**FRAUDULENT CLAIM DETECTION**

**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 companys 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 minimise financial losses and optimise 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.

**Data Loading and Preparation:**

The data is stored as a CSV file and contains information such as customer-related details (age, occupation and hobbies), policy information (coverage limits and premiums), incident specifics (type, severity and location), claim details (amounts and types of damage), vehicle information (make, model and year) and whether the claim was reported as fraud.

* **Insurance\_claims.csv** dataframe loaded using pandas and has data around 1000 records with 40 columns.
* Initially imported standard libraries and suppressing warnings code.

**Data Cleansing** :

Data is later checked for null values and columns **authorities\_contacted, \_c39** are treated for Null values by filling missing values with Mode for **authorities\_contacted** and by deleting **\_c39** column as this column has completely null values.

Once confirming data is free of null values, we worked on finding columns with Redundant data. Here we have removed columns **policy\_number, policy\_bind\_date, policy\_annual\_premium, insured\_zip, incident\_location** as these are high Uniqueness Columns. Fixing data type for field **incident\_date** to correct datetime format for proper analysis.

**Data Preprocessing:**

Splitting the data to train and test data set with 70:30 ratio and doing stratification on the target variable using train\_test\_split from sklearn.model\_selection.

**Exploratory Data Analysis:**

**Training Data:**

* For univariate Analysis, first we identified numerical columns and plotted using Histograms to understand their characteristics.
* Also performed Correlation analysis for these numerical columns and found insights like Age and months\_as\_customer have a higher correlation of 0.92 and fields total\_claim\_amount, injury\_claim, property\_claim and vehicle\_claim have stronger correlations among themselves.
* We have checked class balance for target variable to identify potential class imbalances using Visualisation.
* Investigating the relationship between Categorical Variables and target table by plotting Boxplots.
* Variables Umbrella\_limit, total\_claim\_amount, injury\_claim, vehicle\_claim showed visible differences.
* Then we performed the same analysis on test data for better insights.

**Feature Engineering:**

* Using RandomOverSampler, we handled class imbalance in the training data.
* Created new features like incident\_month, incident\_dayofweek and some more features from total\_claim\_amount.
* For handling redundant data, we have dropped columns incident\_date, property\_claim, injury\_claim, vehicle\_claim, auto\_model, incident\_city and insured\_hobbies.
* Created dummy variables for all Categorical Variables to make into numerical representation for both Train data and validation data.
* Also performed StandardScaling for both train data and validation data.

**Model Building:**

We have created Logistic Regression Model and Random Forest Models to check performance analysis.

**Logistic Regression Model:**

* For model building, we have used RFECV for feature selection using roc\_auc as scoring and we got optimal number of features as 72 and built model with these features after adding constant to train data and fitting accordingly.
* Evaluated VIF of features for multicollinearity and as expected months\_as\_customer and age have more value and we can create model again by removing the months\_as\_customer field and then retrain the model.
* Making predictions on training data and test data using a cutoff of 0.5 initially for classification.
* Evaluated model using metrics accuracy\_score, confusion\_matrix, precision\_score, recall\_score, f1\_score, roc\_curve, auc, sensitivity, specificity.
* Then found optimal cutoff using ROC Curve to improve model performance and repeated the whole metrics and compared with initial model.

**Random Forest Model:**

* Built a basic Random Forest Model to get feature importance scores and select features with high importance scores with a threshold of 0.005.
* Now training the model with selected features and then predicting with training data and calculate performance.
* Evaluated model using metrics accuracy\_score, confusion\_matrix, precision\_score, recall\_score, f1\_score, roc\_curve, auc, sensitivity, specificity.
* Checking for Overfitting of model using cross validation and based on CV accuracy, we had issue with overfitting.
* Using grid search for hyperparameter tuning, we used parameters n\_estimators, max\_depth, min\_samples\_leaf and max\_features to find best parameters.
* Building a random forest model based on hyperparameter tuning results and then evaluating again with training data.
* This definitely improved performance of model by reducing Overfitting with decreased accuracy for training data and improved accuracy for test data.
* Using Optimal Cutoff, and predicting results for Classification of test data.

**Insights and Interpretations:**

Upon observing results from Logistic Regression and Random Forest Models, we can clearly infer about the most predictive features that affect frauds reported which in turn helps in predicting Likelihood for new claims.

1. Incident\_severity
2. Total\_claim\_amount
3. Insured\_hobbies
4. Umbrella\_limit
5. Injury\_claim
6. Incident\_type
7. Insured\_occupation

Also good separation observed between Fraudulent and Legitimate Claims in ROC-AUC.

There are some misclassifications present, but recall and precision were balanced.

**Recommendations:**

* Model performance is good, we can incorporate feature importance analysis into operational decisions as it helps in finding fraudulent claims to most extent.
* But we need to keep retraining models with updated claim data regularly to upgrade the insights for better analysis.
* We can finetune more of these models with more information to improve accuracy and performance of models for detecting Fraud in most efficient way.