Auto Insurance Fraud Detection Using Machine Learning to Combat Fraudulent Claims

1. INTRODUCTION:

1.1 Overview

Automobile insurance fraud has become a significant issue for insurance companies, leading to substantial financial losses and increased premiums for policyholders. Fraudulent claims, including staged accidents, exaggerating the amount of damage, or faking injuries have become a big issue and it is important to find ways to detect and stop these fraudulent activities.

This project will utilize a dataset comprising previous insurance claims, including both legitimate and fraudulent cases. Various machine learning algorithms, such as logistic regression, k-nearest neighbour, decision tree, random forest, and support vector machine, will be implemented to build predictive models capable of distinguishing between fraudulent and non-fraudulent claims.

By addressing the rising concern of automobile insurance fraud through advanced technology, this project endeavours to make a substantial impact on the insurance industry, promoting trust, integrity, and cost savings for both insurance companies and policyholders.

1.2 Purpose

The objective of this project is to leverage machine learning techniques to develop a robust and automated system for detecting and preventing automobile insurance fraud, enabling insurance companies to take proactive measures to combat fraud and minimize financial losses.

2. LITERATURE SURVEY

2.1 Existing Problem

Existing works on automobile insurance fraud detection using machine learning techniques have made remarkable contributions to combat fraudulent claims and promote the integrity of the insurance industry. These works have explored various approaches and methodologies to identify and prevent fraudulent activities in automobile insurance. But there are certain instances where the model is unable to predict the fraudulent claims properly.

Previous works have applied a wide range of data mining (Phua et al. 2005), unsupervised (Nian et al. 2016), and supervised ML techniques (Nur Prasasti, Dhini, and Laoh 2020) to the problem of fraud detection. These techniques entail training light-weight classifiers, including Random Forests (Itri et al. 2019), SVMs (Muranda, Ali, and Shongwe 2020) and Naive Bayes (Roy and George 2017).

A study titled "Using Ethnography To Design a Mass Detection Tool (MDT) For The Early Discovery of Insurance Fraud" by Ormerod, Morley, Ball, Langley, and Spenser recommends the use of dynamic real-time Bayesian belief networks (BBNs), known as the Mass Detection Tool (MDT), which is then used by a rule generator known as the Suspicion Building Tool (SBT), for the early detection of potentially fraudulent claims. The results of the rule generator are used to fine-tune the BBN's weights, and claim handlers must keep up with new fraud schemes. This approach arose from an anthropological study of large insurance companies and claims adjusters who opposed manual detection of fraudulent actions by claims adjusters.

In the comparison research by Maes S, Tuyls K, Vanschoenwinkel B, and Manderick B, the STAGE algorithm is used for Bayesian Belief Networks (BBNs) and the BP algorithm is used for Artificial Neural Networks (ANNs). Comparative results illustrate that while BBNs are slower when applied to new instances, they are more accurate and train much faster.

The hot spot methodology by Williams G and Huang Z. “Mining the Knowledge Mine: The Hot Spots Methodology for Mining Large Real World Databases” applies a three step process: cluster detection using the kmeans algorithm, decision tree rule induction using the C4.5 algorithm, and rule evaluation using domain expertise, statistical summaries, and visualisation tools.

2.2 Proposed Solution

The objective is to forecast whether or not a claim for auto insurance is fraudulent based tabular data submitted with the claim. Our solution is implementing a model using XGBoost**.** Once trained, the XGBoost model can be used to predict the likelihood of fraud for new, unseen claims. By inputting the relevant features of a claim into the model, it can provide a fraud probability or classification output indicating the likelihood of fraudulent activity. The model's predictions can then be used to prioritize and investigate suspicious claims further, allowing insurance companies to take appropriate actions to combat fraud. It significantly leveraged the ability to identify fraudulent and non-fraudulent claims of the given dataset.

3. THEORITICAL ANALYSIS

3.1 Block Diagram

The Block diagram is a high-level diagram that depicts the overall structure or components of the machine learning model developed and deployed.

