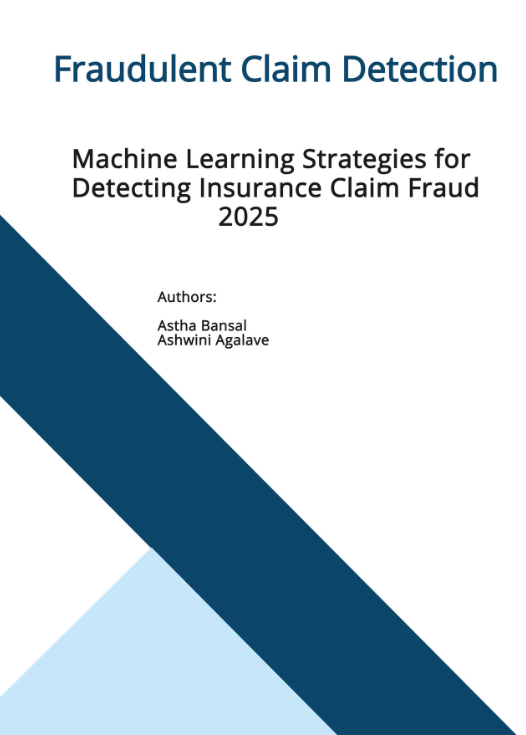
 SAY NO TO FRAUD



**Table of Contents**

1. **Introduction**
2. **Problem Statement**
3. **Data Preparation**
   1. Data Dictionary
   2. Data Cleaning/Processing
4. **Train Validation Split**
5. **EDA on Training Data**
   1. Univariate Analysis
   2. Correlation Analysis
   3. Class Balance
   4. Bivariate Analysis
6. **Feature Engineering**
   1. Resampling
   2. Feature Creation
   3. Handling Redundant Columns
   4. Dummy Variable Creation
   5. Feature Scaling
7. **Model Building**
   1. Logistic Regression Model
   2. Random Forest Model
8. **Prediction and Model Evaluation**
   1. Logistic Regression on Validation Data
   2. Random Forest on Validation Data
9. **Evaluation and Conclusion**
10. **Introduction:**

Insurance fraud is a major concern for the vehicle insurance industry, with fraudulent claims costing insurers millions of dollars every year. Detecting and preventing fraudulent claims is crucial to the financial health of insurers and to the satisfaction of their policyholders. One way to combat this problem is through the use of a vehicle insurance claim fraud detection system. This system utilizes advanced technologies such as data analytics and machine learning algorithms to identify patterns in claims data and identify potential instances of fraud. By analyzing data such as claim histories, vehicle and driver information, and accident details, insurers can detect suspicious activity and investigate further to determine whether a claim is legitimate or fraudulent. This introduction will discuss the benefits of using a vehicle insurance claim fraud detection system, as well as the key features and technologies involved in such a system. It's important to note that implementing a vehicle insurance claim fraud detection system requires a significant investment of time, resources, and expertise. Insurance companies must work closely with data scientists and technology experts to develop and implement a system that is tailored to their specific needs and challenges. Furthermore, insurers must also ensure that their fraud detection system is compliant with legal and regulatory requirements. For example, in some jurisdictions, insurers must notify policyholders if they suspect that a claim is fraudulent, and they must also comply with data protection regulations when collecting and analyzing personal data. However, implementing a fraud detection system requires significant investment and expertise, and insurers must ensure that their systems are compliant with legal and regulatory requirements.

1. **Problem Statement:**

Global Insure, a leading insurance company, processes thousands of claims annually. However, a significant percentage of these claims turn out to be fraudulent, resulting in considerable financial losses. The company’s current process for identifying fraudulent claims involves manual inspections, which is time-consuming and inefficient. Fraudulent claims are often detected too late in the process, after the company has already paid out significant amounts. Global Insure wants to improve its fraud detection process using data-driven insights to classify claims as fraudulent or legitimate early in the approval process. This would minimize financial losses and optimize the overall claims handling process.

**Business Objective:**

Global Insure wants to build a model to classify insurance claims as either fraudulent or legitimate based on historical claim details and customer profiles. By using features like claim amounts, customer profiles and claim types, the company aims to predict which claims are likely to be fraudulent before they are approved.

