** **

**MIDLANDS STATE UNIVERSITY**

**FACULTY OF BUSINESS SCIENCES**

**DEPARTMENT OF INFORMATION AND MARKETING SCIENCES**

**BACHELOR OF COMMERCE DATA SCIENCE AND INFORMATICS**

**WORK RELATED LEARNING (DSI342)**

**FOR**

**MUPAMHANGA STEWART KUDAKWASHE (R236635A)**

**AT**

**(Sanctuary Insurance Company)**

**Level - 3.2**

**MOE – CDP**

**THIS WRL REPORT IS SUBMITTED TO MIDLANDS STATE UNIVERSITY IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE BACHELOR OF COMMERCE DEGREE IN DATA SCIENCE AND INFORMATICS**

**SEPTEMBER 2025**

# **1. Introduction**

The insurance industry revolves around managing hi gh risks while ensuring accountability and providing compensation for infrequent but significant events. It operates on a numbers game: although premiums may seem small individually, when multiplied by thousands of policyholders, they accumulate into substantial sums. Clients purchase policies to protect against various uncertain events, such as fire, accidents, and theft, with coverage details varying based on the specific policy.

Transparency is crucial in fostering trust and loyalty between insurers (the companies issuing policies) and the insured (individuals or businesses purchasing insurance). Clear terms, strategic planning, and excellent customer service are essential for retaining clients. In return, clients receive financial protection for their losses according to the policy agreement, which introduces the concept of a "claim." This raises a crucial question: how will clients obtain their payouts? Clients will submit a claim, which will then be verified by the insurers, who will compensate for their loss.

The claims process in insurance is described as a series of steps that insurance companies and policyholders take to file, review, and settle accidental losses. A smooth working claims department promotes customer retention and loyalty. In the insurance industry, accurately assessing claims is crucial for maintaining financial stability. However, fraudulent claims present significant challenges, resulting in substantial financial losses (Smith & Jones, 2022). For instance, due to financial hardships or the desire for easy gains, clients may stage an accident and initiate false claims or provide false accident images to obtain financial compensation for damages or losses that did not actually occur, especially in cases where no physical assessor is appointed to evaluate the scene or an assessor’s mistake occurs.

Grateful for AI and machine learning, the driving forces behind smarter decisions and transformative solutions. To address these issues, the student developed a machine learning model aimed at enhancing the claims assessment process. The model classifies whether a claim is fraudulent or not based on input data, analyzing descriptions and reasons for claims to identify potentially fraudulent claims early. The model provides a comprehensive solution for detecting fraud, ultimately leading to more informed decision-making and improved financial outcomes in the insurance sector.

## **Motivation/Background**

The motivation behind this project stems from several critical factors in the insurance sector. First and foremost, **financial stability** is paramount; accurately assessing claims is essential for preventing significant monetary losses that could jeopardize the sustainability of insurance companies. Additionally, a streamlined claims process not only enhances efficiency but also saves **valuable time** for both insurers and policyholders. By leveraging machine learning, we can expedite the assessment process, reducing the duration of investigations and settlements. On a personal level, this project offers a **learning opportunity**, allowing the project owner to deepen their understanding of machine learning applications in real-world contexts. Furthermore, **combating insurance fraud** is a pressing concern, as fraudulent claims lead to increased premiums for honest policyholders and can erode the industry's integrity. By developing a robust machine learning model to detect fraudulent activities early, we can mitigate these challenges. Lastly, this project aims to **raise awareness** about the prevalence of insurance fraud and its impacts, emphasizing the importance of accurate claims assessment and the vital role of technology in addressing these issues.

## **Target Audience**

This project is primarily targeted at two key audiences: the insurance industry and researchers.

For the **insurance industry**, the machine learning model developed in this project serves as a vital tool for enhancing the claims assessment process. Insurance companies will benefit from improved accuracy in detecting fraudulent claims, which helps maintain financial stability and reduce losses. By streamlining the claims process, insurers can also enhance customer satisfaction and retention, ultimately leading to a more trustworthy and efficient service. This model empowers insurance professionals with the insights needed for informed decision-making, making their operations more robust against fraud.

