**Digit Insurance Case Study**

***Case Study 1: Predicting Insurance Claim Severity***

**Scenario**

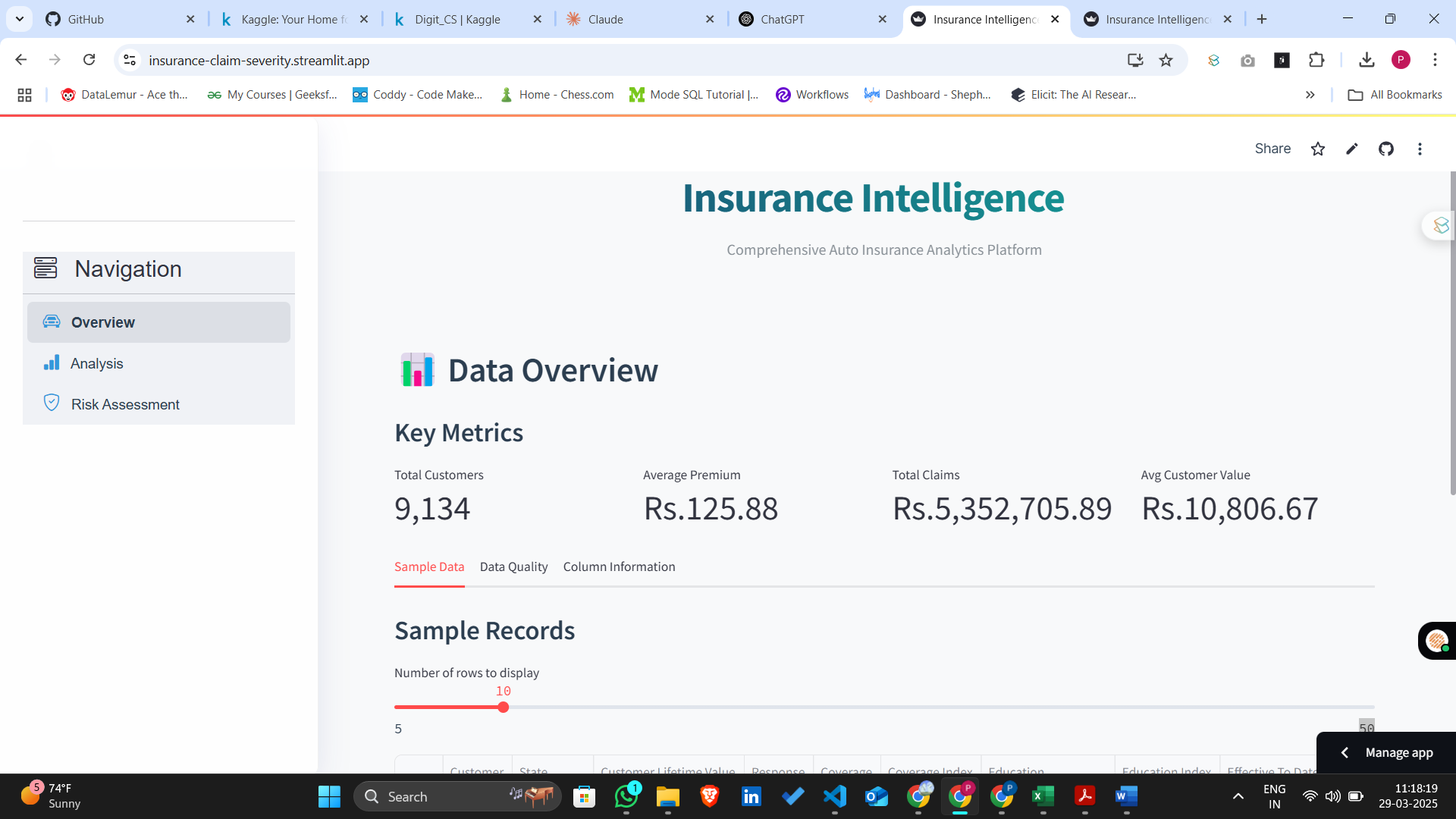
An insurance company is aiming to improve its underwriting process by better predicting the severity of claims. Historical data include claim amounts, policyholder demographics (e.g., age, gender, location), vehicle details, and previous claim history over the past five years. The goal is to develop a predictive model that can estimate claim costs accurately to support pricing decisions and risk management.