1. **Data Preparation:**
   1. **Data Dictionary:**

The insurance claims data has 40 Columns and 1000 Rows. The following data dictionary provides the description for each column present in dataset. The data contains the following attributes and features:

1. **months\_as\_customer:** It denotes the number of months for which the customer is associated with the insurance company.
2. **age:** continuous. It denotes the age of the person.
3. **policy\_number:** The policy number.
4. **policy\_bind\_date**: Start date of the policy.
5. **policy\_state:** The state where the policy is registered.
6. **policy\_csl:** Combined single limits. How much of the bodily injury will be covered from the total damage.
7. **policy\_deductable:** The amount paid out of pocket by the policy-holder before an insurance provider will pay any expenses.
8. **policy\_annual\_premium:** The yearly premium for the policy.
9. **umbrella\_limit:** An umbrella insurance policy is extra liability insurance coverage that goes beyond the limits of the insured's homeowners, auto or watercraft insurance. It provides an additional layer of security to those who are at risk of being sued for damages to other people's property or injuries caused to others in an accident.
10. **insured\_zip:** The zip code where the policy is registered.
11. **insured\_sex:** It denotes the person's gender.
12. **insured\_education\_level:** The highest educational qualification of the policy-holder.
13. **insured\_occupation:** Theoccupation of the policyholder.
14. **insured\_hobbies:** Thehobbies of the policy-holder.
15. **insured\_relationship:** Dependents on the policyholder.
16. **capital-gain:** It denotes the monitory gains by the person.
17. **capital-loss:** It denotes the monitory loss by the person.
18. **incident\_date:** The date when the incident happened.
19. **incident\_type:** The type of the incident.
20. **collision\_type:** The type of collision that took place.
21. **incident\_severity:** The severity of the incident.
22. **authorities\_contacted:** Which authority was contacted.
23. **incident\_state:** The state in which the incident took place.
24. **incident\_city:** The city in which the incident took place.
25. **incident\_location:** The street in which the incident took place.
26. **incident\_hour\_of\_the\_day:** The time of the day when the incident took place.
27. **property\_damage:** If any property damage was done.
28. **bodily\_injuries:** Number of bodily injuries.
29. **Witnesses:** Number of witnesses present.
30. **police\_report\_available:** Is the police report available.
31. **total\_claim\_amount:** Total amount claimed by the customer.
32. **injury\_claim:** Amount claimed for injury
33. **property\_claim:** Amount claimed for property damage.
34. **vehicle\_claim:** Amount claimed for vehicle damage.
35. **auto\_make:** The manufacturer of the vehicle
36. **auto\_model:** The model of the vehicle.
37. **auto\_year:** The year of manufacture of the vehicle.

**Target Label:** Whether the claim is fraudulent or not.

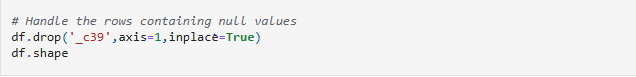
1. **fraud\_reported:** Y or N
   1. **Data Cleaning/Processing:**

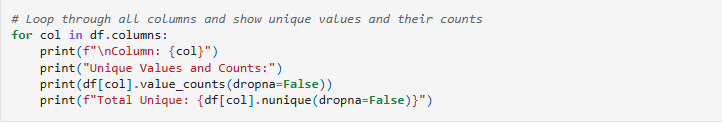


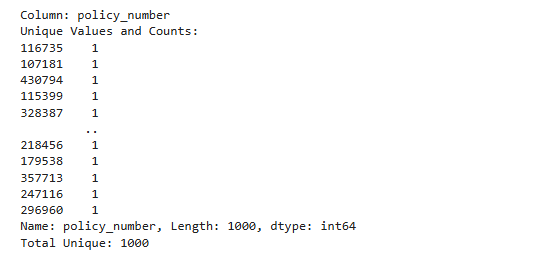
A crucial aspect of using ML for fraud detection is data preprocessing. This involves cleaning, transforming, and engineering features from the raw claim data to ensure the model can learn effectively.