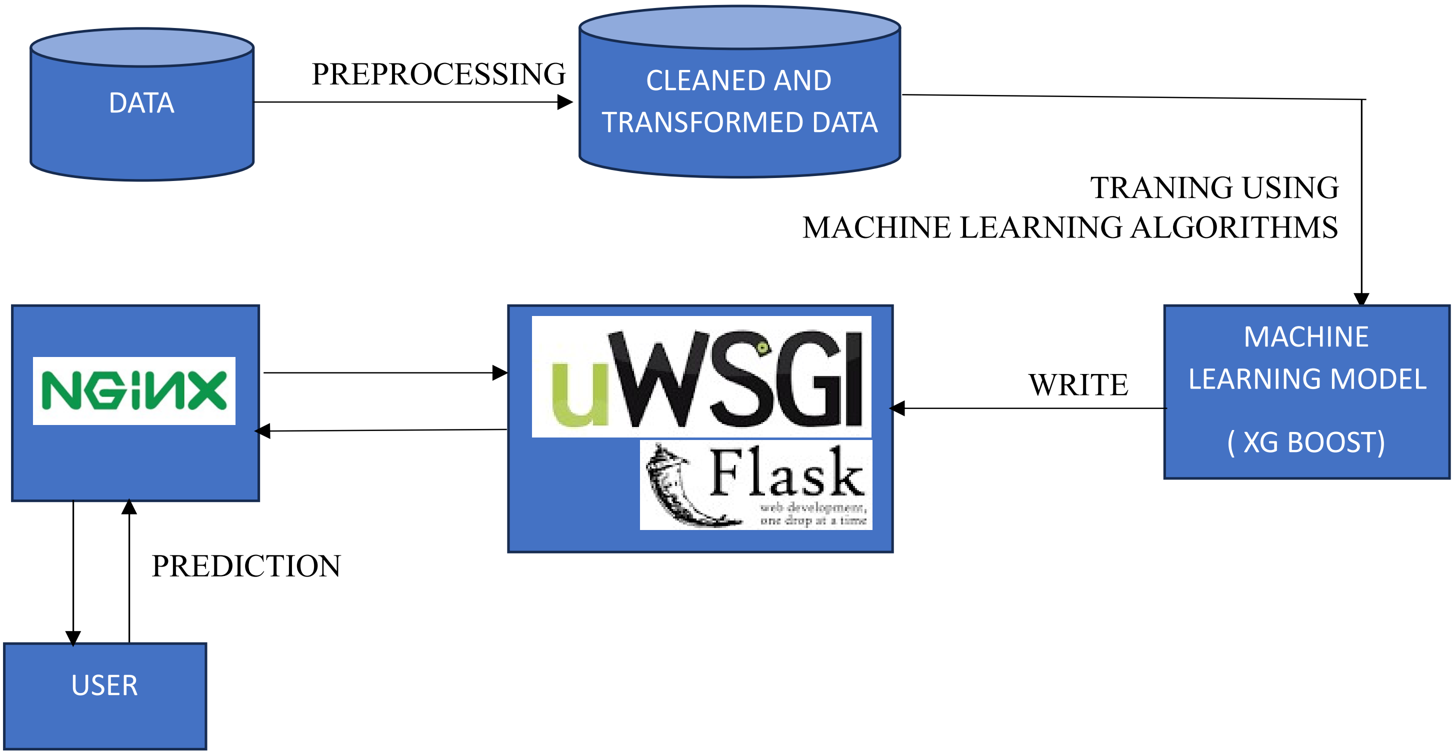


Fig. 1 Block diagram

3.2 Hardware/Software designing

Hardware Requirements:

Computer: A reasonably powerful computer system with sufficient processing power and memory is required to handle the data processing and machine learning tasks involved in the project.

Storage: Adequate storage capacity is needed to store the datasets, models, and other relevant files associated with the project.

Server or Cloud Infrastructure (optional): For larger-scale implementations or real-time processing, a server infrastructure or cloud-based platform may be required to handle the computational workload and ensure scalability.

Software Requirements:

Programming Languages: Knowledge and proficiency in programming languages commonly used in machine learning, such as Python or R, are essential. Additionally, familiarity with libraries and frameworks like scikit-learn will be beneficial.

Data Processing and Analysis Tools: Software tools for data preprocessing, cleaning, and exploratory data analysis are required. Examples include Pandas, NumPy for handling structured data.

Machine Learning Frameworks: Utilize machine learning frameworks like scikit-learn to develop and train machine learning models for fraud detection.

Data Visualization Tools: Tools for visualizing data and model outputs can aid in gaining insights and communicating findings effectively. Popular options include Matplotlib, Seaborn.

Integrated Development Environment (IDE): An IDE such as Jupyter Notebook, or Flask can provide a user-friendly coding environment for development and experimentation.

4. EXPERIMENTAL INVESTIGATIONS

During the experimental investigations of the "Automobile Insurance Fraud Detection Using Machine Learning to Combat Fraudulent Claims" project, several analyses and investigations are conducted to evaluate the proposed solution and its effectiveness in detecting fraudulent claims. Here are some examples of the analysis and investigations that may be performed:

Data Analysis: The collected insurance claims dataset is analysed to gain insights into the characteristics and patterns of fraudulent claims. From a bar between fraud reported and incident type we got to know that most of the incidents are of Multi Vehicle collision.

Feature Importance Analysis: A feature importance analysis is conducted to determine the significance of different features in predicting fraud. This analysis helps identify the most relevant features that contribute the most to fraud detection. Here correlation analysis is employed for removing certain features which are less important. We found that some features like policy_number, insured_zip, policy_bind_date, incident_date, incident_location are unnecessary as they do not contribute for training the model.

Model Performance Evaluation: The performance of the trained machine learning models is assessed using appropriate evaluation metrics. In this project accuracy metrics help measure how well the models can distinguish between fraudulent and non-fraudulent claims. XGBoost has an accuracy of 87%.

Comparative Analysis: Different machine learning algorithms, such as logistic regression, k-nearest neighbours, decision trees, random forests, support vector machines and XGBoost have been evaluated and compared to identify the most effective algorithm for fraud detection. XGBoost has got the highest accuracy.

