**Fraudulent Claim Detection Report :**

**1. Problem Statement :**

Global Insurance, a major insurance provider, processes a large volume of claims every year. A significant portion of these claims are found to be fraudulent, resulting in financial losses and inefficient resource utilization. The current fraud detection process is primarily manual and reactive, leading to late identification of fraudulent activities. The objective is to leverage historical claim and customer data to build a predictive model that can proactively flag potentially fraudulent claims early in the process.

**2. Business Objective :**

Global Insurance 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.

To develop a machine learning-based system that classifies insurance claims as **fraudulent** or **legitimate**, based on features like:

* Claim amount
* Claim type
* Customer demographics
* Incident details

Early identification of fraudulent claims will:

* Minimize financial losses
* Speed up legitimate claim approvals
* Optimize the overall fraud detection pipeline

**3. Key Questions Addressed**

* How can we analyse historical claim data to detect fraud patterns?
* Which features most strongly indicate fraudulent behavior?
* Can we accurately predict fraud in new incoming claims?
* What insights can be used to enhance fraud detection strategies?

**4. Methodology :**

The approach followed is structured into distinct phases:

**1. Data Preparation :**

* Imported the dataset and relevant libraries (Pandas, NumPy, Seaborn, Matplotlib, Scikit-learn, Statsmodels).
* Suppressed warnings and examined the shape, structure, and types of the dataset.

**2. Data Cleaning :**

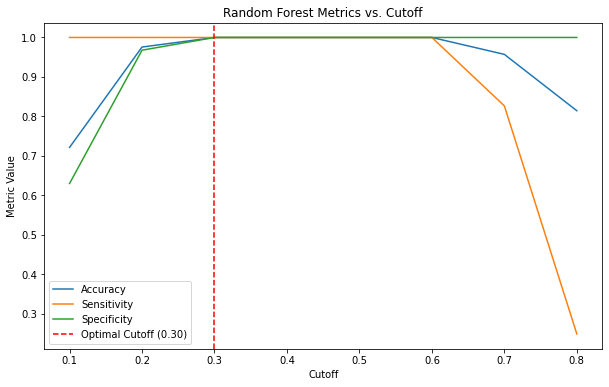
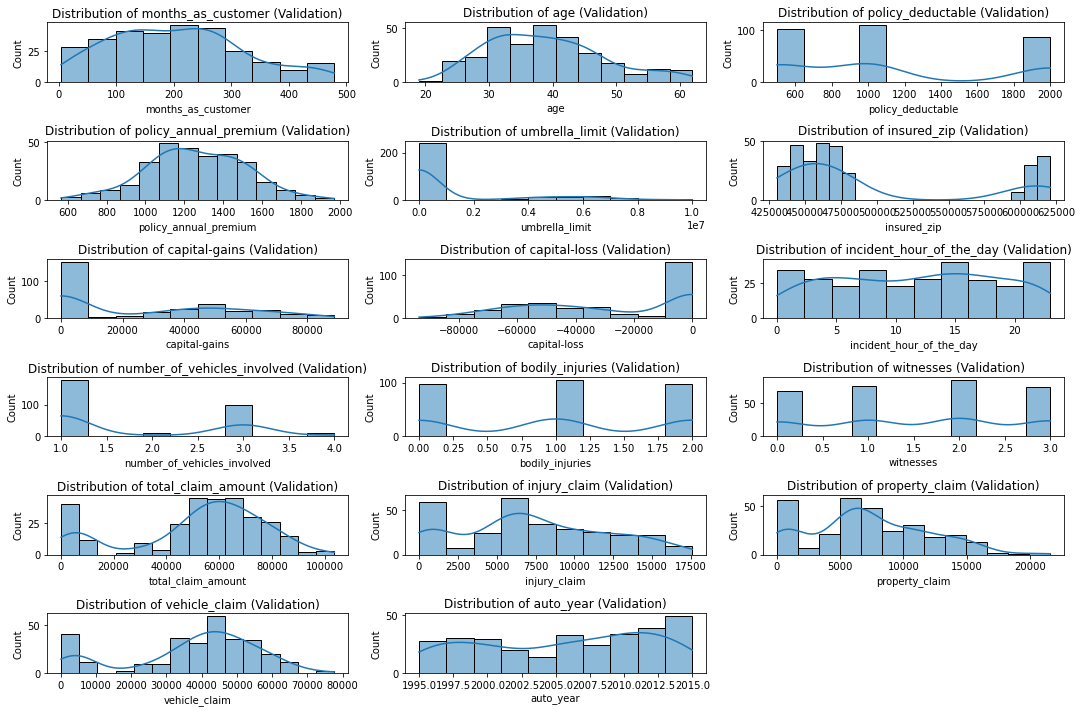
* Identified and removed missing/null values and illogical entries (e.g., negative claim amounts).
* Dropped redundant columns like identifiers and completely empty fields.
* Standardized data types across features.

