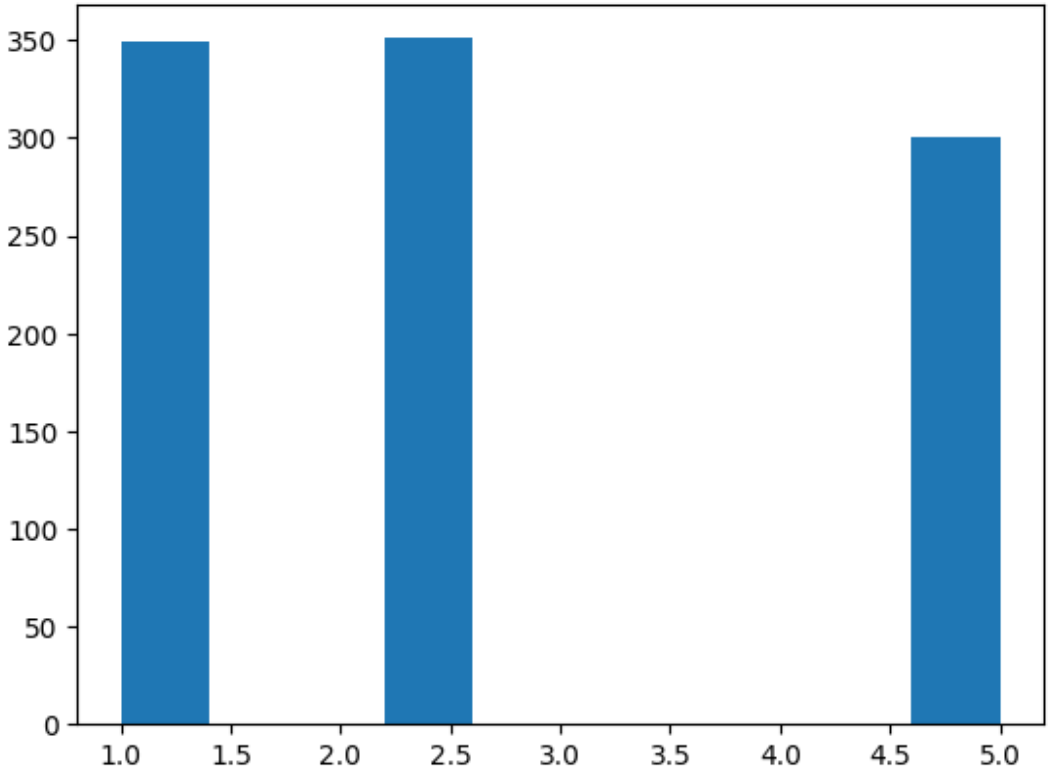
A box-plot was used to find the outliers in ‘Total claim amount data’. Only one outlier was found.