Deployed this Case Study: <https://insurance-claim-severity.streamlit.app/>

Code: <https://github.com/AestheticCoder-rjp/Insurance_Claim_Severity>

**Introduction**

Insurance companies need accurate claim severity predictions to enhance risk management and pricing strategies. This case study aims to develop a predictive model for claim costs using policyholder demographics, vehicle details, and historical claim data. By applying statistical and machine learning techniques, we aim to identify key predictors and improve underwriting decisions.



This is homepage of an Insurance Intelligence platform built using Streamlit. It provides an overview of auto insurance analytics, displaying key metrics like total customers (9,134), average premium (Rs.125.88), total claims (Rs.5,352,705.89), and average customer value (Rs.10,806.67). The left navigation panel includes options for Overview, Analysis, and Risk Assessment. A sample data table is also present, allowing users to adjust the number of displayed rows.

A green and white rectangle

AI-generated content may be incorrect.

A screenshot of a graph

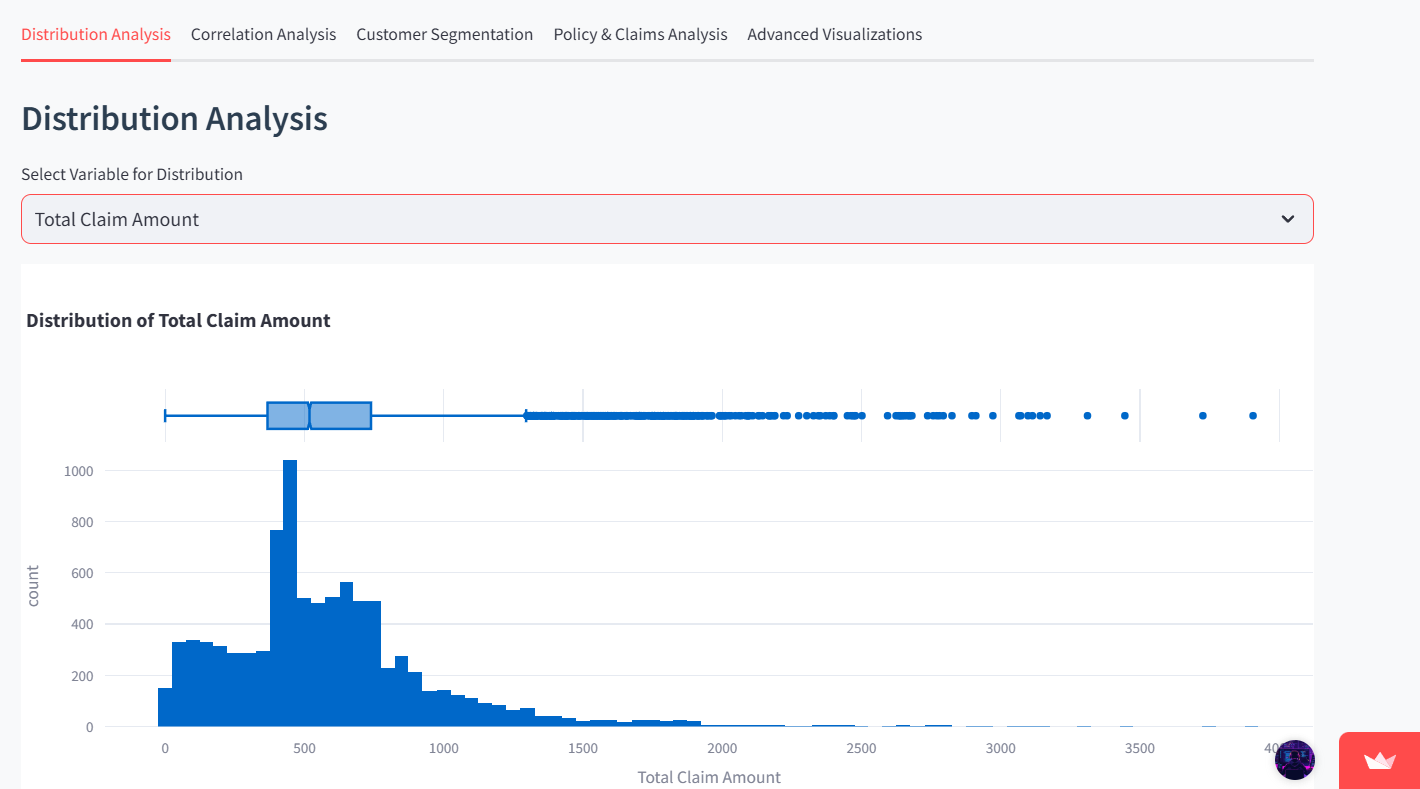
AI-generated content may be incorrect.