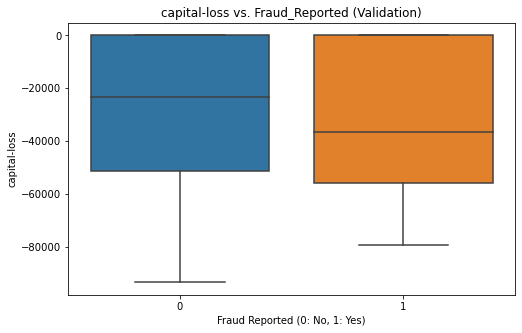
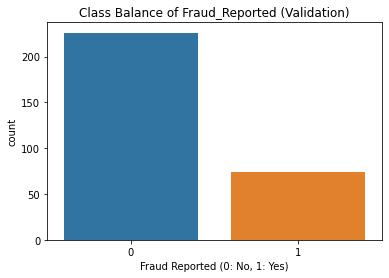
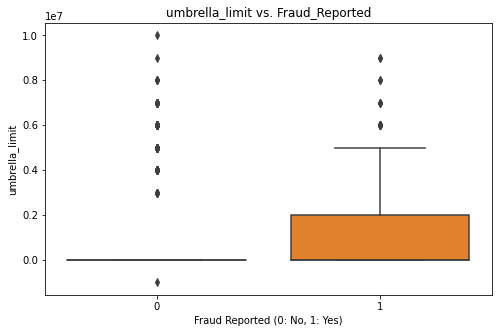
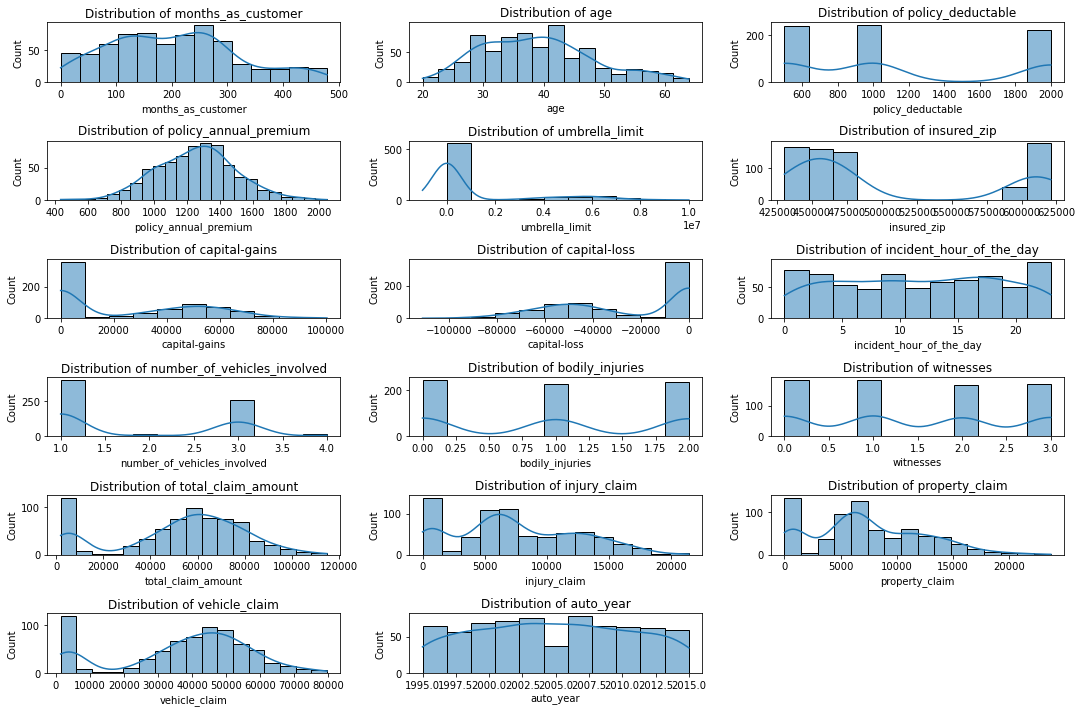
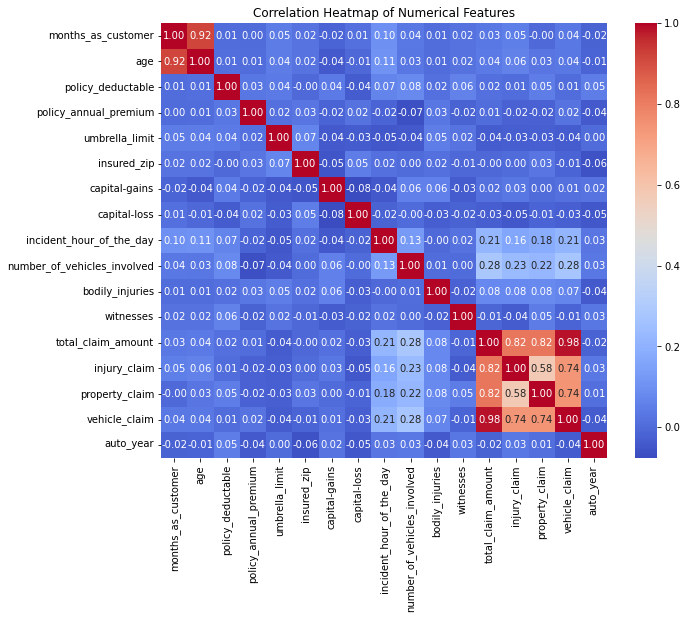
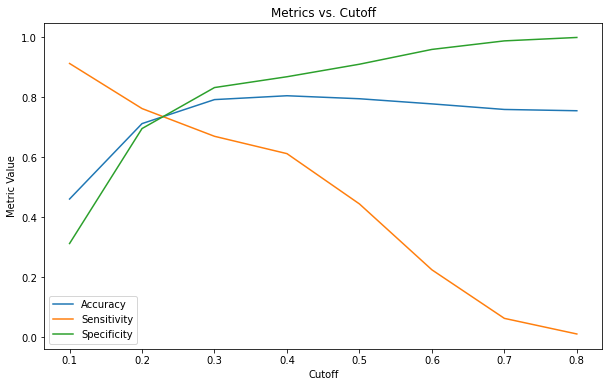