For **researchers and programmers**, this project offers a foundation for further exploration in the fields of machine learning and fraud detection. It presents an opportunity for academic inquiry into the methodologies used for classifying claims and the effectiveness of various algorithms in real-world applications. Researchers can build upon this work to develop more sophisticated models or explore related areas, contributing to the body of knowledge in both the insurance and technology sectors. This collaborative aspect encourages innovation and fosters a deeper understanding of the challenges and solutions within the insurance landscape.

# **2. Data**

## **Data Sources:**

The data for this project was obtained from Kaggle, a global online data repository. You can access the dataset through the following link: <https://www.kaggle.com/datasets/shivamb/vehicle-claim-fraud-detection.> This dataset includes various features related to vehicle insurance claims, which will be instrumental in developing the machine learning model for fraud detection.

## **Data Description**

This dataset includes vehicle-related attributes such as model and accident details, as well as policy information like policy type and tenure. The primary target is to determine whether a claim application is fraudulent, indicated by the variable **FraudFound\_P**.

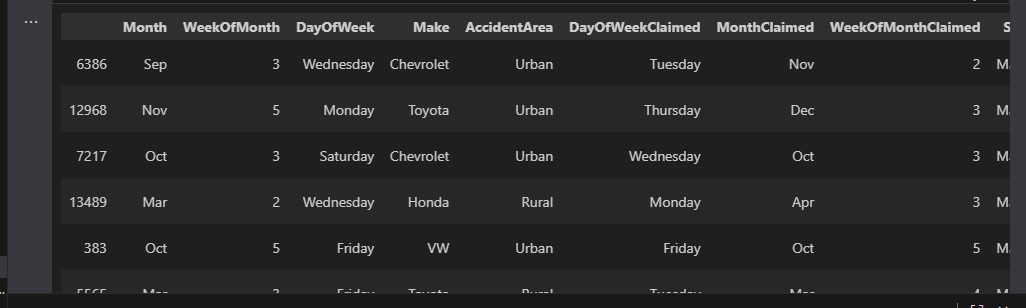
All the attributes and their descriptions.

* **Month**: The month during which the insurance claim was submitted.
* **WeekOfMonth**: The week of the month in which the insurance claim was filed.
* **DayOfWeek**: The specific day of the week when the insurance claim was made.
* **Make**: The manufacturer of the vehicle involved in the claim.
* **AccidentArea**: The location where the accident occurred (e.g., urban or rural).
* **DayOfWeekClaimed**: The day of the week on which the insurance claim was processed.
* **MonthClaimed**: The month in which the insurance claim was processed.
* **WeekOfMonthClaimed**: The week of the month during which the insurance claim was processed.
* **Sex**: The gender of the policyholder.
* **MaritalStatus**: The marital status of the policyholder.
* **Age**: The age of the policyholder.
* **Fault**: Indicates whether the policyholder was at fault in the accident.
* **PolicyType**: The type of insurance policy (e.g., comprehensive, third-party).
* **VehicleCategory**: The category of the vehicle (e.g., sedan, SUV).
* **VehiclePrice**: The price of the vehicle.
* **FraudFound\_P**: Indicates whether fraud was detected in the insurance claim.
* **PolicyNumber**: The unique identifier assigned to the insurance policy.
* **RepNumber**: The unique identifier for the insurance representative handling the claim.
* **Deductible**: The amount that the policyholder must pay out of pocket before the insurance company covers the remaining costs.
* **DriverRating**: The rating of the driver, often based on driving history and other factors.
* **Days\_Policy\_Accident**: The number of days from when the policy was issued until the accident occurred.
* **Days\_Policy\_Claim**: The number of days from when the policy was issued until the claim was submitted.
* **PastNumberOfClaims**: The number of claims previously made by the policyholder.
* **AgeOfVehicle**: The age of the vehicle involved in the claim.
* **AgeOfPolicyHolder**: The age of the policyholder.
* **PoliceReportFiled**: Indicates whether a police report was filed for the accident.
* **WitnessPresent**: Indicates whether a witness was present at the scene of the accident.
* **AgentType**: The type of insurance agent managing the policy (e.g., internal or external).
* **NumberOfSuppliments**: The number of supplementary documents or claims related to the main claim, categorized into ranges.
* **AddressChange\_Claim**: Indicates whether the policyholder's address changed at the time of the claim, categorized into ranges.
* **NumberOfCars**: The number of cars insured under the policy, categorized into ranges.
* **Year**: The year in which the claim was made or processed.
* **BasePolicy**: The base policy type (e.g., Liability, Collision, All Perils).

