Regression with an Insurance Dataset

By: Potri Abhisri Barama

DS 340: Fundamentals and Principles of Data Science
05/07/2025

Project Overview

Purpose

- To gain hands-on experience implementing a machine learning model using the “Regression with an Insurance Dataset” competition.
- To build a model that predicts insurance premium amounts based on a variety of customer and policy-related features.

Objectives

- To preprocess a large dataset by handling missing values through customized imputation strategies.
- To transform and encode categorical variables effectively for machine learning readiness.
- To train a regression model, evaluate its performance using standard metrics, and analyze the results to identify potential improvements.

Preprocessing Pipeline

Data Cleaning

- Dropped irrelevant columns such as id before model training
- Converted Policy Start Date from string to datetime format
- Ensured all categorical and numerical columns had consistent data types
- Checked for and removed rows with missing target values (Premium Amount) in the training set

Handling Missing Values

- Replaced missing Age, Annual Income, and Credit Score with the median
- Used the mode to impute missing values for Marital Status, Customer Feedback, and Occupation (grouped by education where needed)
- Imputed missing Exercise Frequency using the most common category
- Filled Previous Claims with 0, assuming no claim history

Feature Engineering

- Extracted Policy Year, Month, and DayOfWeek from the Policy Start Date
- Encoded categorical variables using:
  - Binary encoding for Gender and Smoking Status
  - Ordinal encoding for ordered categories like Education Level and Customer Feedback
- Mapped high-cardinality features like Occupation, Location, and Property Type to numeric codes
- Dropped the original Policy Start Date column after extraction

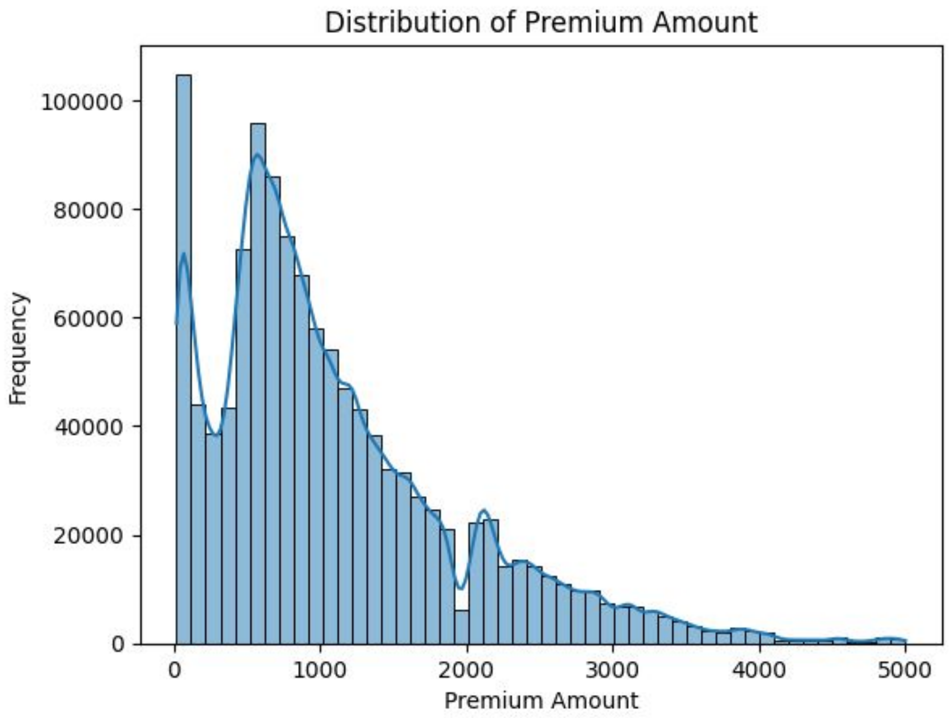
Model Selection

- RandomForestRegressor was chosen for its robustness with mixed feature types, ability to handle non-linear relationships, and minimal need for preprocessing
- Also, performs well even with minimal hyperparameter tuning
- Chose conservative parameters (max_depth=15, n_estimators=50, min_samples_leaf=10) to manage memory usage