The image displays a data summary related to an insurance claim analytics platform. It presents a table listing various data columns along with their respective data types, including object, int64, and float64. Additionally, there is a pie chart on the right side visualizing the distribution of these data types. The chart indicates that int64 constitutes 50% of the dataset, object makes up 44.1%, and float64 accounts for 5.88%. This representation aids in understanding the dataset's structure before performing analysis.

A screenshot of a computer

AI-generated content may be incorrect.

**Methodology and Exploratory Data Analysis (EDA)**



This image focuses on distribution analysis of the Total Claim Amount, featuring a histogram and boxplot. The histogram shows a right-skewed distribution, indicating that most claims are on the lower end while a few high-value claims act as outliers. The boxplot confirms the presence of several extreme values, making it crucial to handle outliers for effective model training.

A screenshot of a graph

AI-generated content may be incorrect.

The image displays a geographic distribution analysis. It includes a bar chart showing the number of claims across different locations—Suburban, Rural, and Urban—where Suburban has the highest count. Below the chart, a numerical breakdown highlights the exact values: 5,779 for Suburban, 1,773 for Rural, and 1,582 for Urban.

A screenshot of a computer

AI-generated content may be incorrect.

The image shows an "Insurance Intelligence" dashboard for auto insurance analytics. It focuses on "Exploratory Data Analysis & Insights" under the "Customer Segmentation" tab. A pie chart visualizes customer demographics based on marital status, with three categories: Married (58%), Single (27%), and Divorced (15%).

A screenshot of a computer

AI-generated content may be incorrect.It features an interactive navigation panel with options like "Overview," "Analysis," and "Risk Assessment." The main content includes visualizations such as a scatter plot for "Premium vs Claims by Policy Type" and a box plot for "Claim Amount by Vehicle Class."

A screenshot of a graph

AI-generated content may be incorrect.

It showcases an Exploratory Data Analysis & Insights dashboard under the Advanced Visualizations section. It features a Sunburst Plot representing the Claim Amount distribution by State and Vehicle Class. The hierarchical visualization breaks down claims by state (e.g., California, Arizona, Oregon) and further into vehicle categories like SUVs, Two-Door Cars, and Passenger Cars. This visualization helps in understanding claim patterns across different regions and vehicle types.

A screenshot of a computer

AI-generated content may be incorrect.