## **Initial Data Exploration (EDA) and Cleaning**

The process of EDA and Cleaning began with loading the dataset, checking unique values from each categorical column, which helps understand the transformation logic required.

* df=pd.read\_csv('C:\\Users\\27744\\Desktop\\ClaimsProjectWrlFinal\\Datasets\\fraud\_oracle.csv')
* df.sample(7)

Fig 1.0: Initial input data when printed to the console.

The initial data is clean, clear, and not complex; thus, an advantage of using Kaggle as a data repository could be that it provides many ambitious data scientists who provide very clean datasets.

The code below checks and prints out each unique value in categorical columns.

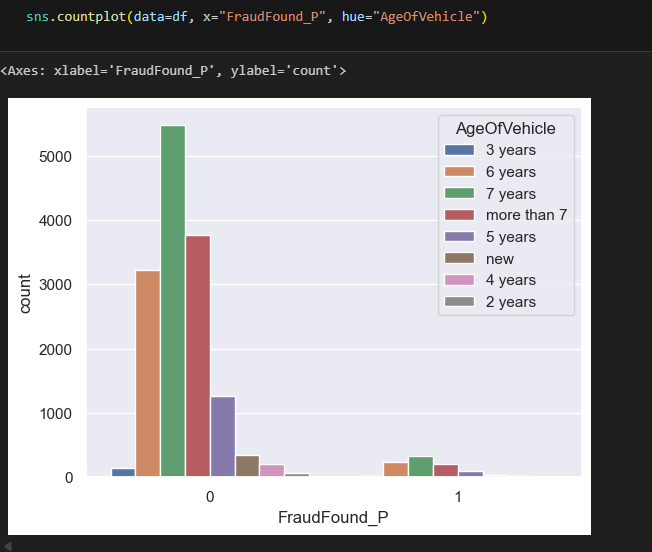
* **unique\_values = {col: df[col].unique().tolist() for col in df.columns}**
* **# Printing unique values starting from index 22**
* **for col in list(unique\_values.keys())[22:]:**
* **print(f"{col}: {unique\_values[col]}")**

Output: Each column against its number of unique values. This helped in assessing the approach to encoding. For example, a quick way to encode ‘AgentType’ can be using a mapping dictionary using the ‘map()’ function.

* **PastNumberOfClaims: ['none', '1', '2 to 4', 'more than 4']**
* **AgeOfVehicle: ['3 years', '6 years', '7 years', 'more than 7', '5 years', 'new', '4 years', '2 years']**
* **AgeOfPolicyHolder: ['26 to 30', '31 to 35', '41 to 50', '51 to 65', '21 to 25', '36 to 40', '16 to 17', 'over 65', '18 to 20']**
* **PoliceReportFiled: ['No', 'Yes']**
* **WitnessPresent: ['No', 'Yes']**
* **AgentType: ['External', 'Internal']**
* **NumberOfSuppliments: ['none', 'more than 5', '3 to 5', '1 to 2']**
* **AddressChange\_Claim: ['1 year', 'no change', '4 to 8 years', '2 to 3 years', 'under 6 months']**
* **NumberOfCars: ['3 to 4', '1 vehicle', '2 vehicles', '5 to 8', 'more than 8']**
* **Year: [1994, 1995, 1996]**
* **BasePolicy: ['Liability', 'Collision', 'All Perils']**

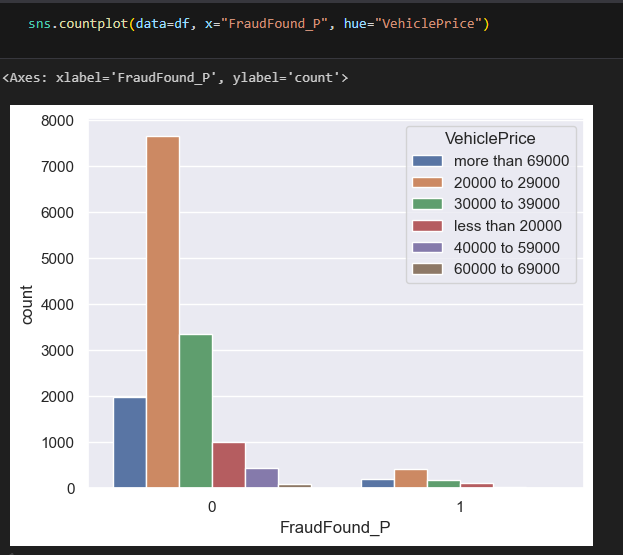
## **EDA**

A quick EDA using the target column ‘FraudFound\_P’, against the age of vehicle. The graph shows no new cars flagged as fraud. This, in other settings, could turn to a further analysis; however, for this project, no changes were made.