Exploratory Data Analysis

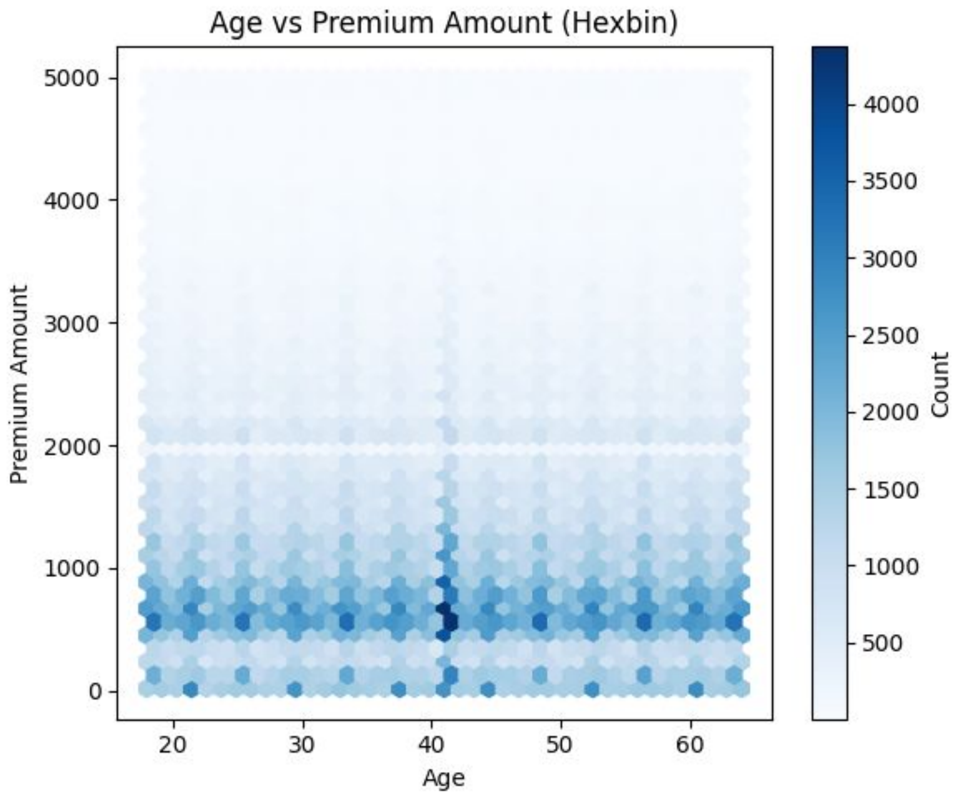
Distribution of Premium Amount

- This histogram with KDE (kernel density estimate) displays the distribution of Premium Amount values across the training dataset:
- The distribution is right-skewed (positively skewed), meaning:
    - A large number of policies have low premium amounts (e.g., under $1000)
    - Fewer policies have high premium amounts, with a long tail extending past $4000
- There are multiple local peaks, indicating potential clusters or price tiers in the data
- This kind of skew is typical for financial or pricing data, and it may suggest the need for log transformation during modeling