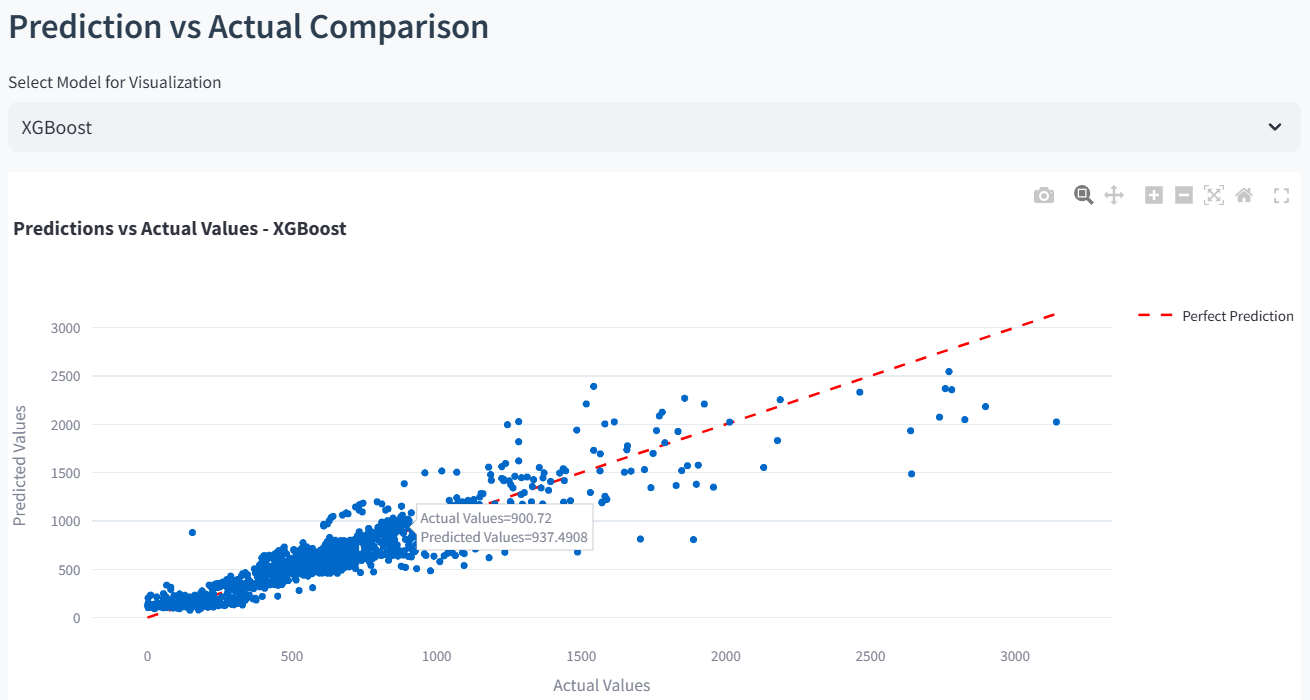
Id displays a hierarchical treemap visualization. It represents the breakdown of total claim amounts by policy type and coverage, distinguishing categories like "Personal Auto," "Corporate Auto," and "Special Auto." The treemap helps in identifying which policy type and coverage combinations result in higher claims.