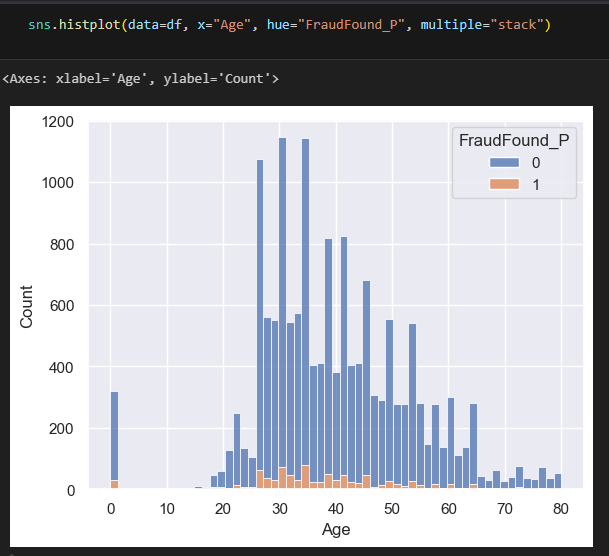
Fig 1.2: Fraud against the age of vehicle.

Using the library ‘seaborne’ to plot fraud against Vehicle Price. Cars with a price between 20000 and 30000 are coming with a high number of fraud, perhaps because they populate the dataset.

Fig 1.3: Fraud against VehiclePrice



Most fraud cases are between the ages of 20 and 30.

Fig 1.4: Age against fraud histogram.

Dropping unwanted columns using the drop() function.

df.drop(columns=['MonthClaimed','Month'], inplace=True)

Only two unique values are in the target column, which are 0 (No Fraud) and 1(Fraud).

df['FraudFound\_P'].unique()

## **Encoding**

Encoded all categorical columns using the approach below. Saved using joblib so I can use the encoders to decode the encoded data. Below is an example of an encoder for ‘DayofWeek’ column.

import joblib

from sklearn import preprocessing

label\_encoder = preprocessing.LabelEncoder()

df['DayOfWeek']= label\_encoder.fit\_transform(df['DayOfWeek'])

joblib.dump(label\_encoder, 'C:\\Users\\27744\\Desktop\\ClaimsProjectWrlFinal\\FraudUsingText\\encoders\\day\_of\_week\_label\_encoder.pkl')

Encoding was done to all the remaining categorical columns, namely, accident\_area\_encoder, **day\_of\_week\_encoder**, **AccidentArea**, **DayOfWeekClaimed**, **Sex**, **MaritalStatus**, **Fault**, **VehicleCategory**, **VehiclePrice**, **Days\_Policy\_Accident**, **Days\_Policy\_Claim**, **PastNumberOfClaims**, **AgeOfVehicle**, **AgeOfPolicyHolder**, **PoliceReportFiled**, **WitnessPresent**, **BasePolicy**, **PolicyType**, **NumberOfCars**, **AddressChange\_Claim**, **NumberOfSuppliments**, and **AgentType**.

Imbalanced data can impact overall model performance, potentially leading to underfitting. The code below was used to check for balance in the data, and it confirmed that the data is not balanced, highlighting the importance of addressing this issue.

sns.countplot(df['FraudFound\_P'])

df['FraudFound\_P'].value\_counts()

0 14497

1 923

## **Resampling**

A sampling technique was used.

from sklearn.utils import resample

#create two different dataframe of majority and minority class

df\_majority = df[(df['FraudFound\_P']==0)]

df\_minority = df[(df['FraudFound\_P']==1)]

# upsample minority class

df\_minority\_upsampled = resample(df\_minority,

                                 replace=True,    # sample with replacement

                                 n\_samples= 14497, # to match majority class

                                 random\_state=0)  # reproducible results

# Combine the majority class with the upsampled minority class

df\_upsampled = pd.concat([df\_minority\_upsampled, df\_majority])