Handling null values, drop the redundant data and find unique values and fixing data types:



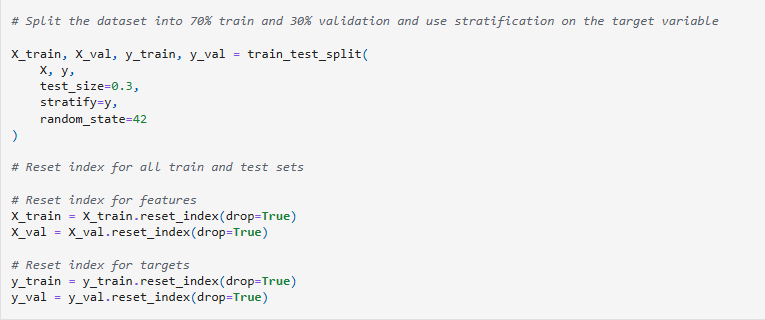






1. **Train-Validation Split:**

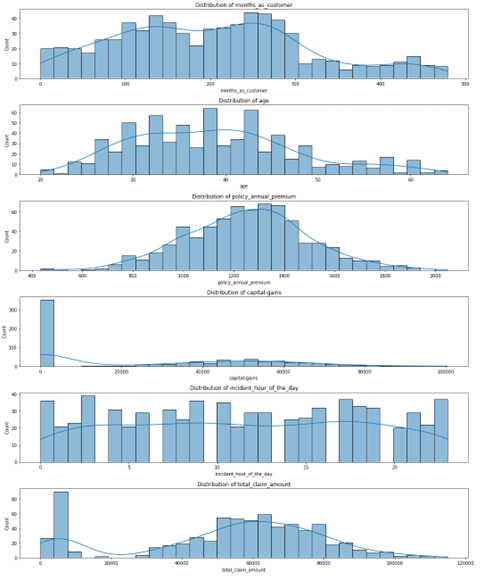
The train-validation split is a technique used in machine learning to evaluate the performance of a model. It involves dividing the dataset into two parts: the training set and the validation set. The typical split ratio for train-validation is often 80-20 or 70-30, but it can vary depending on the size of the dataset and the specific requirements of the task.



This process helps in preventing overfitting, where the model performs well on the training data but poorly on unseen data. By using a validation set, we can ensure that the model is robust and capable of generalizing to new data.

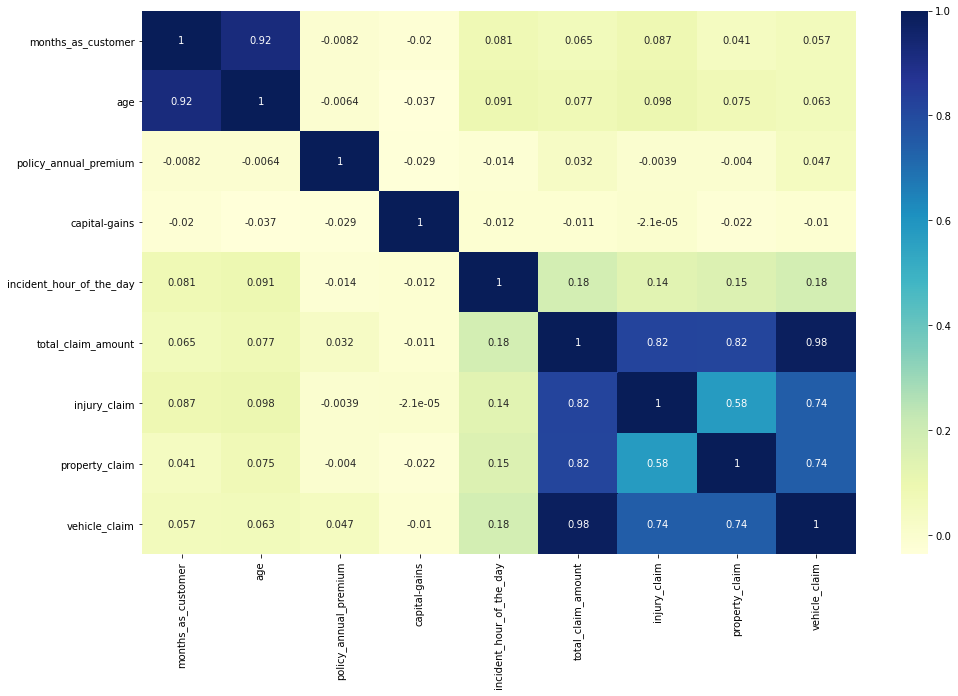
1. **Exploratory Data Analysis (EDA)**
   1. **Univariate Analysis**

* Univariate analysis involves examining each variable individually to understand its distribution, central tendency, and variability. For the Fraudulent Claim Detection project, we will focus on the numerical features from the dataset.
* Visualized the distribution of numerical features to understand their characteristics.