**Model Building**

A screenshot of a computer

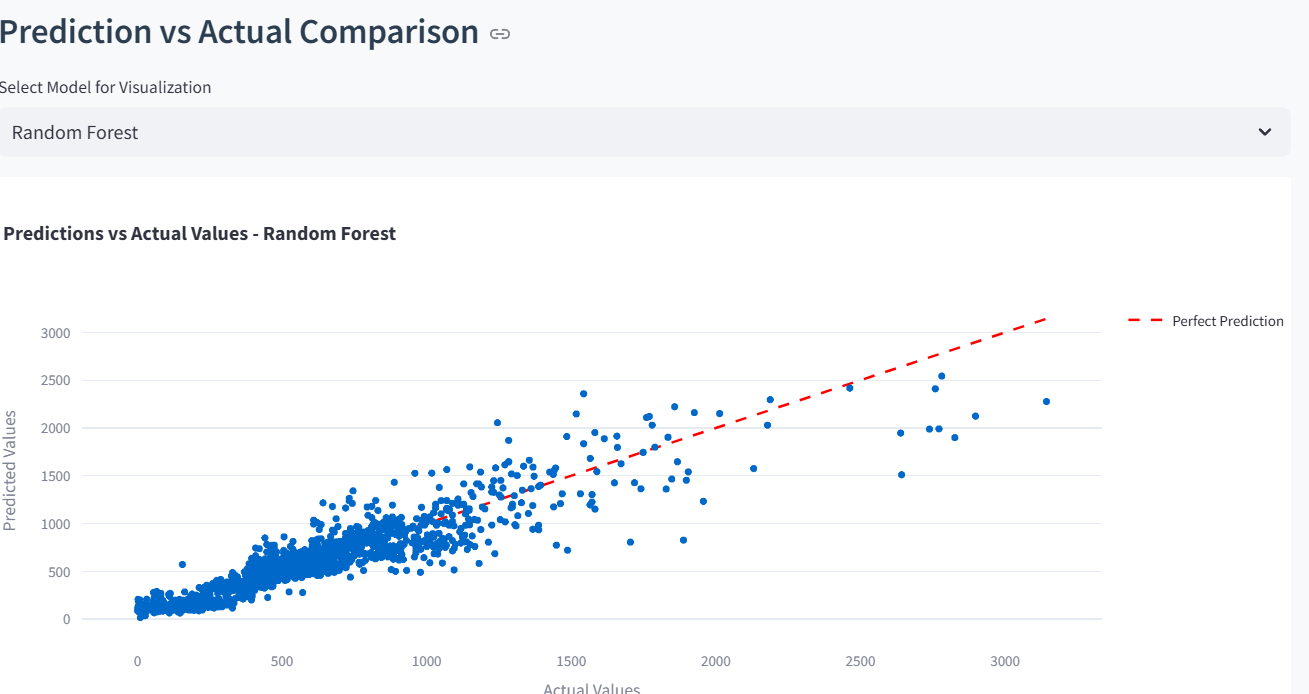
AI-generated content may be incorrect.

Based on the feature importance graph, emphasize which features had the most impact (e.g., Monthly Premium Auto and Location\_Suburban had the highest importance).



The image is a scatter plot comparing predicted vs. actual values for an XGBoost model. The plot include

* Blue dots representing actual vs. predicted values.
* A red dashed line indicating a perfect prediction (y = x).
* A tooltip showing an example point where the actual value is 900.72 and the predicted value is 937.49, indicating some prediction error.



The image presents a Prediction vs. Actual Comparison scatter plot for a Random Forest model. Blue dots represent actual vs. predicted values, with a red dashed line indicating perfect predictions. The spread of points shows the model's performance, with some deviations from the ideal line. The general trend suggests that the model captures the pattern but with some prediction errors.