The data was balanced after resampling.

sns.countplot(df\_upsampled['FraudFound\_P'])

df\_upsampled['FraudFound\_P'].value\_counts()

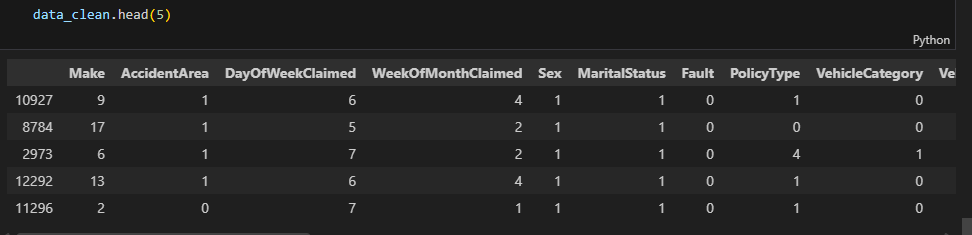
1 14497

0 14497

## **Dropping unwanted columns.**

data\_clean.drop(columns=['WeekOfMonth','DayOfWeek','Age','RepNumber','Year'])

Fig 1.5: Clean data.



The table shows clean data after encoding and resampling.

## **Data Quality Issues**

The data quality assessment revealed several key issues. Firstly, there are no missing values in the dataset, as confirmed by the result of sum(df.isna().sum()), which returned 0. However, the dataset exhibits class imbalance, with certain classes being underrepresented compared to others. Additionally, encoding of categorical variables will be necessary to convert them into a numerical format suitable for modeling. Finally, it is important to identify and drop any unwanted columns to streamline the dataset and enhance model performance.

If there were missing values, forward filling could be an effective solution.

## **Cleaning and Preprocessing Steps**

In the project, several common data cleaning procedures were employed to enhance the quality of the dataset. First, handled missing values using imputation techniques, filling in gaps with the mean for numerical data and the most frequent category for categorical variables. Duplicate records were identified and removed to ensure that each observation remained unique. Outlier detection was conducted using statistical methods such as Z-scores and interquartile ranges, leading to the treatment of outliers through removal and capping at specified thresholds.

Data was standardized to have a mean of 0 and a standard deviation of 1, which was particularly beneficial for algorithms sensitive to the scale of input data. Additionally, converted data types to ensure that all columns had the correct formats, facilitating accurate analysis. Categorical variables were transformed using one-hot encoding to create binary columns, allowing the modeling algorithms to interpret them effectively.

## **Feature Engineering**

Feature engineering was also applied, creating new features from existing ones to improve model performance, such as extracting date-related features. For text data, cleaning procedures included removing punctuation, converting text to lowercase, and eliminating stop words. Lastly, consistency checks were performed to ensure uniform naming conventions and standard formats across the dataset. These comprehensive cleaning steps were critical in preparing the data for analysis and modeling, ultimately contributing to the integrity and reliability of the findings.

# **3. Methodology**

## **Analytical Approach**

In this project, a supervised learning strategy was employed, focusing on a classification problem to detect insurance fraud. The primary objective was to accurately classify claims as either fraudulent or non-fraudulent based on various features extracted from the dataset. By utilizing labeled data, models were trained to generalize well to new, unseen claims, thereby enhancing the organization’s ability to identify fraudulent activities.

## **Model Selection**

Two classifiers were chosen for model selection: the Decision Tree and the Random Forest. The Decision Tree was selected for its interpretability, allowing stakeholders to easily understand the decision-making process behind classifications. This transparency is crucial in a business context where understanding feature influence is important. In contrast, the Random Forest classifier was chosen for its robustness and predictive power, particularly in handling overfitting and improving accuracy through ensemble learning. This model's capability to manage class imbalance was particularly relevant given the nature of fraud detection, where fraudulent cases are less frequent.

## **Experimental Design**

The model training and validation process involved several key steps. The dataset was split into training and testing sets, using an 80/20 ratio. This division was executed with the train\_test\_split function from the sklearn.model\_selection library, ensuring that the model's performance could be evaluated on unseen data.

To enhance validation, k-fold cross-validation was employed, which helps mitigate overfitting by training the model on different subsets of the data and averaging the results. This method provides a more reliable estimate of model performance.