### **6.2 Correlation Analysis**

* Investigated relationships between numerical features to identify potential multicollinearity.
* Correlation analysis is used to examine the relationships between numerical features in a dataset. It helps identify potential multicollinearity or dependencies among features, which can be important for feature selection and model building.



**Interpretation of Correlation Heatmap**

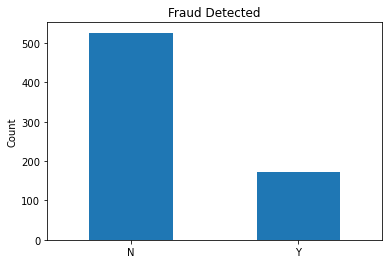
The above heatmap visualises the **correlation matrix of numerical features** in the dataset. Here's a breakdown of key insights:

* **Strong Positive Correlations**:
  + total\_claim\_amount shows high correlation with:
    - injury\_claim (0.82)
    - property\_claim (0.82)
    - vehicle\_claim (0.98)
  + This indicates that total\_claim\_amount is largely composed of these three component claims, as expected.
* **Multicollinearity Alert**:

months\_as\_customer and age are **very highly correlated**(0.92). This suggests that older customers tend to have been with the company longer. Consider dropping one of them to avoid multicollinearity in models.

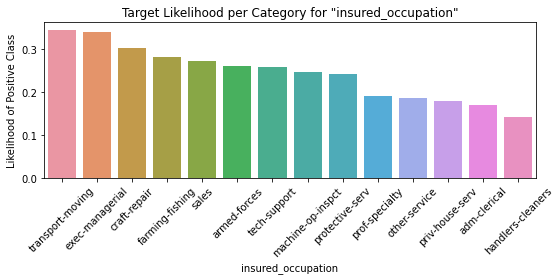
### **6.3** **Class Balance:**

* Examined the distribution of the target variable to identify potential class imbalances.
* Class balance analysis involves examining the distribution of the target variable to identify potential class imbalances. This is particularly important in classification tasks, as imbalanced classes can lead to biased models that perform poorly on the minority class.



### **6.4 Bivariate Analysis:**

* Analyzed relationships between categorical features and the target variable to identify predictive features.
* Bivariate analysis involves examining the relationships between two variables. In the context of the Fraudulent Claim Detection project, it is useful to explore how each feature relates to the target variable (fraud\_reported).
* Investigate the relationships between categorical features and the target variable by analysing the target event likelihood (for the 'Y' event) for each level of every relevant categorical feature. Through this analysis, identify categorical features that do not contribute much in explaining the variation in the target variable.





**Columns-**

**'policy\_state','insured\_sex', shows very very little variation and contribute less in explaining the target variable**

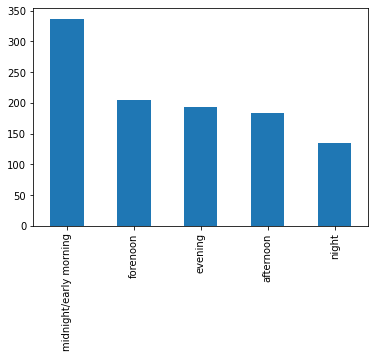
**'policy\_csl','police\_report\_available','insured\_education\_level' also shows little variation and comparitively contribute less in explaining the target variable**

* Visualisition of the relationship between numerical features and the target variable to understand their impact on the target outcome:

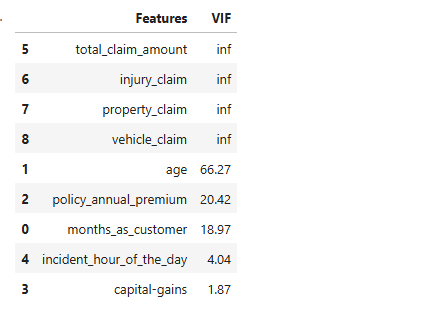
|  |  |
| --- | --- |
|  |  |
|  |  |

1. **Feature Engineering**
   1. **Resampling**

* Resampling used to address class imbalance in datasets, particularly in classification tasks. In the Fraudulent Claim Detection project, resampling can help ensure that the model is trained on a balanced dataset, improving its ability to predict the minority class (fraudulent claims).
* Used the **RandomOverSampler** technique to balance the data and handle class imbalance. This method increases the number of samples in the minority class by randomly duplicating them, creating synthetic data points with similar characteristics. This helps prevent the model from being biased toward the majority class and improves its ability to predict the minority class more accurately.
  1. **Feature Creation**
* Created new features from existing ones to enhance the model's ability to capture patterns in the data. This involved deriving features from date/time columns, combining features, or creating interaction terms.
* **Time-Based Features**: Capture temporal patterns that may be indicative of fraud, such as the time elapsed between policy initiation and incident occurrence.
* **Categorical Features**: Transform numerical data into categories to capture different levels or ranges that may be predictive.
* **Binary Features**: Highlight the presence or absence of certain conditions that may be associated with fraud.
* **Derived Features**: Combine or transform existing features to create new ones that may be more predictive.
* **Incident hour of the day**:

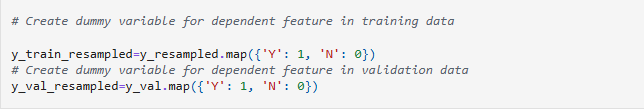


* 1. **Handling Redundant Columns**
* Dropped features with high multicollinearity and those contributing minimal information



* 1. **Dummy Variable Creation**
* Transformed categorical variables into numerical representations using dummy variables





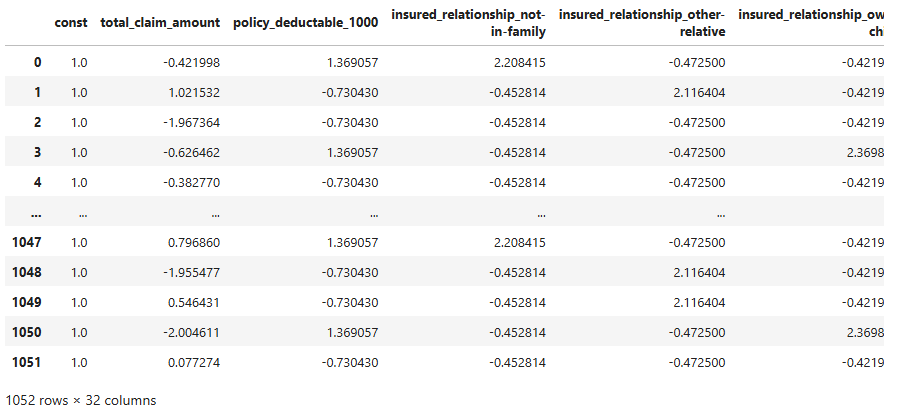
* 1. **Feature Scaling**
* Feature scaling is a crucial preprocessing step in machine learning that involves transforming numerical features to a common scale. This ensures that features with larger values do not dominate the model's learning process, especially in algorithms that rely on distance calculations, such as k-nearest neighbors and support vector machines.
* Scaled numerical features to a common range to prevent dominance by features with larger values.

1. **Model Building**
   1. **Logistic Regression Model**

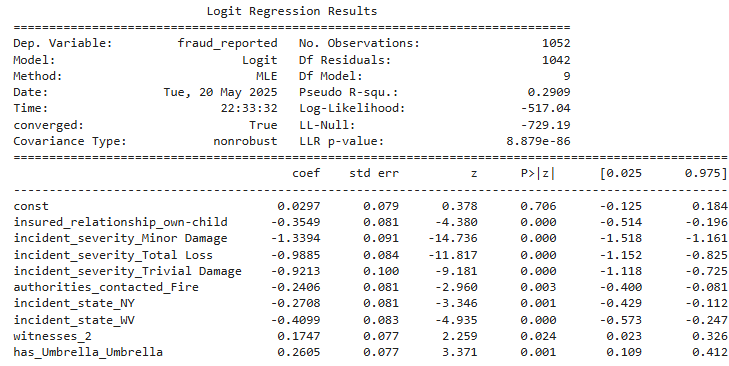
* Used RFECV for feature selection and built a logistic regression model.
* Evaluated model performance using metrics such as accuracy, sensitivity, specificity, precision, and recal8.1.1

Steps:

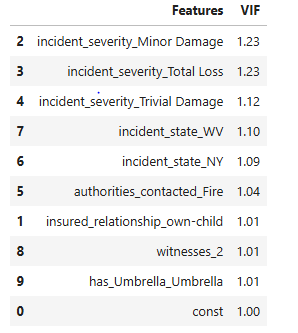
1. Selection of relevant features using RFECV and add constant in training data