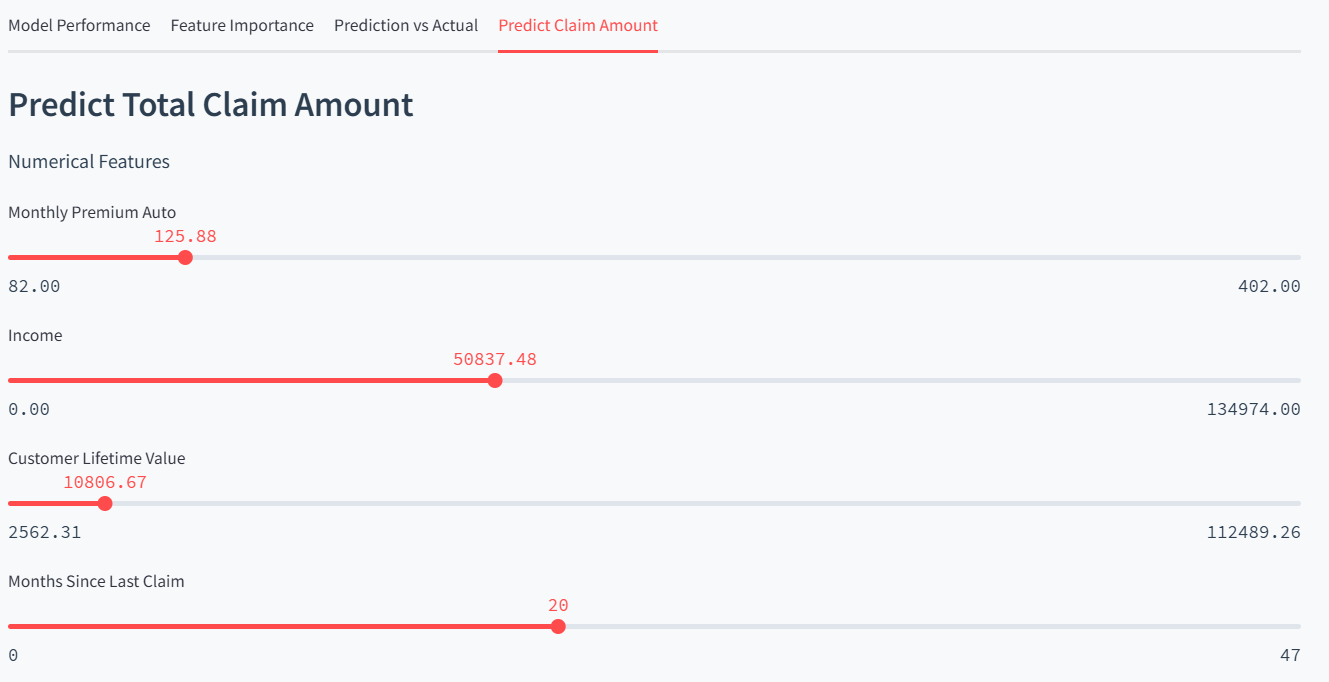
**Interpretation & Recommendations**



A screen shot of a computer

AI-generated content may be incorrect.

The image displays the Model Performance Metrics for two machine learning models, XGBoost and Random Forest, used to predict insurance claim amounts. It includes three key performance indicators: R² Score, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). Both models show similar R² scores (~0.849), indicating strong predictive performance. However, Random Forest has a slightly lower MAE, suggesting better average prediction accuracy.



A screenshot of a facebook account

AI-generated content may be incorrect.

The images depict a Claim Amount Prediction Interface using machine learning models. The image shows categorical feature selections (Location, Gender, and Marital Status) and predicted claim amounts from XGBoost and Random Forest, which are nearly identical. The another image displays numerical feature inputs like Monthly Premium Auto, Income, Customer Lifetime Value, and Months Since Last Claim used for prediction. The interface allows users to adjust input values to observe changes in predicted claim amounts.

**Conclusion**

Our predictive model for insurance claim severity demonstrates the potential of machine learning in enhancing risk assessment and pricing strategies. By leveraging policyholder demographics, vehicle details, and historical claim data, we identified significant predictors such as Monthly Premium Auto, Location, and Coverage Type.

The **XGBoost and Random Forest models** achieved strong predictive performance, with **R² values around 0.85**, indicating a good fit. However, the model's performance could be further improved by incorporating **external factors** such as weather conditions, accident history, and economic indicators.

Key takeaways include:

* **Higher Monthly Premiums** correlate with increased claim amounts.
* **Suburban policyholders** tend to file higher claims than urban or rural ones.
* **Longer policy durations** show a trend of higher claims.

For **business implications**, insurers can refine premium pricing, adjust underwriting policies, and offer incentives to low-risk customers based on these insights. Future enhancements should explore **deep learning models** and **additional external data sources** for even more accurate claim predictions.