# AI-Powered Fraud Detection in Auto Insurance: Predictive Modeling for Smarter Claims Management

## Problem Statement:-

Insurance fraud is a significant and growing concern in the auto insurance industry, costing companies billions each year. Fraudulent claims not only lead to financial losses but also result in increased premiums for genuine policyholders and operational inefficiencies for insurers.

The objective of this project is to develop an intelligent, data-driven system capable of accurately identifying potentially fraudulent auto insurance claims. By analyzing customer and policy-related features, the system should be able to differentiate between genuine and suspicious claims, even when the patterns are subtle or obscured.

To achieve this, machine learning techniques are applied to historical claim data, leveraging both traditional and ensemble models. The ultimate goal is to enhance the efficiency and accuracy of fraud detection while minimizing false positives, thereby supporting better decision-making in the claims processing workflow.

## Methodology

**The development of the fraud detection system followed a structured pipeline consisting of several key stages:**

**1. Data Collection and Understanding**

- **Source: A structured dataset containing real-world auto insurance claim records.**
- **Dataset Characteristics:**
    - **Multiple numerical and categorical features.**
    - **Target variable: Fraud_Ind (1 = Fraudulent claim, 0 = Genuine claim).**

**2. Data Preprocessing**

- **Missing Value Treatment: Imputed or removed missing entries.**
- **Data Type Conversions: Dates were converted to meaningful numerical intervals (e.g., days between accident and license expiry).**
- **Outlier Detection and Treatment: IQR-based capping was used for selected numerical features.**
- **Encoding:**
    - **Label Encoding for high-cardinality categorical features.**
    - **One-Hot Encoding for nominal variables.**
- **Feature Alignment: Ensured consistent feature structure across training, testing, and validation datasets**

**3. Feature Engineering & Selection**

- **Top 5 Features Identified (based on multiple selection methods like Mutual Information, F-test, Boruta, Chi-Square):**
    - **Policy_Num, Accident_Location, Insured_Zip, Policy_Premium, DL_Expiry_Before_Accident_Days**

**4. Class Imbalance Handling**

- **Applied SMOTE (Synthetic Minority Over-sampling Technique) on the training set to handle class imbalance in the Fraud_Ind column.**

**5. Model Training & Evaluation**

**Trained 10 different machine learning models, including:**

- **Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, Naive Bayes, XGBoost, LightGBM, CatBoost, Gradient Boosting, AdaBoost**

**Each model was trained using a scikit-learn Pipeline that includes preprocessing and classification steps.**

**6. Evaluation Metrics**

**Each model was evaluated using multiple performance metrics:**

- **Accuracy, Precision, Recall (Sensitivity), Specificity, F1-Score, ROC-AUC Score, MCC (Matthews Correlation Coefficient), Log Loss**

**Additionally, overfitting was checked by comparing training vs. testing accuracy.**

**7. Best Model Selection**

**The Decision Tree classifier was selected as the best model based on:**

- **Perfect scores across all metrics (1.0 for Accuracy, Precision, Recall, etc.).**
- **Zero overfitting gap (Train Accuracy = Test Accuracy = 1.0).**
- **Lowest log loss.**

**8. Validation on Unseen Data**

- **The best model was used to predict fraud outcomes on a separate validation dataset of 10,000 records.**
- **Final predictions were saved in: Validation_Predictions_By_Best_Model.csv.**

**9. Visualization & Interpretation**

- **Model-wise metric comparison using boxplots.**
- **A flow diagram was generated to depict the end-to-end process.**

---

# Results:

This section highlights the performance of all ten machine learning models trained on the fraud detection dataset. Multiple evaluation metrics were used to ensure a robust and fair comparison across models.

| Model | Accuracy | Precision | Recall | Specificity | F1-Score | ROC-AUC | MCC | Log Loss |
|---|---|---|---|---|---|---|---|---|
| **Logistic Regression** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.0239 |
| **Decision Tree** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| **Random Forest** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| **Gradient Boosting** | 0.736 | 1 | 0.0075 | 1 | 0.0149 | 0.9148 | 0.0744 | 0.4879 |
| **AdaBoost** | 0.736 | 1 | 0.0075 | 1 | 0.0149 | 0.5966 | 0.0744 | 0.5711 |
| **Naive Bayes** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| **Support Vector Machine** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| **XGBoost** | 0.9956 | 1 | 0.9833 | 1 | 0.9916 | 1 | 0.9886 | 0.2486 |
| **LightGBM** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.1687 |
| **CatBoost** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.0697 |

Overfitting Detection:-

(Higher gap = more overfitting)

| Model | Train Accuracy | Test Accuracy | Overfitting Gap |
|---|---|---|---|
| AdaBoost | 1 | 0.736 | 0.264 |
| Gradient Boosting | 1 | 0.736 | 0.264 |
| XGBoost | 1 | 0.9956 | 0.0044 |
| Logistic Regression | 1 | 1 | 0 |
| Decision Tree | 1 | 1 | 0 |
| Random Forest | 1 | 1 | 0 |
| Naive Bayes | 1 | 1 | 0 |
| SVM | 1 | 1 | 0 |
| LightGBM | 1 | 1 | 0 |
| CatBoost | 1 | 1 | 0 |

Participates:

- ❖ Debasish Senapati- 22cse328
- ❖ Rajesh Kumar Panda- 22cse857
- ❖ Rakesh Senapati- 22cse329
- ❖ M.Bharat:- 22cse793