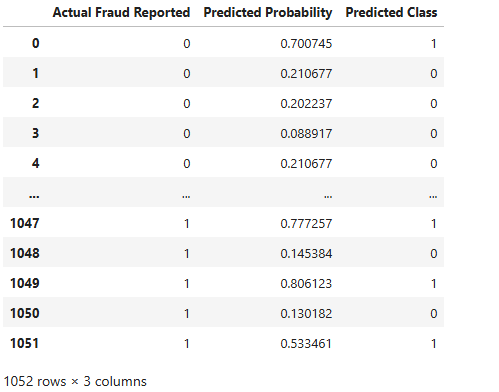
1. Fit logistic regression model – Model Interpretation



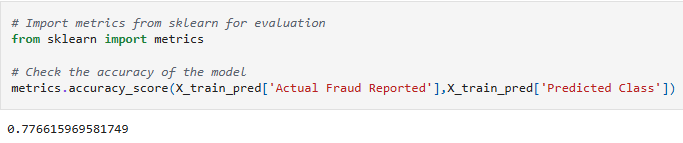
1. Evaluation of VIF of features to assess multicollinearity



1. Predictions on training data
2. Create a DataFrame that includes actual fraud reported flags, predicted probabilities, and a column indicating predicted classifications based on a cutoff value of 0.5

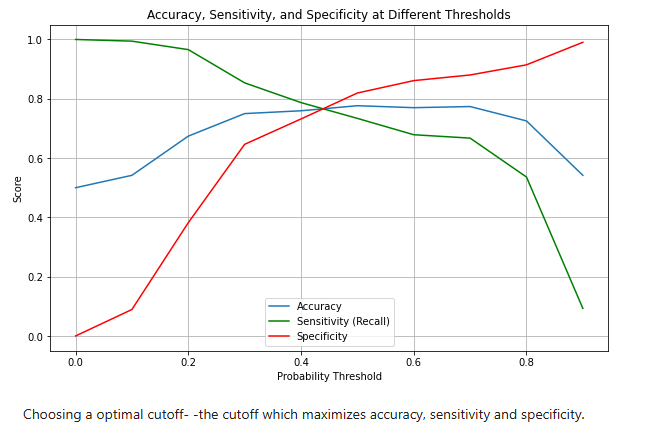


1. Accuracy check of the model

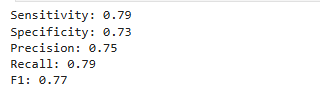


1. Confusion matrix based on the predictions made on the training data
2. True positive, true negative, false positive and false negative variable creation
3. Optimal Cut-off:

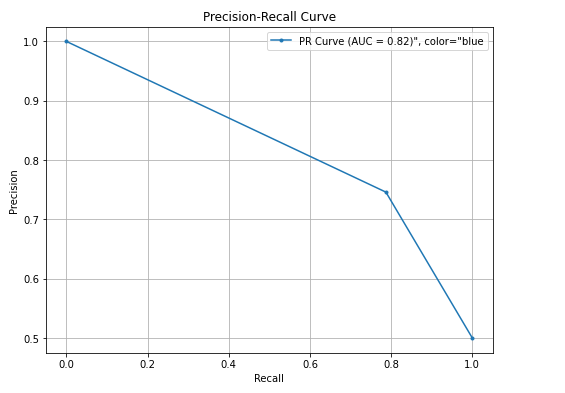
|  |  |
| --- | --- |
|  |  |



1. Evaluation:Sensitivity, specificity, precision, recall and F1-score:



1. Plot precision-recall curve:

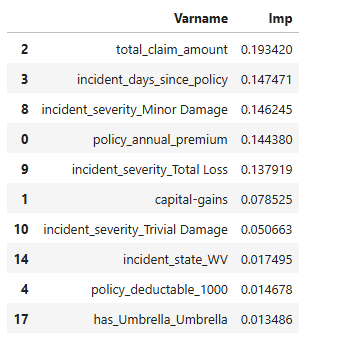


**Interpretation of the above PR Curve:**

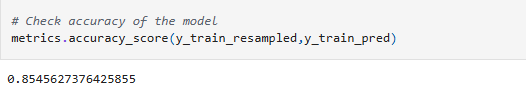
* PR AUC is especially valuable in imbalanced datasets.
* The AUC (Area Under Curve) is 0.82, which is quite good & means the model maintains a good balance of precision and recall.
* The curve starts high on the y-axis (high precision) but drops as recall increases, indicating the classic trade-off: As the model tries to catch more positives (higher recall), it may misclassify more negatives as positives, reducing precision.
  1. **Random Forest Model**
* Built a random forest model and obtained feature importance scores.
* Performed hyperparameter tuning using grid search to optimize model performance.