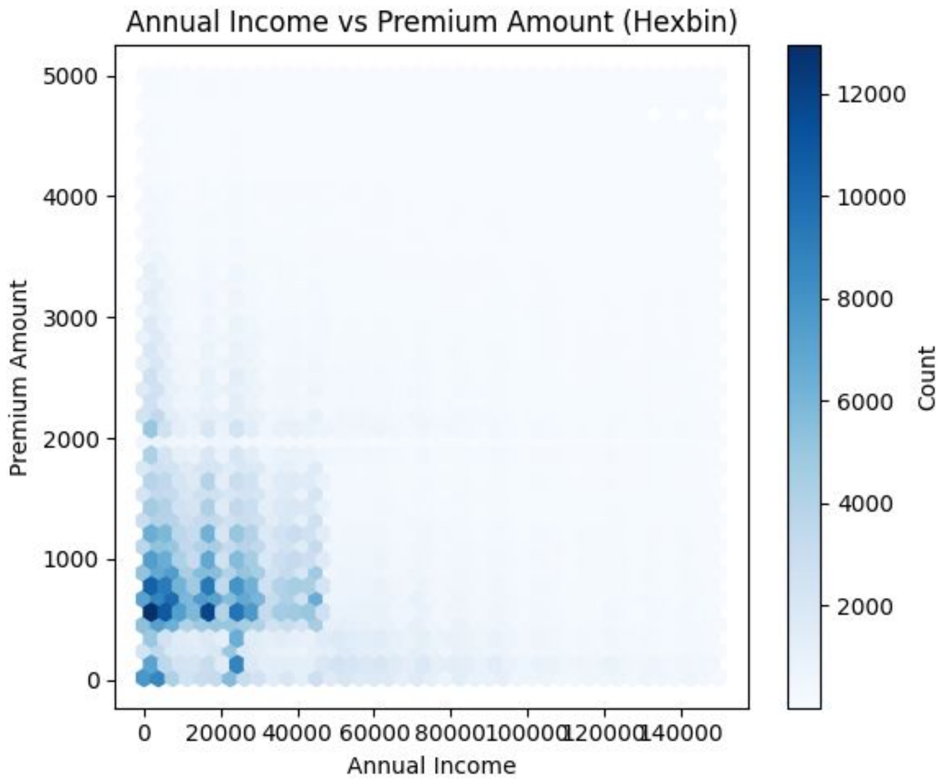
Age vs Premium Amount

- Most policyholders are aged 25 to 45, with premiums clustering between \$500–\$1500.
- The vertical lines suggest age values may be binned or rounded, especially around ages like 40.
- Premium amounts show horizontal bands at common values (e.g., \$500, \$1000), indicating possible pricing tiers.
- There's no clear trend between age and premium, suggesting age alone may not be a strong predictor.



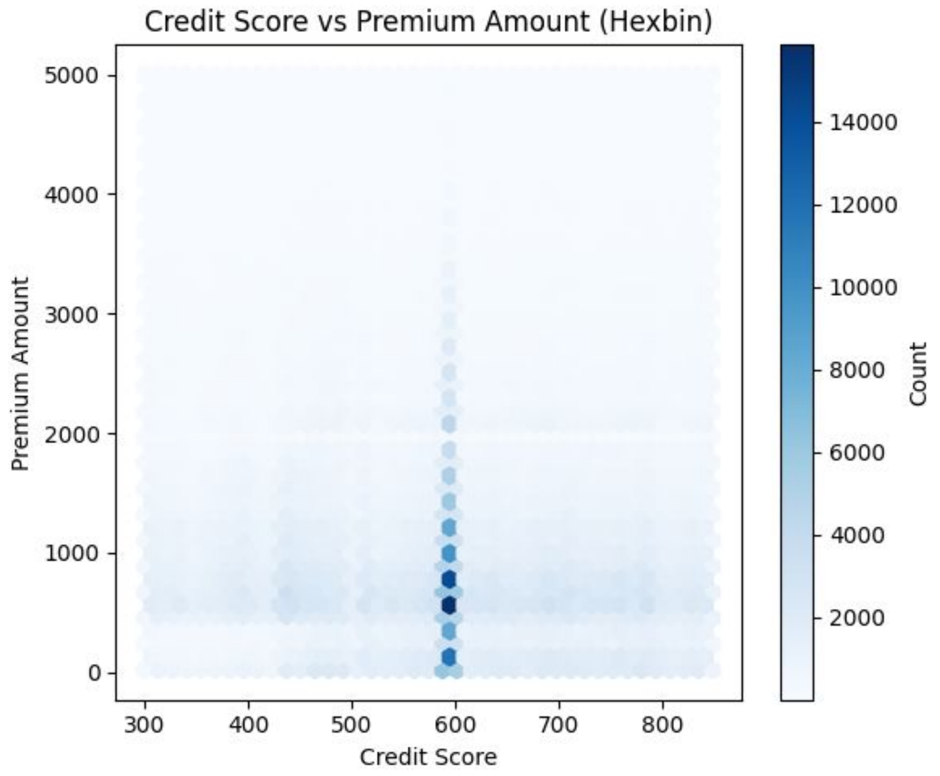
Annual Income vs Premium Amount

- The majority of policyholders earn less than $50,000, and most premiums fall between $500–$1500.
- A dense cluster exists at low income and low premium ranges, indicating the model has most of its training data in that zone.
- Higher incomes don't clearly correlate with higher premiums — premiums remain fairly consistent across income brackets.
- The horizontal banding again suggests that premium values may be tiered or pre-binned.