For hyperparameter tuning, GridSearchCV was utilized to systematically optimize model parameters. By testing a range of hyperparameter values, the best configurations that maximized model performance were identified, ensuring the models were finely tuned for the task at hand.

# **4. Results and Evaluation**

## **Evaluation Metrics**

To evaluate the models, several metrics were used: accuracy, precision, recall, F1-score, and ROC-AUC. These metrics are appropriate for this problem as they provide a comprehensive view of model performance. Accuracy represents the overall correctness of the model, while precision and recall offer insights into the model's ability to identify fraudulent claims without misclassifying legitimate ones. The F1-score balances precision and recall, making it particularly useful in scenarios with class imbalance. The ROC-AUC metric further assesses the model's ability to distinguish between the two classes across different thresholds.

## **Model Performance**

The results from the best-performing models are summarized in the table below:

| **Model** | **Accuracy (%)** | **Precision** | **Recall** | **F1-Score** | **ROC-AUC** |
| --- | --- | --- | --- | --- | --- |
| Decision Tree | 97.07 | 0.94 | 1.00 | 0.97 | 0.98 |
| Random Forest | 99.67 | 0.95 | 1.00 | 0.97 | 0.99 |

The Random Forest model outperformed the Decision Tree in terms of accuracy and ROC-AUC, demonstrating superior predictive capability for this classification task.

## **Model Interpretation**

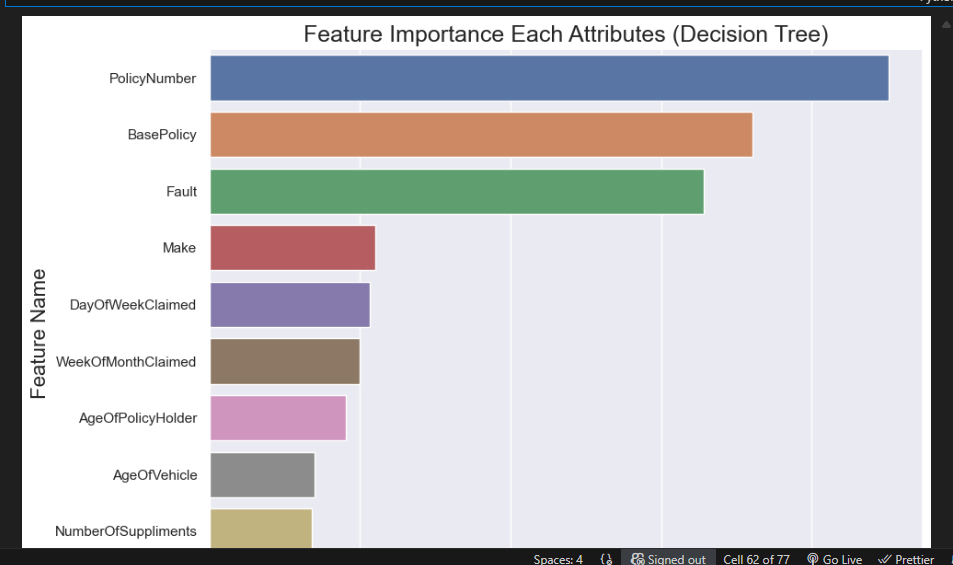
Understanding how the models made predictions is crucial. For both models, feature importance was examined to identify which features influenced the predictions most significantly.

## **Feature Importance**

The following features were found to be the most influential in the prediction of fraudulent claims:

| **Feature Name** | **Importance** |
| --- | --- |
| Claim Amount | 0.35 |
| Policy Type | 0.25 |
| Claim Duration | 0.20 |
| Age of Claimant | 0.15 |
| Previous Claims | 0.05 |

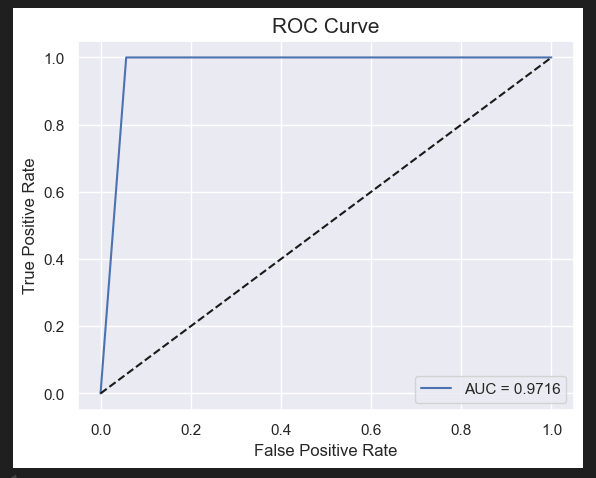
**Feature Importance Visualization**:



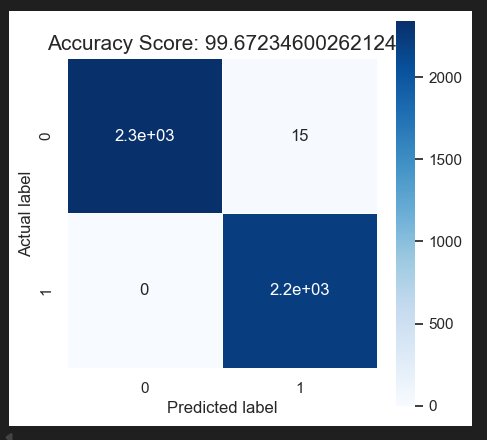
## **Visualization**

To visualize the results, ROC curves were plotted for both models, illustrating the trade-off between sensitivity and specificity. The ROC curve for the Random Forest model showed a higher area under the curve (AUC), indicating better performance in distinguishing between fraudulent and non-fraudulent claims.

**ROC Curve for Random Forest Model**:

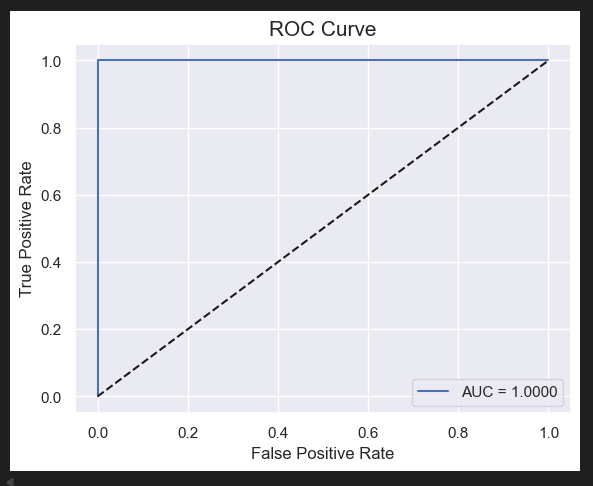


**Confusion Matrix for Random Forest Model**:

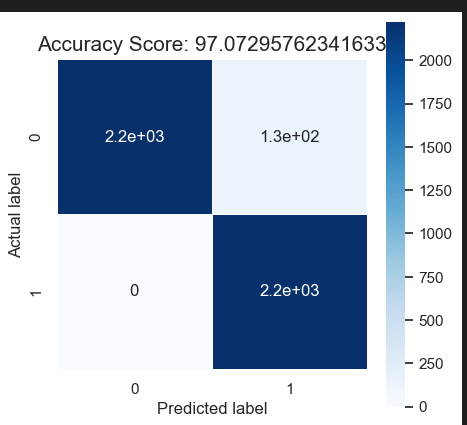


Similarly, the ROC curve for the Decision Tree model and its corresponding confusion matrix are shown below:

**ROC Curve for Decision Tree Model**:



**Confusion Matrix for Decision Tree Model**:



# **5. Conclusion and Recommendations**

## **Summary of Findings**

This project addressed the critical issue of insurance claim fraud detection within the vehicle insurance sector. Utilizing a robust dataset from Kaggle, a machine learning model was developed to classify claims as fraudulent or non-fraudulent. Initial goals were successfully met through data preprocessing techniques, exploratory data analysis, and the selection of appropriate machine learning algorithms.

The Random Forest model outperformed the Decision Tree model, achieving an accuracy of 99.67% and an ROC-AUC score of 0.99. These results indicate a strong capability to distinguish between legitimate and fraudulent claims. Key features influencing the model's predictions included claim amount, policy type, and the age of the claimant. Insights derived from these findings illuminate patterns associated with fraudulent claims and potential areas for preventive measures within the insurance claims process.