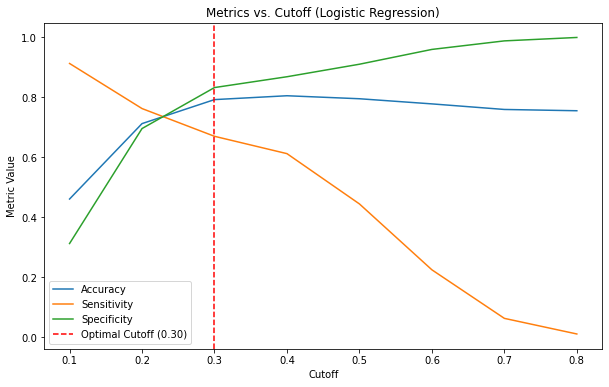
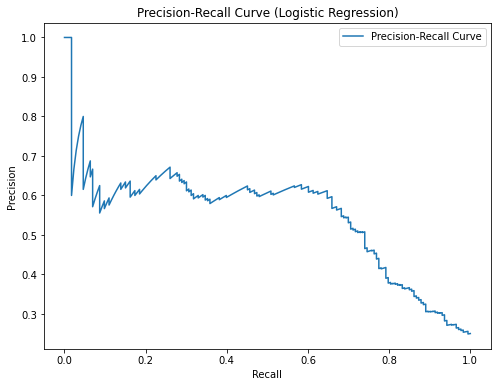
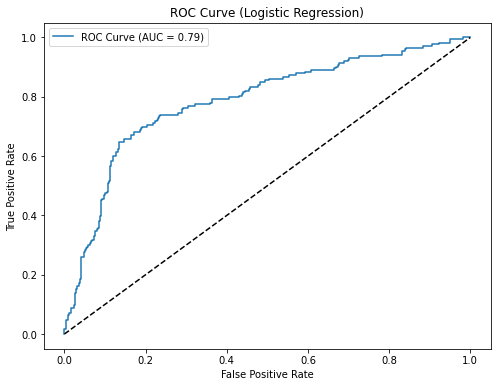
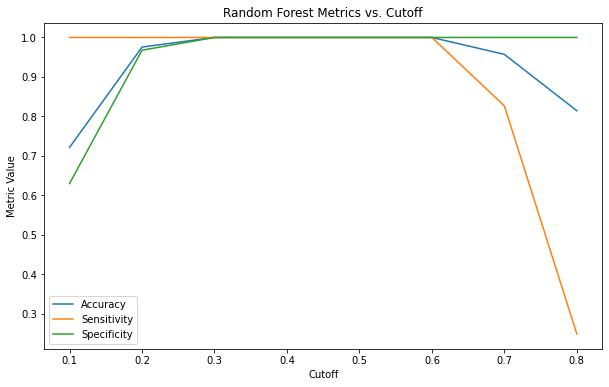
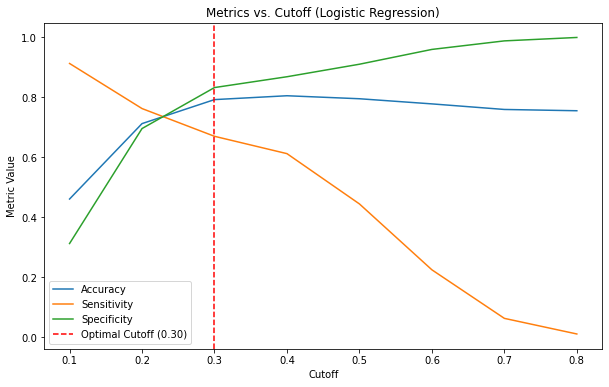
**3. Data Splitting :**