Credit Score vs Premium Amount

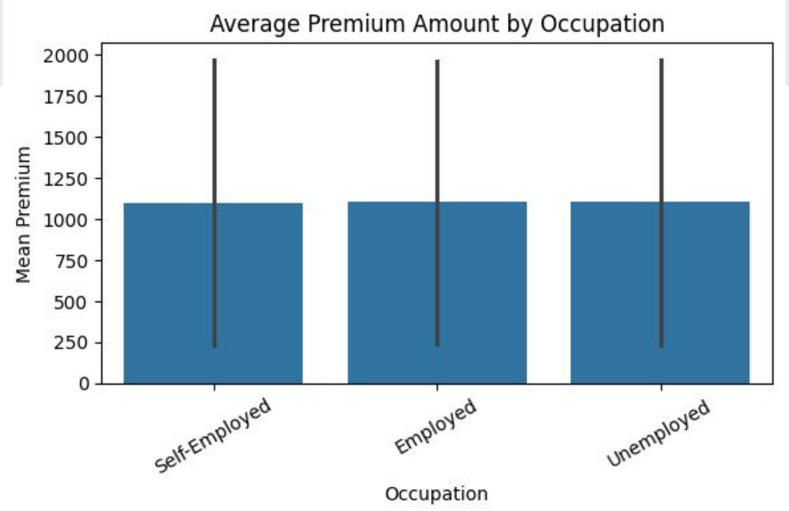
- The plot shows a sharp vertical band centered around a credit score of ~600, suggesting that many records have similar or identical scores.
- Most premiums again cluster below $1500, regardless of credit score.
- There's no clear upward or downward trend, indicating that credit score may have limited direct influence on premium amounts in this dataset.
- The uniform vertical pattern may suggest that credit score was discretized or imputed in many rows.

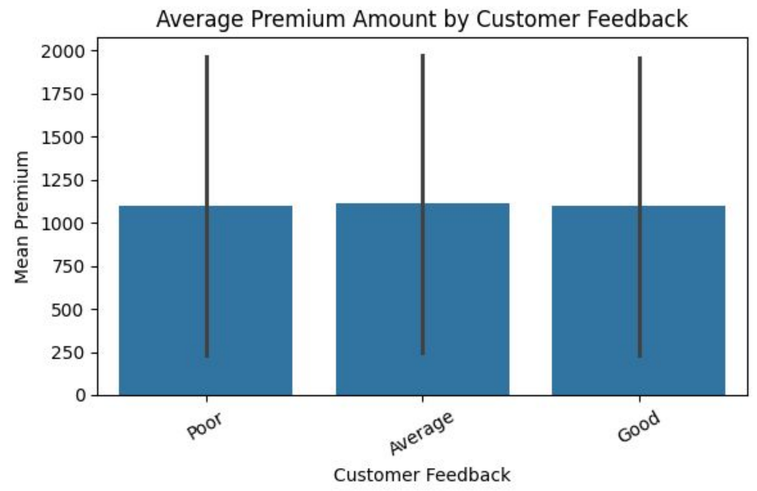


Avg. Premium by Features

- The average premium is fairly consistent across employment types.
- This suggests that occupation may not be a strong driver of premium variation in the dataset.
- The large error bars indicate high variance, meaning individual premiums vary widely within each group.

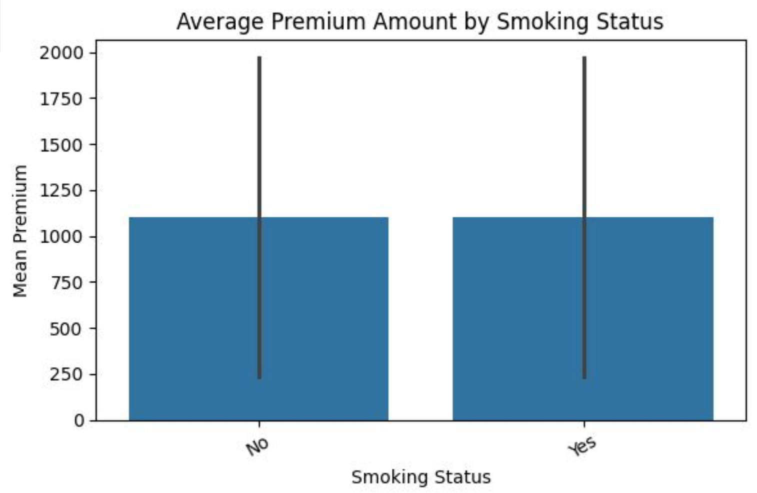
- Customers with Poor, Average, and Good feedback have similar average premiums.
- The lack of clear separation implies customer feedback may have limited direct influence on premium pricing.
- High error bars again reflect variation in premiums within each feedback category.