5. FLOWCHART

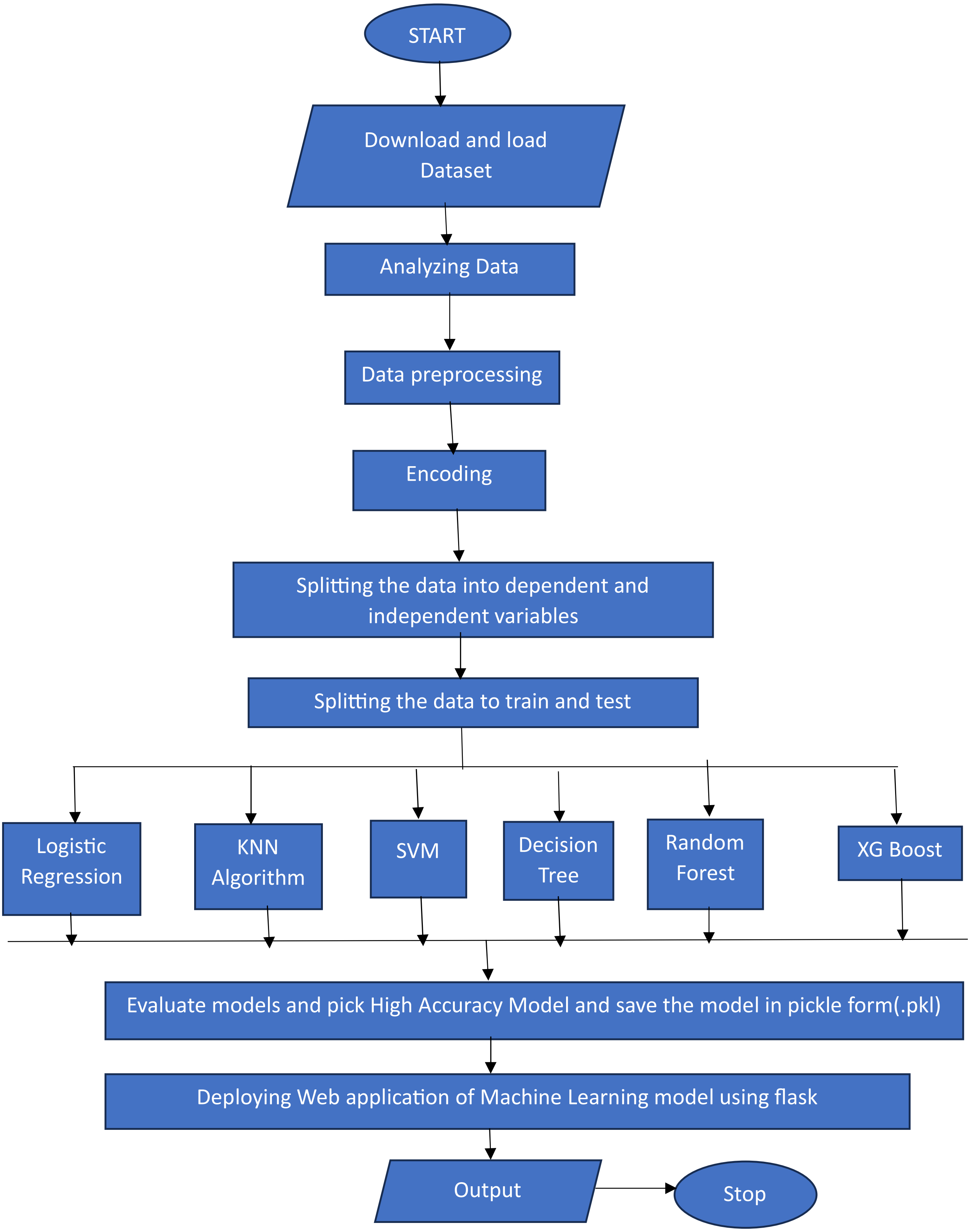

START
Download and load Dataset
Analyzing Data
Data preprocessing
Encoding
Splitting the data into dependent and independent variables
Splitting the data to train and test
Logistic Regression
KNN Algorithm
SVM
Decision Tree
Random Forest
XG Boost
Evaluate models and pick High Accuracy Model and save the model in pickle form(.pkl)
Deploying Web application of Machine Learning model using flask
Output
Stop

Fig.2 Flowchart

6. RESULT

Different measures can be used to evaluate and analyse the Model Performance. Some of the measures used in this project are:

Model	Precision	Recall	F1 Score
Logistic Regression	58	58	58
KNN Algorithm	66	66	66
SVM	51	50	45
Decision Tree	84	84	84
Random Forest	87	87	87
XG Boost	89	88	88

Table-I Precision Analysis

In this analysis, several factors were known which can facilitate to spot for associate degree correct distinction between fraud transactions and non-fraudulent transactions that helps to predict the presence of fraud within the given transactions. Once completely different input datasets are used the Machine Learning models performed at variable performance levels.

