PROJECT REPORT: Car Insurance Claim Prediction

Student: Michael Sharuk

Course: Data Science (GUVI | HCL)

Domain: Insurance / Finance / Risk Analytics

---

1. Project Overview

The objective of this project is to build a predictive model to determine whether a customer will make a car insurance claim in the upcoming policy period. This involves analyzing demographic, vehicle, and policy-related features to improve business outcomes.

Business Use Cases

- **Fraud Prevention:** Identifying high-risk customers to reduce losses.
- **Pricing Optimization:** Adjusting premiums based on predicted likelihood of claims.
- **Operational Efficiency:** Assisting claim departments in resource allocation.

---

2. Data & Feature Engineering

The dataset consists of 40+ variables. Key features include:

- **Numerical:** `policy_tenure`, `age_of_car`, `age_of_policyholder`, `population_density`.
  +3
- **Categorical:** `area_cluster`, `segment`, `fuel_type`, `transmission_type`.
  +3
- **Safety Specs:** `airbags`, `is_esc`, `ncap_rating`, `is_brake_assist`.
  +3
- **Target:** `is_claim` (Boolean flag indicating if a claim was filed).

3.Technical Approach

3.1 Preprocessing

Following the modular approach, I implemented a clean pipeline:

1. **Missing Values:** Handled null entries to ensure data integrity.
2. **Encoding:** Applied One-Hot Encoding to categorical variables like `fuel_type` and `segment`.
3. **Scaling:** Normalized numerical features for algorithm compatibility.

3.2 Model Training

I transitioned from baseline models (Logistic Regression, Decision Trees) to advanced ensemble methods:

- **Random Forest:** For feature robustness.
- **XGBoost:** For high-performance classification and handling class imbalance.

---

4. Implementation Code

```
# Model Evaluation & Metrics
from sklearn.metrics import classification_report, roc_auc_score, ConfusionMatrixDisplay,
import matplotlib.pyplot as plt

# 1. Prediction Generation
y_pred = clf_pipeline.predict(X_val)
y_proba = clf_pipeline.predict_proba(X_val)[:, 1]

# 2. Evaluation Metrics
print(f"ROC-AUC Score: {roc_auc_score(y_val, y_proba):.4f}")
print("\nClassification Report:")
print(classification_report(y_val, y_pred))

# 3. Visual Error Analysis
ConfusionMatrixDisplay.from_estimator(clf_pipeline, X_val, y_val, cmap='Blues')
plt.title("Confusion Matrix: Insurance Claim Prediction")
plt.show()
```

5.Results and Evaluation

- **Predictive Power:** Developed an ML model capable of predicting claim probability.
- **Feature Importance:** Identified top predictive features (e.g., car age and tenure) influencing claims.
- **Metrics achieved:** Evaluated based on Accuracy, Precision, Recall, F1-Score, and ROC-AUC.

6.Deployment

The final model was serialized using `joblib` and deployed via a **Streamlit** dashboard. This allows end-users to input vehicle details and receive an instant risk assessment.

Python

```
# Streamlit deployment command
!streamlit run app.py
```

7.Project Guidelines & Standards

- **Coding:** Followed PEP8 conventions with modular functions and clear variable names.
- **Version Control:** Used Git to track progress and maintain history.
- **Reproducibility:** Fixed random seeds in all models to ensure consistent results