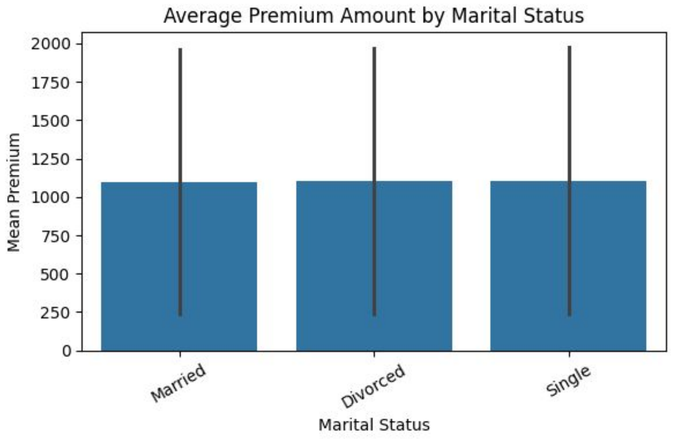


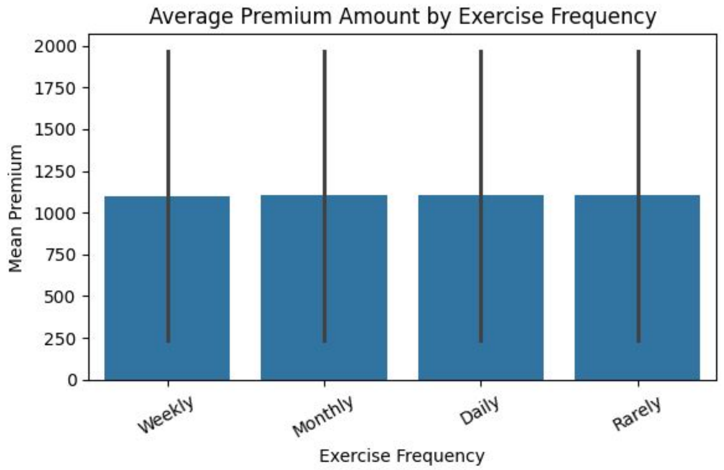


Avg. Premium by Features

- Across all three features, the average premium amounts are very similar between categories, showing no strong influence on pricing.
- Error bars indicate high variability within each group.
- These features show limited individual predictive power.