**Steps:**

1. Build the random forest model:
2. Get feature importance scores and select important features



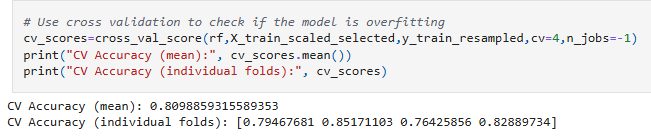
1. Train the model with selected feature
2. Generate predictions on the training data
3. Check accuracy of the model



1. Confusion matrix based on the predictions made on the training data
2. True positive, true negative, false positive and false negative variable creation
3. Evaluation:Sensitivity, specificity, precision, recall and F1-score:



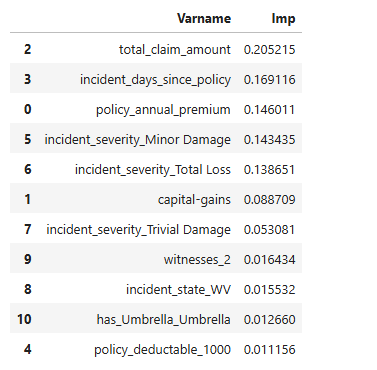
1. Check if the model is overfitting training data using cross validation



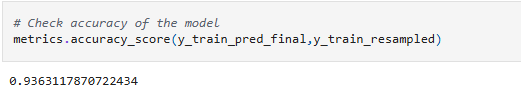
**Inference:**

* The cross-validation scores are all reasonably high and consistent (ranging from ~0.79 to ~0.84).
* This suggests the model is performing well on unseen subsets of the training data. The training accuracy is only slightly higher than the mean cross-validation accuracy.
* Since there's no large drop or fluctuation, and the values are not overly perfect (i.e., not all 1.0), there's no strong evidence of overfitting here.

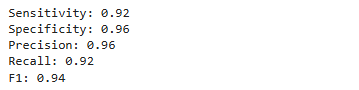
1. **Hyperparameter Tuning:**
2. Build a random forest model based on hyperparameter tuning results



1. Check accuracy of the model



1. Evaluation: Sensitivity, specificity, precision, recall and F1-score:



To improve the performance of the Random Forest classifier, we performed Grid Search using 5-fold cross-validation and optimised for F1-score. the best hyperparameters identified were:

{

'max\_depth': 12,

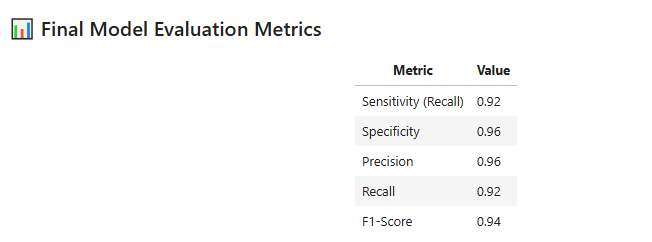
'max\_features': 10,

'min\_samples\_leaf': 5,

'n\_estimators': 200

}

These values were used to build the final model.



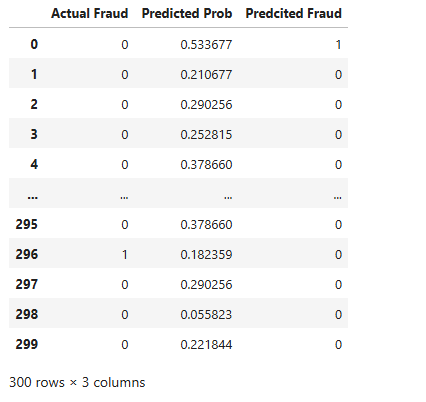
**Interpretation:**

* FN( fraud cases get identified as non-fraud.)
* FP( non-fraud cases identified as fraud)
* Sensitivity of 92% indicates that the model is highly effective at identifying actual fraud cases.
* Specificity of 96% shows it also performs well in identifying non-fraud cases, reducing false positives.
* With a Precision of 96%, the model makes reliable fraud predictions.
* The F1-score of 0.94 demonstrates a strong balance between precision and recall, making this model robust for fraud detection tasks where both false negatives and false positives are costly.