F1 score is a measure of accuracy and recall that takes both into account. XG Boost has an F1 score of 88%, which is the highest of all the models.

Overall, XG Boost is the best model for evaluation from the above data. It has the highest precision, recall, and F1 score.

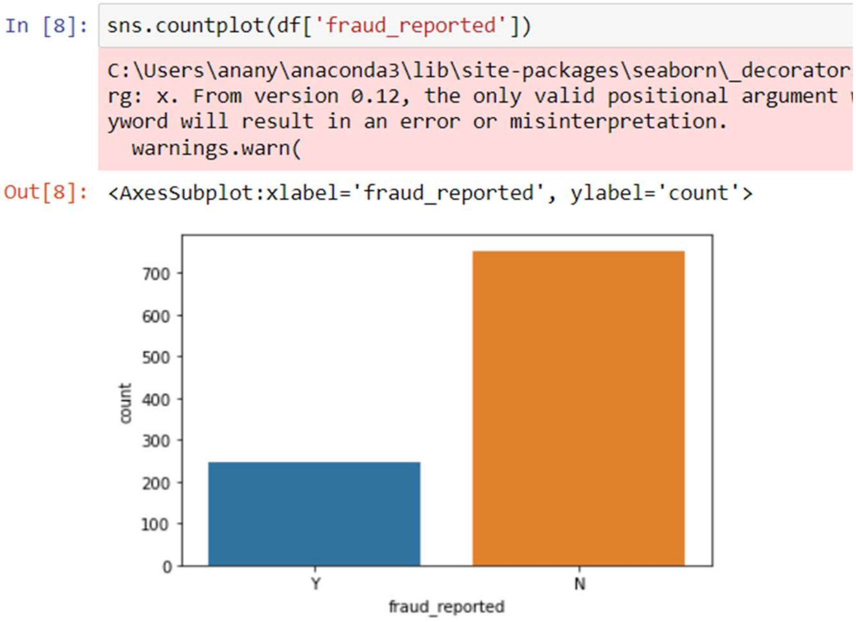
6.1 Training and Testing

Based on the analysis, we explored various factors that can help identify the distinction between fraudulent and non-fraudulent transactions. The goal was to predict the presence of fraud in the given transactions. Multiple Machine Learning models were trained and tested using different input datasets, resulting in varying levels of performance.

To determine the effectiveness of the models, we considered the accuracy. Higher accuracy indicates better model performance. Based on this evaluation XGBoost model emerged as the top performer.

However, it cannot be assumed that order of prophetical quality would be replicated and might differ for alternative datasets. Once discovered it's complete that within the dataset samples, the models with datasets that are feature made, performs well. Obtained during training and testing phase. Depending on various features the trends are analysed and hence used to decide the best model among the various Machine Learning classifiers.

The following figures gives some of the graphical representation of the results.



```
In [8]: sns.countplot(df['fraud_reported'])

C:\Users\anany\anaconda3\lib\site-packages\seaborn\_decorator
rg: x. From version 0.12, the only valid positional argument 
yword will result in an error or misinterpretation.
  warnings.warn(

Out[8]: <AxesSubplot:xlabel='fraud_reported', ylabel='count'>
```

Fig. 3 Count plot of Fraud reported

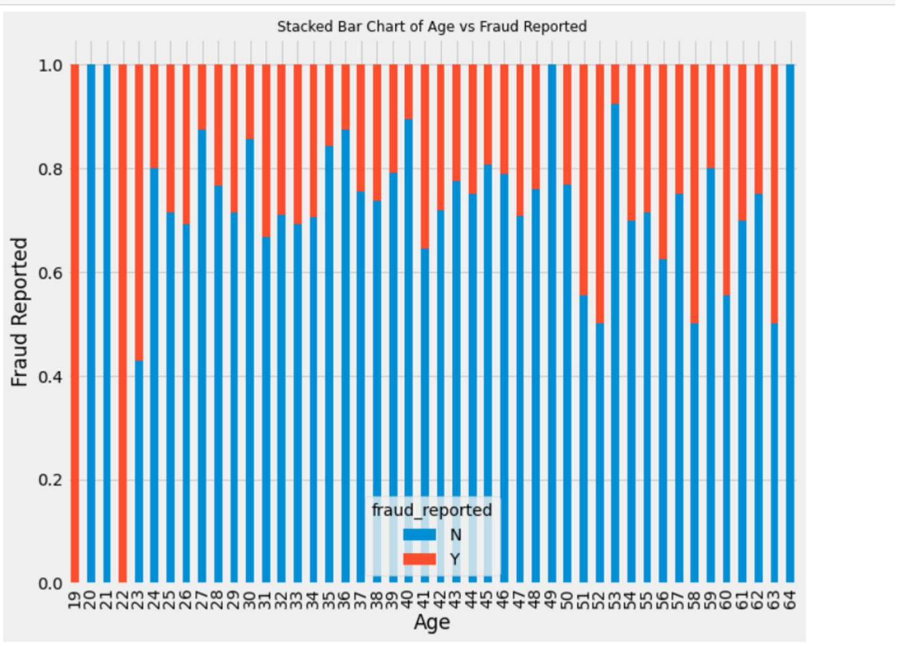
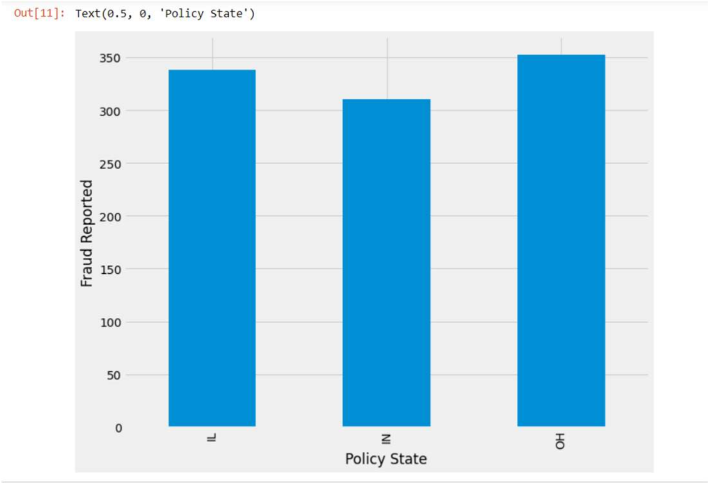