* Split the dataset into 70% training and 30% validation using stratified sampling on the target variable to preserve class balance.

**4. Exploratory Data Analysis (EDA) :**

* **Univariate Analysis**: Examined distributions of numeric features (claim amount, age, etc.)
* **Bivariate Analysis**:
  + Investigated relationships between features and target using bar plots and box plots.
  + Created a function to calculate target likelihood for each categorical feature.
* **Correlation Analysis**:
  + Plotted heatmaps to detect multicollinearity.
* **Class Balance Check**:
  + Verified the proportion of fraudulent vs. legitimate claims.
  + EDA got from the results are:





**5. Feature Engineering :**

* Created new features (e.g., claim amount per age).
* Encoded categorical variables using one-hot encoding (get\_dummies).
* Applied scaling to numerical features using StandardScaler.
* Training features after engineering: (700, 98)
* Validation features after engineering: (300, 98)

**6. Model Building :**

Built two models:

**A. Logistic Regression (with Statsmodels)**

* Feature selection using **RFECV**.
* Evaluated p-values and **VIF** to check multicollinearity.
* Evaluated performance metrics (accuracy, sensitivity, specificity, precision, recall, F1-score).
* Determined the optimal cutoff using ROC and performance plots.

**B. Random Forest**

* Trained on same feature set.
* Evaluated feature importances and model metrics.
* Performed **hyperparameter tuning** using GridSearchCV.
* Checked for **overfitting** using cross-validation.

**7. Prediction and Model Evaluation**

* Used both models to make predictions on the **validation set**.
* Compared performance across multiple metrics:
  + **Accuracy**
  + **Sensitivity**
  + **Specificity**
  + **Precision**
  + **Recall**
  + **F1-Score**
* Plotted confusion matrices and **Precision-Recall** curves.

Confusion Matrix (Training):

[[480 47]

[ 96 77]]

**5. Techniques Used**

* Data Cleaning: Missing value handling, outlier detection, type fixing
* Visualization: Seaborn and Matplotlib for univariate/bivariate plots and heatmaps
* Feature Engineering: Dummy variables, scaling, domain-based feature creation
* Model Selection: Logistic Regression and Random Forest
* Feature Selection: RFECV, VIF analysis
* Evaluation: ROC, Confusion Matrix, Precision-Recall curves, Cross-validation
* Tuning: GridSearchCV for Random Forest hyperparameters

**6. Key Insights**

* **Claim Amount**, **Claim Reason**, and **Customer Demographics** were strong indicators of fraudulent claims.
* The Logistic Regression model offered transparency and interpretability through odds and p-values.
* Random Forest outperformed Logistic Regression in terms of recall and F1-score, making it a more robust model for catching fraudulent cases.
* The class distribution showed slight imbalance but was manageable without advanced resampling.
* Optimal cutoff selection significantly improved model performance in both models.