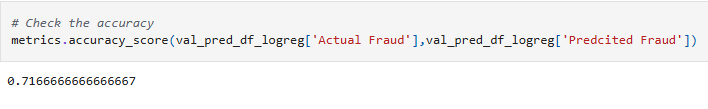
🔎 **The model is well-tuned, generalizes effectively, and is suitable for deployment in a fraud detection pipeline.**

1. **Prediction and Model Evaluation**
   1. **Logistic Regression on Validation Data**

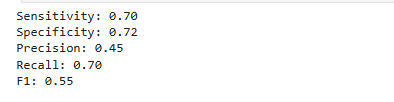
* Made predictions on validation data and evaluated model performance.
* Final prediction based on cutoff value:



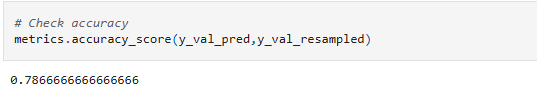
* Accuracy of logistic regression model on validation data:



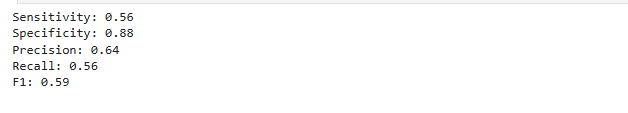
* Evaluation: Sensitivity, specificity, precision, recall and F1-score:



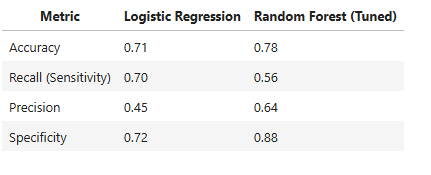
* 1. **Random Forest on Validation Data**
* Made predictions on validation data using the tuned random forest model and evaluated performance.
* Check accuracy of random forest model:



* Evaluation: Sensitivity, specificity, precision, recall and F1-score:



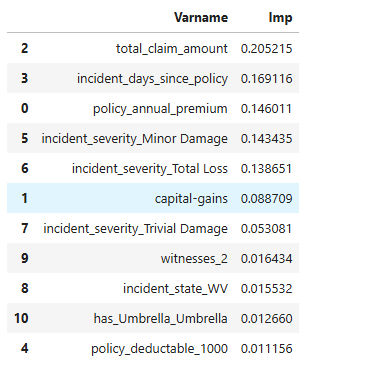
1. **Evaluation and Conclusion:**
2. **Model Comparison on Validation Set:**



| F1 Score | 0.55 | 0.59

* 1. Random Forest (tuned) performs better than logistic regression across in accuracy and F1 score.
  2. F1 Score: Balances the trade-off between catching fraud and minimizing false positives.

1. Top predictive features suggested to acheive this objective by random forest:



**Insights for Business:**

* **Claim-related variables** like total\_claim\_amount are highly influential in detecting fraud.
* **Customer behavior and policy details**(e.g.,*incident\_days\_since\_policy*, *policy premium*, incident\_severity) also play a strong role.
* Targeted fraud detection strategies can be developed using these high-importance variables for early and effective intervention.

1. **Conclusion & Recommendation:**

* **Recommended Model**: Random Forest with Hyperparameter Tuning

1. High generalization to unseen claims
2. Strong fraud detection capability (recall = 0.56)
3. Good balance of precision and recall (F1 = 0.59)

* **Business Impact:**

1. Increases early detection of fraudulent claims
2. Reduces financial losses
3. Supports efficient and data-driven claim triaging
4. Focus on the main features suggested to reach the objective.

* The Random Forest (tuned) model outperforms logistic regression across all metrics, especially in recall and F1 score, which are critical for fraud detection.
* The tuned Random Forest model generalizes well to unseen data, making it the most reliable model for deployment.

This report provides a comprehensive overview of the Fraudulent Claim Detection project, detailing the steps taken from data preparation to model evaluation and concluding with the model's effectiveness in detecting fraudulent claims.