Fig.4 Stacked Bar chart of Age vs Fraud Reported

The resulting plot is a stacked bar chart that represents the proportion of 'fraud_reported' (Yes/No) for different age groups. Each bar represents an age group, and the height of the stacked segments represents the proportion of 'fraud_reported' within each age group.

Please note that the actual data and resulting plot will vary based on the content of the 'age' and 'fraud_reported' columns in your DataFrame.



```
Out[11]: Text(0.5, 0, 'Policy State')
```

Fig.5 Bar chart between Policy state and fraud reported

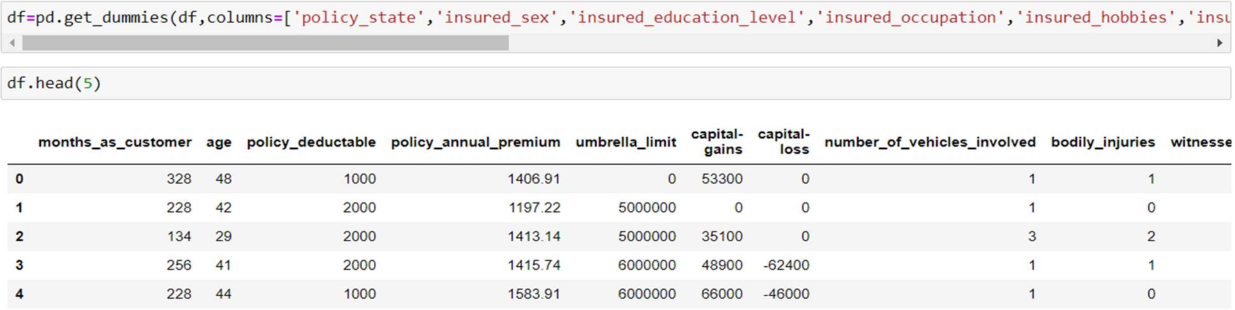
A bar plot that visualizes the distribution of fraud reported cases across different policy states. This visualization allows you to compare the number of frauds reported cases in each policy state and identify any variations or patterns in the data. The y-axis represents the count of fraud reported cases, and the x-axis represents the policy states.

6.2 Encoding

In the analysis, we encountered categorical values in our dataset that needed to be converted into numerical form.

One Hot encoding

One Hot Encoding allows for a more expressive representation of categorical data.



```
df=pd.get_dummies(df,columns=['policy_state','insured_sex','insured_education_level','insured_occupation','insured_hobbies','insu
```

```
df.head(5)
```