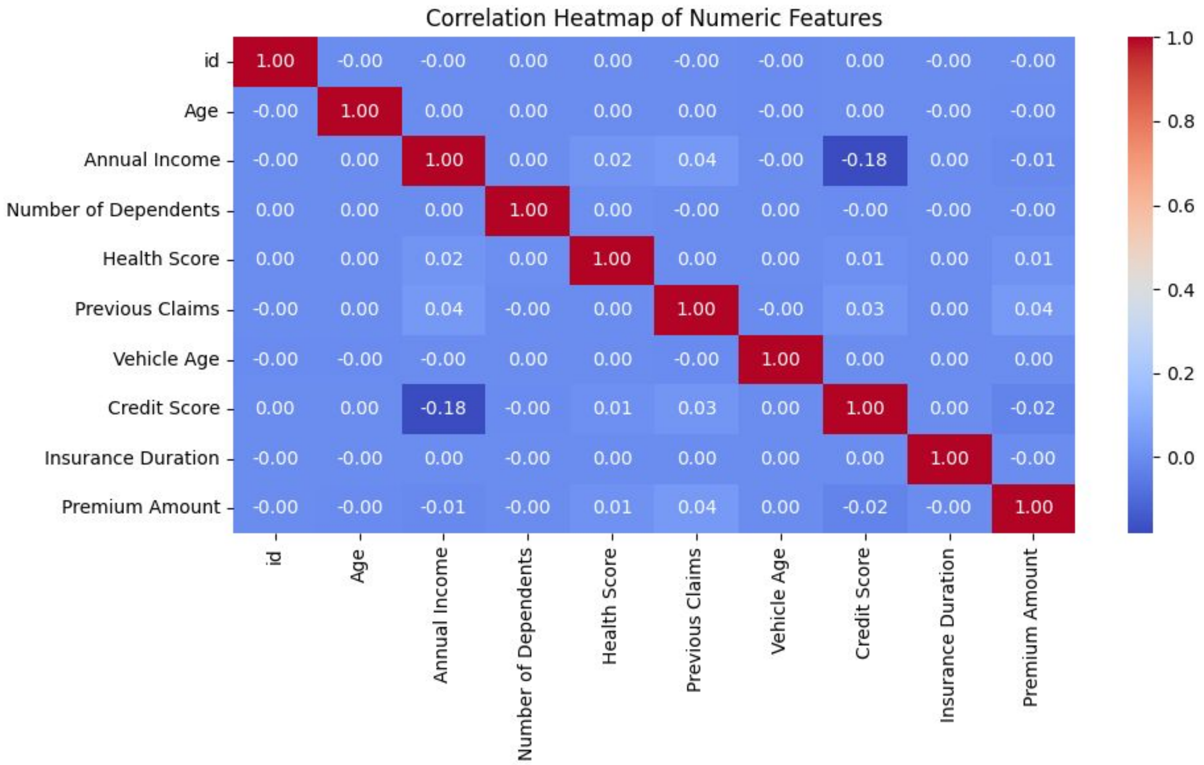






Correlation Heatmap of Numeric Features

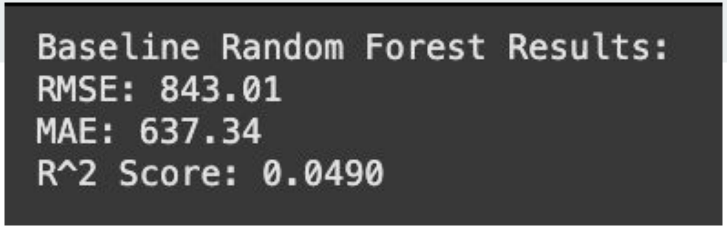
- Premium Amount has very weak correlations with all numeric features, suggesting limited linear influence.
- Previous Claims has the highest correlation with the target, but still only around 0.04.
- Credit Score shows a slight negative correlation with Annual Income (−0.18).
- These low correlations highlight the need for non-linear modeling or feature interaction to improve prediction.



	id	Age	Annual Income	Number of Dependents	Health Score	Previous Claims	Vehicle Age	Credit Score	Insurance Duration	Premium Amount
id	1.00	-0.00	-0.00	0.00	0.00	-0.00	-0.00	0.00	-0.00	-0.00
Age	-0.00	1.00	0.00	0.00	0.00	0.00	-0.00	0.00	-0.00	-0.00
Annual Income	-0.00	0.00	1.00	0.00	0.02	0.04	-0.00	-0.18	0.00	-0.01
Number of Dependents	0.00	0.00	0.00	1.00	0.00	-0.00	0.00	-0.00	-0.00	-0.00
Health Score	0.00	0.00	0.02	0.00	1.00	0.00	0.00	0.01	0.00	0.01
Previous Claims	-0.00	0.00	0.04	-0.00	0.00	1.00	-0.00	0.03	0.00	0.04
Vehicle Age	-0.00	-0.00	-0.00	0.00	0.00	-0.00	1.00	0.00	0.00	0.00
Credit Score	0.00	0.00	-0.18	-0.00	0.01	0.03	0.00	1.00	0.00	-0.02
Insurance Duration	-0.00	-0.00	0.00	-0.00	0.00	0.00	0.00	0.00	1.00	-0.00
Premium Amount	-0.00	-0.00	-0.01	-0.00	0.01	0.04	0.00	-0.02	-0.00	1.00

Model Performance Metrics

Baseline Random Forest Results



```
Baseline Random Forest Results:
RMSE: 843.01
MAE: 637.34
R^2 Score: 0.0490
```

RMSE (Root Mean Squared Error): 843.01

This measures the average size of the errors, giving more weight to large errors. An RMSE of ~843 means the model’s predictions are off by about $843 on average, with larger mistakes penalized more heavily.

MAE (Mean Absolute Error): 637.34

This gives the average magnitude of all prediction errors, treating each equally. On average, the model’s predictions differ from the actual premiums by about $637, regardless of over or underestimation.

R^2 Score (R-squared): 0.0490

This indicates that the model explains only 4.9% of the variance in the premium amounts. A perfect model scores 1.0; a score near 0 means the model performs only slightly better than predicting the mean for all cases.