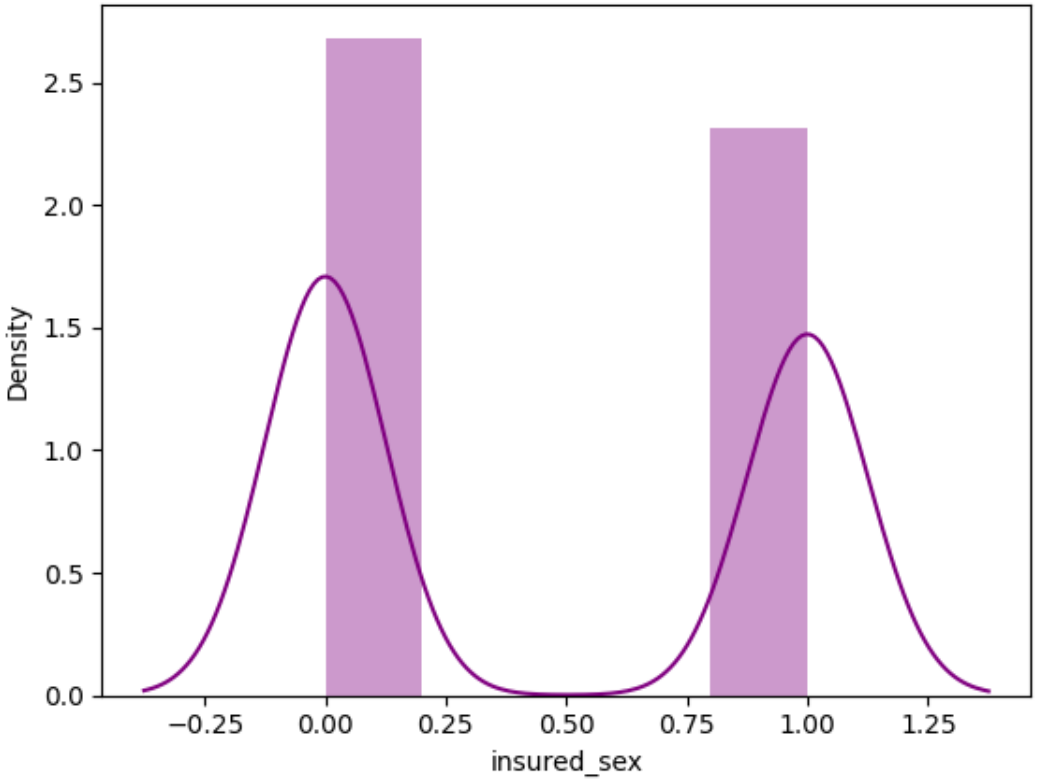
- Distribution Graph of Total claim



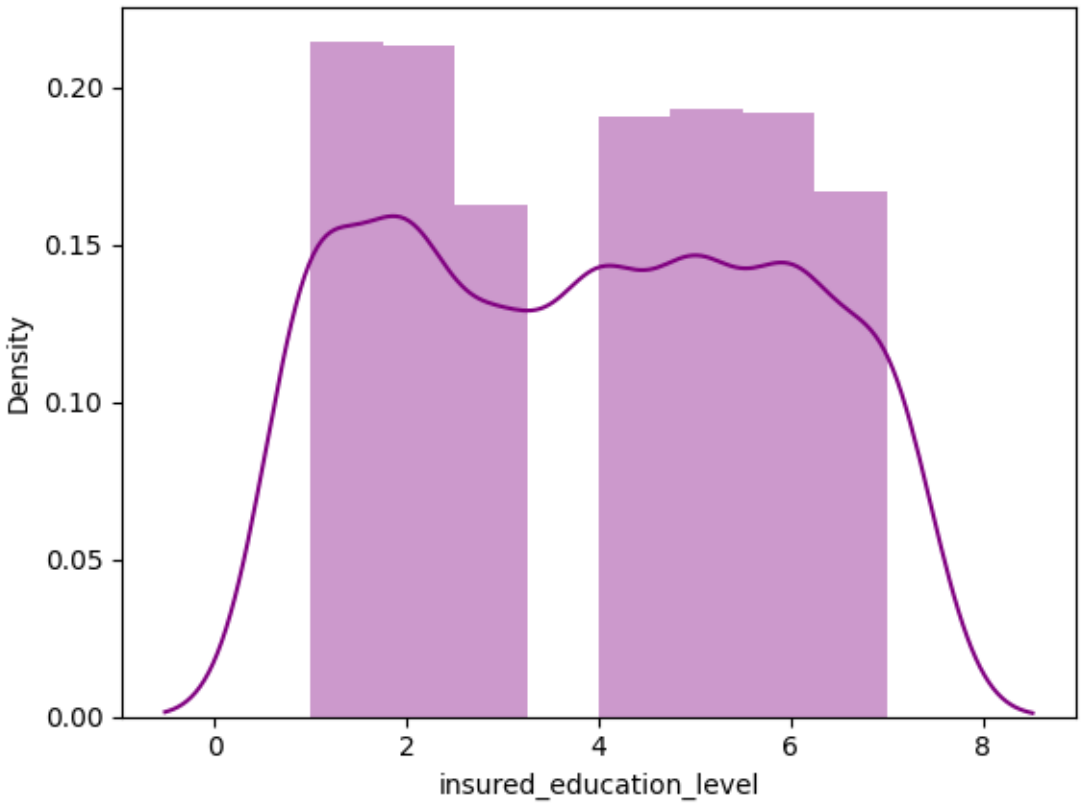

- Policy CSL is almost uniformly distributed.