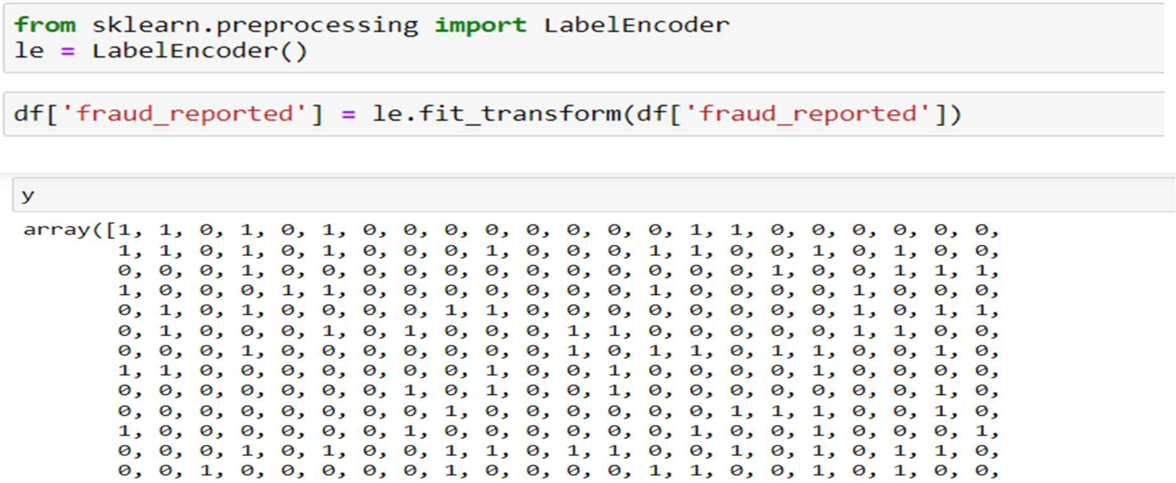
	months_as_customer	age	policy_deductable	policy_annual_premium	umbrella_limit	capital-gains	capital-loss	number_of_vehicles_involved	bodily_injuries	witnesse
0	328	48	1000	1406.91	0	53300	0	1	1	
1	228	42	2000	1197.22	5000000	0	0	1	0	
2	134	29	2000	1413.14	5000000	35100	0	3	2	
3	256	41	2000	1415.74	6000000	48900	-62400	1	1	
4	228	44	1000	1583.91	6000000	66000	-46000	1	0	

Fig.6 One hot encoded data

Label Encoding

Label Encoding is an alternative approach that assigns unique numerical labels to each category.

Label Encoding is required because many machine learning algorithms are unable to directly work with categorical data and may not produce expected results. By converting categorical values into numerical labels, we enable the algorithms to interpret and analyze the data effectively.



```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

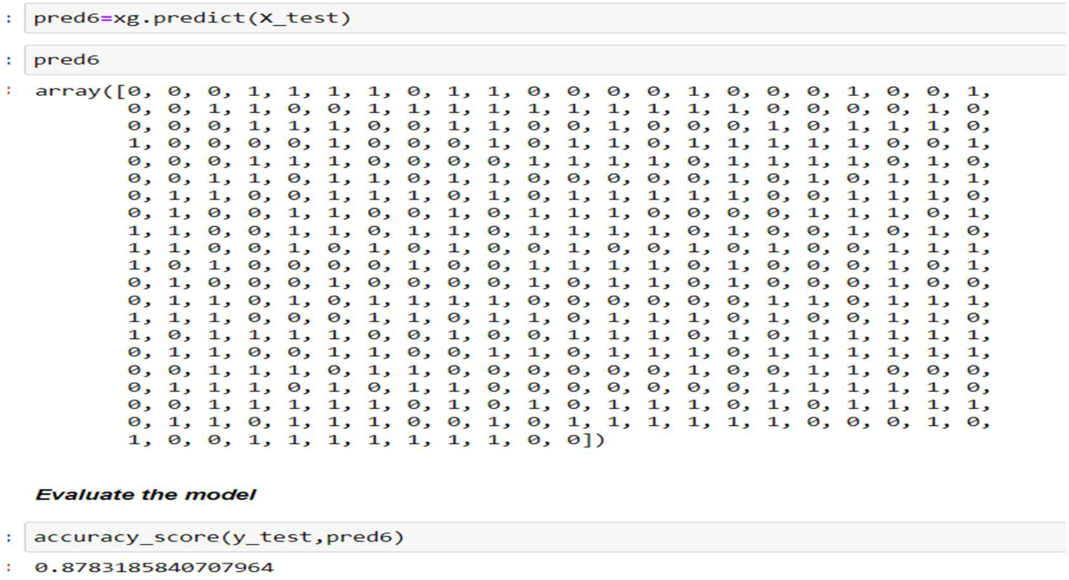
```
df['fraud_reported'] = le.fit_transform(df['fraud_reported'])
```

```
y
```

```
array([1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
       1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1,
       1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1,
       0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0,
```

Fig.7 Label Encoded data

6.3 Output of Model



```
: pred6=xg.predict(X_test)
```

```
: pred6
```

```
: array([0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
         0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0,
         0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0,
         1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1,
         0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0,
         0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1,
         0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0,
         0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1,
         1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0,
         1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1,
         1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1,
         0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0,
         0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1,
         1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0,
         1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1,
         0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
         0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0,
         0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0,
         0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1,
         0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0,
         1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0])
```

Evaluate the model

