CAR INSURANCE PREMIUM PREDICTION USING PYTHON REGRESSION FOR A MID-TIER INSURANCE ANALYTICS TEAM

1. Introduction

An insurance analytics division within a mid-tier auto insurer approached us to modernize their premium estimation process. The firm relied on fixed pricing brackets developed through historical heuristics, which did not reflect the risk profile of new policyholders accurately. This often resulted in mispriced premiums, revenue leakage, and customer dissatisfaction.

They required a machine learning solution developed in Python that could predict the expected annual premium for each customer using available policyholder and vehicle-level data. The focus was on building a model that was both explainable and operationally deployable, with clear documentation for internal teams.

2. Objective

- To analyze the key drivers of car insurance premium pricing using statistical and machine learning techniques.
- To develop, validate, and interpret a regression model using Python libraries suitable for production use.
- To provide full model transparency to support internal pricing reviews and compliance documentation.
- To deliver a reusable codebase and a report that could be used both by technical staff and business decision-makers.

3. Data Used

The dataset consisted of **3,000 anonymized customer records**, each representing an active or recently renewed policy. The data was collected over a one-year period and included:

- Premium_Amount – actual annual premium paid by the customer (continuous, target variable)
- Customer_Age – age of the policyholder (continuous)
- Vehicle_Age – age of the car in years (continuous)
- Annual_Mileage – self-reported driving distance per year (continuous)

- Vehicle_Segment – segment of the car (A/B/C) (categorical)
- Accident_History – whether the customer had a reported accident in the last 2 years (binary categorical)
- City_Type – classification of customer's primary city (Tier 1 / Tier 2 / Tier 3)

Dummy variables were created for all categorical fields, resulting in 7 final independent variables and 1 interaction term.

4. Methodology

4.1 Data Preprocessing

- Checked for and addressed missing values using mean imputation for continuous variables and mode imputation for categorical variables.
- Encoded Vehicle_Segment, City_Type, and Accident_History using one-hot encoding.
- Created an **interaction variable between Vehicle_Age and Accident_History**, based on business input that accident history in older vehicles leads to higher claims.
- Identified and removed 12 outliers using a combination of **IQR and Cook's Distance**, which had a disproportionate impact on the regression line.

4.2 Model Development and Evaluation

- Built and compared two models:
    - **Multiple Linear Regression** (base model) using statsmodels
    - **Decision Tree Regressor** as a benchmark using sklearn
- Used **80/20 train-test split** with **5-fold cross-validation** to ensure stable performance.
- Evaluated both models on:
    - R-squared
    - RMSE
    - MAE
    - Residual normality (using QQ plot)
    - Homoscedasticity (residual scatter plots)

4.3 Tools & Environment

- **Jupyter Notebook** (delivered as .ipynb)

- Python Libraries: pandas, numpy, seaborn, matplotlib, scikit-learn, statsmodels
- Execution in both **Google Colab** and **VS Code**, with environment setup instructions

5. Key Results

- **Best Model**: Multiple Linear Regression
- **R-squared** (Test Set): **0.79**
- **RMSE**: ₹3,200
- **MAE**: ₹2,500
- **Key Drivers Identified**:
    - Vehicle_Age: Older vehicles led to higher premiums
    - Accident_History: Customers with prior accidents had an average ₹5,000 higher premium
    - City_Type: Tier 1 cities had higher premiums, likely due to higher repair costs
    - Vehicle_Segment_C: Larger segment cars attracted higher premiums
    - The interaction term between Vehicle_Age and Accident_History was statistically significant with **$p < 0.01$**

6. Delivered Report Summary

- **Technical Deliverables**:
    - Complete Python code with markdown comments, data preprocessing, model training, and visualization
    - Excel file with prediction outputs for test dataset
    - JSON dump of final model coefficients and metrics
    - Reusability notes and instructions for retraining with new data
- **Report (PDF, 8 pages)**:
    - Executive summary with 3 key takeaways
    - Tables showing model coefficients and variable significance
    - Visuals: Correlation heatmap, residual plot, predicted vs actual
    - Explanation of how to interpret the model outputs in business terms

    - Recommendations for internal use (pricing policy changes)
- **Bonus Deliverable**:
    - Sample email/slide template for internal team presentations, summarizing the model in simple language

7. Business Impact

- The pricing team used the model to identify underpriced Tier 1 customer segments and adjusted baseline premiums by ~7%
- The customer support team used the model insights to train agents on explaining premium changes to customers using data
- Enabled scenario planning: client could simulate how changing car segment or accident history would impact pricing
- Reduced over-reliance on legacy rule-based pricing templates and improved premium forecasting accuracy

Estimated **yearly impact**:

- $12–14 million recovered through improved pricing decisions
- 40% reduction in manual overrides by branch officers

8. Future Scope

- Move from linear to **non-linear models** like Random Forest or XGBoost for capturing hidden patterns
- Integrate the model with the firm's **internal CRM** using an API for real-time quote estimation
- Include **external factors** like fuel type, policy type (own damage/comprehensive), and claim frequency
- Conduct **model retraining every 6 months** to account for shifts in customer behavior and car trends
- Extend the same methodology to build **claim prediction models** as a next phase