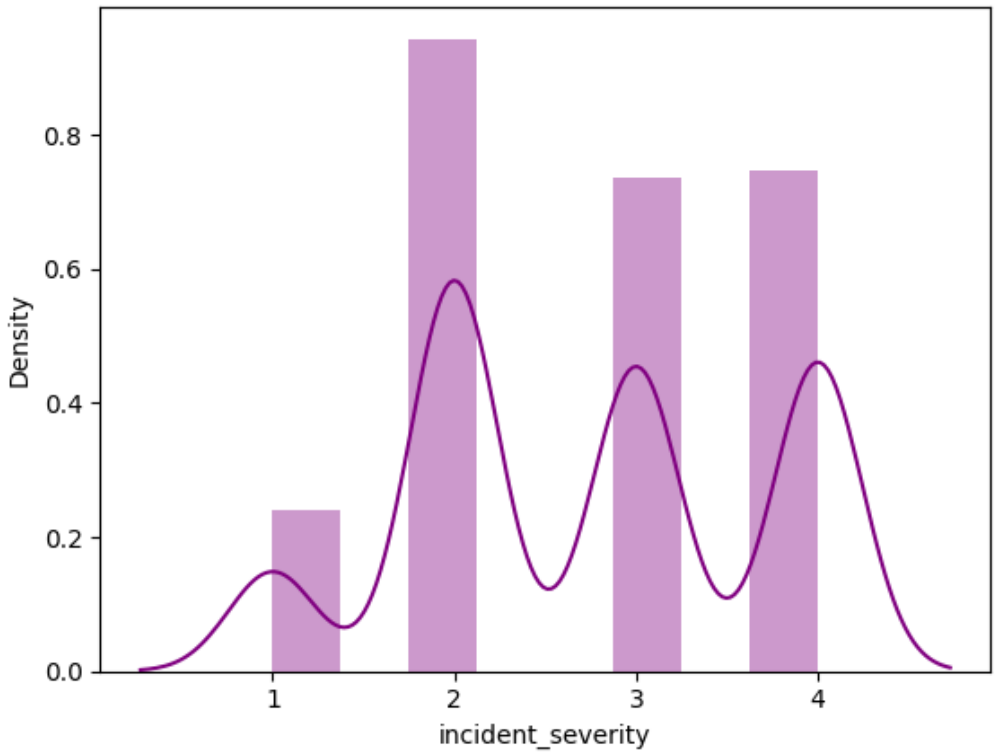

- Diagram showing the distribution of Insured sex.




- Image shows that for all the levels of education of insured, the distribution is similar.




- There are least claims for trivial incidents, most claims for minor incidents, and for major and Total loss incidents the claims are almost equal.