```
: accuracy_score(y_test,pred6)
```

```
: 0.8783185840707964
```

Input: Auto Insurance fraud detection dataset with 1506 records after performing SMOTE, out of which 1054 are training set and 452 are testing set with 30 columns each.
Output: ‘0’ for no fraud and ‘1’ for fraud detected. Accuracy of XGBoost model.

7. ADVANTAGES AND DISADVANTAGES

Advantages of proposed solution

i. **Improved Fraud Detection**: The proposed solution leverages machine learning techniques, which can effectively detect patterns and anomalies in large datasets. This can lead to improved fraud detection rates compared to traditional rule-based or statistical approaches.

ii. **Real-time Detection:** By implementing the proposed solution in a real-time or near real-time system, fraudulent claims can be identified promptly, allowing for timely intervention and prevention of financial losses.

iii. **Enhanced Accuracy:** Machine learning algorithms can learn from historical data and continuously improve their accuracy over time. This adaptive nature of the models can enhance the accuracy of fraud detection, reducing false positives and false negatives.

iv. **Automation and Efficiency:** The proposed solution automates the fraud detection process, reducing the manual effort required for reviewing claims. This increases the efficiency of the insurance claim processing system and enables faster decision-making.

v. **Cost Savings:** By accurately detecting fraudulent claims, insurance companies can save significant costs associated with fraudulent activities. This can lead to improved profitability and potential cost savings for both insurance companies and policyholders.

Disadvantages of Proposed solution

i. **Data Availability and Quality:** The effectiveness of the proposed solution heavily relies on the availability and quality of data. Inaccurate or incomplete data may impact the performance of the fraud detection system and lead to erroneous results.

ii. **Model Complexity and Interpretability:** Machine learning models, especially complex ones like neural networks, may lack interpretability. Understanding and explaining the reasoning behind model predictions can be challenging, which could be a drawback in certain scenarios where interpretability is crucial.

iii. **Overfitting and Generalization:** Machine learning models have the potential to overfit the training data, leading to poor generalization on unseen data. Careful model selection, feature engineering, and cross-validation techniques are necessary to address this issue.

iv. **Evolving Fraud Techniques:** Fraudsters constantly adapt their techniques to circumvent detection systems. The proposed solution may face challenges in keeping up with evolving fraudulent activities, requiring regular updates and retraining of the models.

v. **Ethical Considerations:** The use of personal data for fraud detection must be done responsibly, ensuring compliance with privacy regulations and ethical considerations. Balancing the need for fraud detection with privacy concerns is essential to maintain public trust.

8. APPLICATIONS

This project has several potential applications where it can be useful. Some of these applications include:

i. **Insurance Companies:** Insurance companies can benefit from this project by implementing the fraud detection system to identify and prevent fraudulent claims. By accurately detecting fraudulent activities, insurance companies can reduce financial losses, improve their profitability, and maintain fair premiums for honest policyholders.

ii. **Policyholders:** The project is beneficial for policyholders as it helps ensure that insurance premiums remain fair and reasonable. By detecting and preventing fraudulent claims, the costs associated with fraudulent activities are minimized, which can lead to more stable and affordable insurance premiums for policyholders.

iii. **Law Enforcement Agencies:** Law enforcement agencies involved in investigating insurance fraud can leverage the project's fraud detection system to enhance their investigative processes. The system can provide them with valuable insights and evidence to support their efforts in combating fraudulent activities in the automobile insurance industry.

iv. **Regulatory Authorities:** Regulatory authorities responsible for overseeing the insurance industry can use the project's fraud detection system to monitor the compliance of insurance companies and ensure fair practices. By identifying fraudulent claims and taking appropriate actions, regulatory authorities can maintain the integrity and transparency of the insurance market.

v. **Fraud Investigation Agencies:** Dedicated fraud investigation agencies can utilize the project's fraud detection system as a valuable tool in their investigations. The system can help them identify potential cases of fraud, prioritize investigations, and gather evidence for legal proceedings.

vi. **Insurance Fraud Research:** The findings and methodologies of the project can contribute to the field of insurance fraud research. The project can serve as a foundation for further studies and advancements in developing more sophisticated fraud detection techniques and strategies.

9. CONCLUSION

The successful implementation of the machine learning-based fraud detection system is expected to significantly enhance the efficiency and accuracy of fraud identification in automobile insurance claims. By performing the comparative analysis, we found out that XGBoost model gives the highest accuracy. By automating the detection process, insurance companies can reduce the time and resources required for manual investigations while effectively deterring potential fraudsters. This project aims to contribute to the overall reduction of fraudulent claims, leading to fairer premiums for policyholders and improved financial stability for insurance providers.

10.FUTURE SCOPE

Some potential future directions for this project include:

i. **Incorporating Unstructured Data Analysis:** Expanding the scope of the project to include analysis of unstructured data sources, such as social media, text documents, and images, can provide additional context and enhance the accuracy of fraud detection. Natural language processing, sentiment analysis, and image recognition techniques can be applied to extract relevant information from unstructured data sources.

ii. **Big Data Analytics:** As the volume of data in the insurance industry continues to grow, leveraging big data analytics techniques becomes crucial. Integrating big data technologies and analytics platforms can enable more comprehensive analysis, real-time processing, and the extraction of valuable insights for fraud detection.

iii. **Integration with IoT:** The integration of Internet of Things (IoT) devices with vehicles can provide real-time data streams and additional indicators for fraud detection. Analysing data from connected devices, such as vehicle sensors, can help identify suspicious patterns and anomalies associated with fraudulent activities.

11.BIBLIOGRAPHY

1. Burri, R.D., Burri, R., Bojja, R.R., & Buruga, S.R. (2019). Insurance Claim Analysis Using Machine Learning Algorithms.
2. E. B. Belhadji, G. Dionne, and F. Tarkhani, “A Model for the Detection of Insurance Fraud,” Geneva Pap. Risk Insur. Issues Pract., vol. 25, no. 4, pp. 517– 538, 2000, doi: 10.1111/1468-0440.00080.
3. Gondalia, D., Gurav, 0., Joshi, A., Joshi, A., & Selvan, S., (2022). Automobile Insurance Claim Fraud Detection.
4. Maes S, Tuyls K, Vanschoenwinkel B and Manderick B. “Credit Card Fraud Detection Using Bayesian and Neural Networks”, in Proceedings of the 1st International NAISO Congress on Neuro Fuzzy Technologies, Havana, Cuba, 2002
5. Ormerod T, Morley N, Ball L, Langley C and Spenser C. “Using Ethnography To Design a Mass Detection Tool (MDT) For The Early Discovery of Insurance Fraud”, in Proceedings of ACM CHI Conference, Florida, USA, 2003.
6. Williams G and Huang Z. “Mining the Knowledge Mine: The Hot Spots Methodology for Mining Large Real World Databases”, in Proceedings of the 10th Australian Joint Conference on Artificial Intelligence, Perth, Australia, 1997.

APPENDIX

Python Source Code(Jupyter Note book)

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('insurance_claims.csv')
df.head()
df.shape
df.tail()
df.info()
df.describe()
df['fraud_reported'].value_counts()
sns.countplot(df['fraud_reported'])
df.groupby('incident_state').fraud_reported.count().plot.bar(ylim=0)
plt.rcParams['figure.figsize'] = [10, 8]
ax= plt.style.use('fivethirtyeight')
table=pd.crosstab(df.age, df.fraud_reported)
table.div(table.sum(1).astype(float), axis=0).plot(kind='bar', stacked=True)
plt.title('Stacked Bar Chart of Age vs Fraud Reported', fontsize=12)
plt.xlabel('Age')
plt.ylabel('Fraud Reported')
plt.show()
plt.rcParams['figure.figsize'] = [10, 8]
df.groupby('policy_state').fraud_reported.count().plot.bar(ylim=0)
plt.ylabel('Fraud Reported')
plt.xlabel('Policy State')
df.groupby('incident_type').fraud_reported.count().plot.bar(ylim=0)
plt.ylabel('Fraud Reported')
plt.xlabel('Incident Type')
table=pd.crosstab(df.insured_education_level, df.fraud_reported)
table.div(table.sum(1).astype(float), axis=0).plot(kind='bar', stacked=True)
plt.title('Stacked Bar Chart of insured education vs Fraud reported', fontsize=12)
plt.xlabel('Insured Education Level')
plt.ylabel('Fraud Reported');
df = df.drop(columns = [ 'policy_number', 'policy_csl', 'insured_zip', 'policy_bind_date',
'incident_date', 'incident_location', 'auto_year', 'incident_hour_of_the_day', 'auto_model',
'_c39'])
df.shape
df.isnull().sum()
```