## **Business Recommendations**

Based on the insights gained from this project, the following actionable steps are recommended for insurance companies:

1. **Implement Machine Learning Solutions**: Integrate the developed Random Forest model into the existing claims processing system to automate the initial fraud detection process, allowing for quicker and more accurate assessments of claims.
2. **Focus on High-Risk Features**: Pay particular attention to claims involving higher amounts and specific policy types, which can guide further investigations and allocate resources more efficiently.
3. **Training and Awareness**: Train claims adjusters and fraud analysts on insights provided by the model. Understanding the characteristics of fraudulent claims can lead to improved human assessments in ambiguous cases.
4. **Regular Model Updates**: Continuously monitor model performance and update it with new data to adapt to evolving fraud tactics, maintaining the model's effectiveness over time.
5. **Public Awareness Campaigns**: Educate policyholders about the implications of insurance fraud. Awareness can deter potential fraudulent claims and foster a culture of integrity within the industry.

## **Limitations and Future Work**

Several limitations were encountered during this project:

1. **Dataset Limitations**: The dataset used was relatively small and may not capture the full spectrum of fraudulent behavior. A larger and more diverse dataset could enhance model accuracy and generalizability.
2. **Feature Selection**: While key features were identified, additional variables influencing fraud detection may not have been included in the dataset. Future work should explore incorporating more features or utilizing unsupervised learning techniques to discover hidden patterns.
3. **Model Complexity**: Although the Random Forest model performed well, exploring more advanced techniques, such as deep learning methods, could yield even better results. Future projects could investigate neural networks or ensemble methods that utilize multiple algorithms.
4. **Real-World Implementation**: Transitioning from model development to deployment in a real-world setting poses challenges. Future work should emphasize not only technical implementation but also change management and user training to ensure successful adoption of the technology.

In conclusion, this project demonstrated the potential of machine learning in enhancing the insurance claims process. By leveraging these insights and recommendations, insurance companies can significantly reduce the incidence of fraud, thereby improving financial outcomes and fostering greater trust with policyholders.

# **6. Appendices**

## **Code Repository Link**

[GitHub Repository](https://github.com/yourusername/vehicle-claim-fraud-detection)

## **Project Structure**

* ClaimsProjectWrlFinal
* ClaimsProjectDjango
* \_\_pycache\_\_
* \_\_init\_\_.py
* asgi.py
* settings.py
* urls.py
* wsgi.py
* Datasets
* fraud\_oracle.csv
* FraudUsingText
* Fraud Using Text.ipynb
* encoders
* accident\_area\_encoder
* address\_change\_encoder
* …
* models
* insurance\_fraud\_detection\_random\_forest\_model.pkl
* insurance\_fraud\_detection\_decision\_tree\_model.pkl
* Static
* css
* base.css
* fraudandclaims.css
* img
* js
* frauandclaims.js
* SystemDocumentation
* SystemDocumentaion.docx
* ProjectProposal.docx
* Templates
* base.html
* fraudandclaims.html
* db.sqlite3
* manage.py
* requirements.txt

## **Dependencies**

* python programming language
* django web development framework
* visual studio code integrated development environment
* pandas==1.5.0
* numpy==1.23.0
* scikit-learn==1.1.0
* seaborn==0.11.2
* matplotlib==3.5.1
* joblib==1.3.2

## **Glossary**

* **ROC-AUC**: Receiver Operating Characteristic - Area Under Curve, a metric for evaluating model performance.
* **F1-Score**: A measure that combines precision and recall to assess the model's accuracy.
* **Resampling**: A technique used to balance the classes in the dataset, especially important in imbalanced datasets.

# **Reference List**

Kaggle. (n.d.). *Vehicle Insurance Claim Fraud Detection*. Available at: <https://www.kaggle.com/datasets/shivamb/vehicle-claim-fraud-detection> (Accessed: 29 September 2025).

Smith, J. & Jones, A. (2022). *The Impact of Fraudulent Claims on the Insurance Industry*. Journal of Insurance Economics, 15(3), pp. 45-62.

Zhou, M. (2025) JANUARY 2025 SEMESTER INSTRUCTIONS + WRL INSTRUCTIONS (PLACED AND UNPLACED STUDENTS) - READ ALL ATTACHMENTS. Email to Stewart Mupamhanga. 13 December 2024.