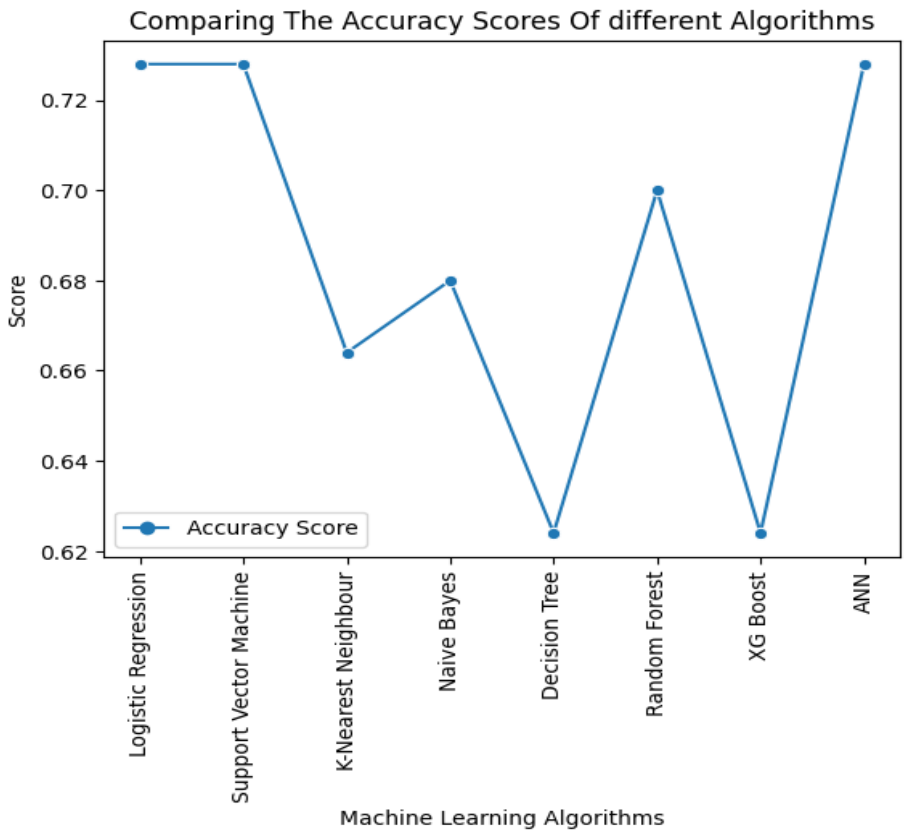


**Discussion and Results:**

However Different measures can be used to evaluate and analyze the model performance, accuracy was the measure used in this project.

| | Accuracy Score |
|---|---|
| **Logistic Regression** | 0.728 |
| **Support Vector Machine** | 0.728 |
| **K-Nearest Neighbour** | 0.664 |
| **Naive Bayes** | 0.680 |
| **Decision Tree** | 0.624 |
| **Random Forest** | 0.716 |
| **XG Boost** | 0.624 |
| **ANN** | 0.728 |

Among different classification models, Logistic Regression (0.728), Support Vector Machine (0.728) and Artificial Neural Network (0.728) were found to have the most accuracy.



# 6. Conclusion

We developed a solution for detecting auto insurance fraud using machine learning algorithms that can save insurance firms money and increase profitability while making sure that sincere policyholders are not penalised by false claims. By applying machine learning methods like Logistic Regression, Support Vector Machine, ANN, etc., we discovered that these algorithms had superior accuracy than other algorithms. In order to boost the speed of this system, we employed a few python modules, such as numpy, pandas, sklearn, and seaborn for data visualisation. After running operations on the dataset, we discovered that the majority of the claims relate to rather minor incidences. In the final outcome, we predicted if a given insurance claim is fraudulent or not. To enhance system performance, future work on upgrades and the development of commercial tools for large-scale applications may be necessary.

# 7. References


[1] Abhijeet Urunkar , Amruta Khot , Rashmi Bhat , Nandinee Mudegol “Fraud Detection and Analysis for Insurance Claim using Machine Learning”, 2022 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)

[2] Riya Roy, Thomas George K ,“Detecting Insurance Claims Fraud Using Machine Learning Techniques”, 2017 International Conference on circuits Power and Computing Technologies [ICCPCT]

[3] Hritik Kalra, Ranvir Singh, Dr.T. Senthil Kumar “Fraud Claims Detection in Insurance Using Machine Learning”, Journal of Pharmaceutical Negative Results ¦ Volume 13 ¦ Special Issue 3 ¦ 2022

[4] Dilkhaz Y. Mohammed, “Detection of Vehicle Insurance Claim Fraud A Fraud Detection Use-Case for the Vehicle Insurance Industry” International Journal of Progressive Sciences and Technologies (IJPSAT) ISSN: 2509-0119

[5] Shrutee Phadke, Princia Koli, Soham Shah, Shweta Sharma “Insurance Fraud Detection” International Research Journal of Engineering and Technology (IRJET)

[6] Rama Devi Burri , “Insurance Claim Analysis Using Machine Learning Algorithms”, IJITEE 2019

[7] Shivani S.Waghade, “A Comprehensive Study of Healthcare Fraud Detection based on Machine Learning” Int. J. Appl. Eng. Res. 2018

[8] Sunita Mall et all, “Management of Fraud: Case of an Indian Insurance Company” Accounting and Finace Research 2018

[9] Najmeddine Dhieb et all, “Extreme Gradient Boosting Machine Learning Algorithm for Safe Auto Insurance operations” LCVES, 2019