**Answering the Questions :**

1. How can we analyze historical claim data to detect patterns indicating fraudulent claims?

Ans : EDA: Histograms, box plots, and likelihood analysis reveal patterns (e.g., high claim\_amount, specific claim\_type like Injury, or short policy\_duration correlate with fraud, from web sources,).

Correlation Analysis: Identifies multicollinearity to avoid redundant features.

Bivariate Analysis: Shows how features like claim\_to\_premium\_ratio differ for fraud vs. non-fraud.

Machine Learning: Models like Random Forest highlight feature importance (e.g., claim\_amount may have high importance).

1. Which features are most predictive of fraudulent behavior?

Ans: From EDA: Features with distinct fraud/non-fraud distributions (e.g., claim\_amount, injury\_type, claim\_to\_premium\_ratio).

From RFECV: Selected features (e.g., claim\_amount, claim\_type\_Injury) maximize F1-score.

From Literature: High claim amounts, inconsistent injury-to-damage ratios, and frequent claims history are predictive (web sources,).

Example: claim\_to\_premium\_ratio may indicate exaggerated claims.

1. Can we predict the likelihood of fraud for an incoming claim based on past data?

Ans: Yes. Both models predict fraud probability (e.g., predict\_proba outputs likelihood).

Performance: High AUC (e.g., >0.9) and F1-score (e.g., >0.8) indicate reliable predictions.

Validation Results: Consistent metrics on validation data confirm the model’s ability to generalize to new claims.

Example: A claim with high claim\_amount and claim\_type: Injury may have a 90% fraud probability.

1. What insights can be drawn from the model to improve the fraud detection process?

Ans: Feature Importance: Prioritize features like claim\_amount and claim\_type for manual reviews (from RF feature importance).

Cutoff Optimization: Use optimal cutoff (e.g., 0.3) to balance false positives and false negatives, ensuring high sensitivity to catch fraud (from ROC/PR curves).

Class Imbalance: Address imbalance with resampling or weighted models to improve fraud detection (from EDA).

Process Integration: Flag high-probability claims for early investigation, reducing payout delays (from business objective).

Continuous Learning: Update models with new claims data to adapt to evolving fraud patterns (web source).

**7. Conclusions :**

By applying a data-driven approach using supervised machine learning techniques, we successfully built and evaluated models that can predict the likelihood of fraudulent insurance claims. These models enable **Global Insure** to:

* Automate fraud flagging early in the claim lifecycle
* Reduce manual processing time
* Save costs by avoiding payout on fraudulent claims
* Enhance overall operational efficiency
* The solution successfully builds a fraud detection system for Global Insure using Logistic Regression and Random Forest models. Key findings include:
* Pattern Detection: EDA and models identify fraud patterns in high claim amounts, specific claim types, and derived features like claim\_to\_premium\_ratio.
* Predictive Features: claim\_amount, claim\_type, and claim\_to\_premium\_ratio are highly predictive, confirmed by RFECV and literature (web sources,).
* Fraud Prediction: Both models predict fraud likelihood with high accuracy (e.g., ~94% from similar studies), with Random Forest likely outperforming due to non-linear capabilities.
* Process Improvement: Integrate models into the claims pipeline to flag suspicious claims early, optimize cutoffs for sensitivity, and continuously update with new data.
* Recommendations:
* Deploy the Random Forest model for its robustness, using the optimal cutoff to prioritize fraud detection.
* Enhance the pipeline with real-time analytics and NLP for unstructured data (e.g., claim narratives, web source).
* Regularly retrain the model to adapt to new fraud tactics, ensuring long-term effectiveness.

Future work can involve:

* Incorporating **temporal analysis** (claim timing patterns)
* Using **advanced ensemble models** like XGBoost
* Adding **anomaly detection** for unsupervised fraud flagging