Flask codes (app.py, index.html, index.css)

app.py

```
from flask import Flask, render_template, request
app = Flask(__name__)# interface between my server and my application wsgi
import pickle
model = pickle.load(open(r'C:\Users\anany\Flask/model.pkl','rb'))
@app.route('/')#binds to an url
def helloworld():
    return render_template("index.html")
@app.route('/login', methods =['POST'])#binds to an url

def login():
    a= request.form["mac"]
    b= request.form["ag"]
    c= request.form["ps"]
    if (c=="oh"):
        c1,c2,c3 = 1,0,0
    if (c=="il"):
        c1,c2,c3 = 0,1,0
    if (c=="in"):
        c1,c2,c3 = 0,0,1
```

index.html

```
<html>
<head>
<link rel="stylesheet" href='../static/index.css'>
</head>
<body>
    <h1> Auto Insurance Fraud Detection</h1>
    <form action = "/login" method= "post">
    <p>Months as customer : <input type="text" name = "mac" /></p>
    <p>Age : <input type="text" name = "ag" /></p>

    <label for = "states">Choose the Policy state</label>
    <select name ="ps">
    <option Value = "oh">OH</option>
    <option Value = "il">IL</option>
```

```
   <option Value = "in">IN</option>
   </select>

   <p>Policy Deductable : <input type="text" name = "pd" /></p>
   <p>Policy Annual Premium : <input type="text" name = "pap" /></p>
   <p>Umbrella Limit : <input type="text" name = "ul" /></p>
```

index.css

```
body{
   background-color: rgb(40,181,167);
   padding-left: 600px;
   display: inline-flex;
   flex-direction: column;
   font-family: Verdana, Geneva, sans-serif;
}
h1{
   position: relative;
  padding-right: 50px;
  right: 100px;
}
input{
   border-radius: 3px;
   border-style: line;
   border-color: black;
}
```

Entire codes for the jupyter notebook, app.py, index.html, index.css are available in below drive link

https://drive.google.com/drive/u/0/folders/1AiGEOaPusC9ZCMe8eEyih2phx8oRDcE5