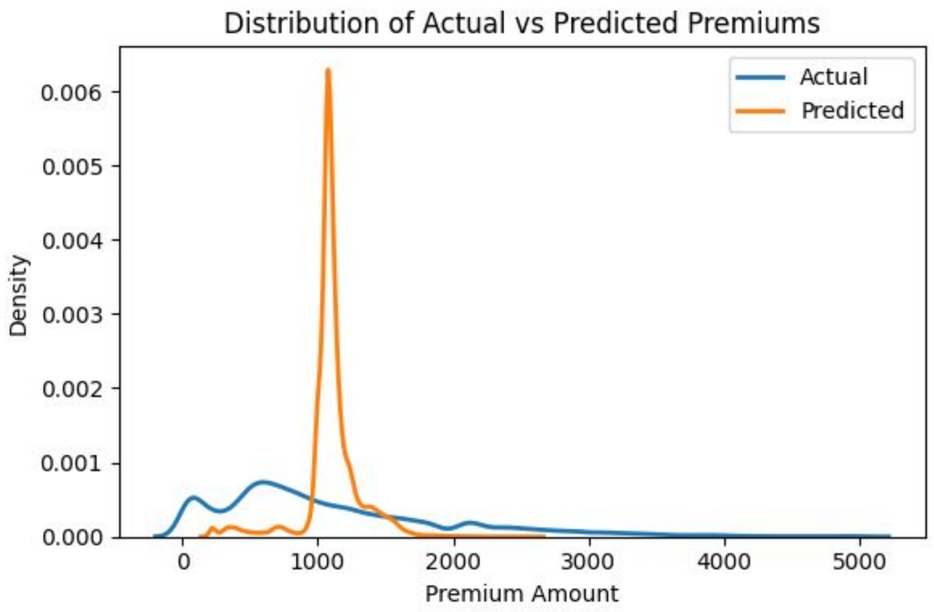
Distribution of Actual vs Predicted Premiums

The blue curve (Actual) is much wider and more spread out, showing that real premium amounts vary significantly across the dataset.

The orange curve (Predicted) is sharply peaked around a narrow range (near $1000), indicating the model predicts similar values for most cases.

This mismatch shows the model is not capturing the true variability in the data — it tends to play it safe and outputs values near the mean.

The narrow prediction range explains the low R² score and the model's tendency to underfit.



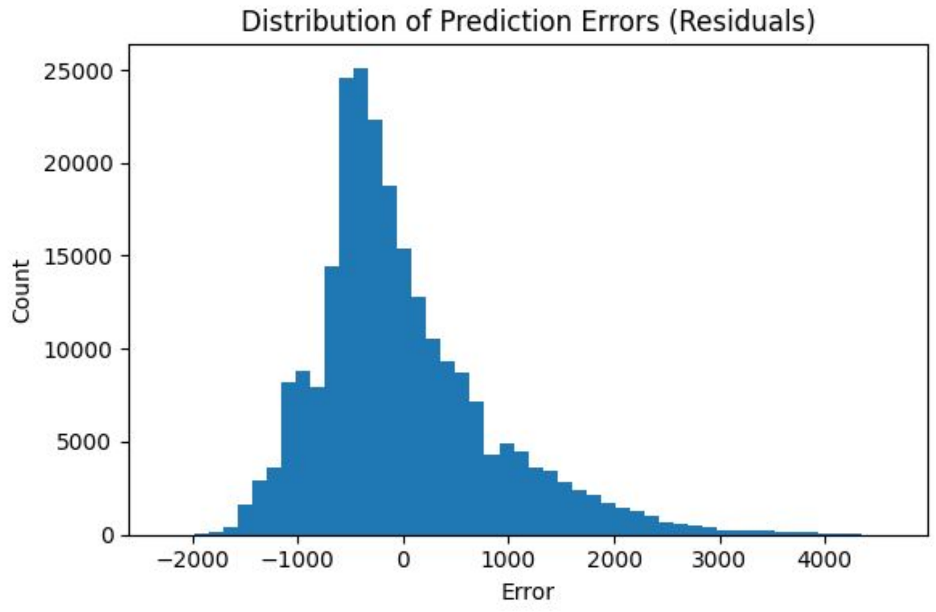
Distribution of Prediction Errors

Most prediction errors fall between -1000 and +1000, indicating the model is usually off by around $1000 or less.

The peak left of zero (around -500) means the model tends to underpredict premium amounts more often than it overpredicts.

The long right tail suggests that when the model does overpredict, it can make larger mistakes.

The overall shape is skewed, not centered at zero, reinforcing that the model has bias and isn't well-calibrated.



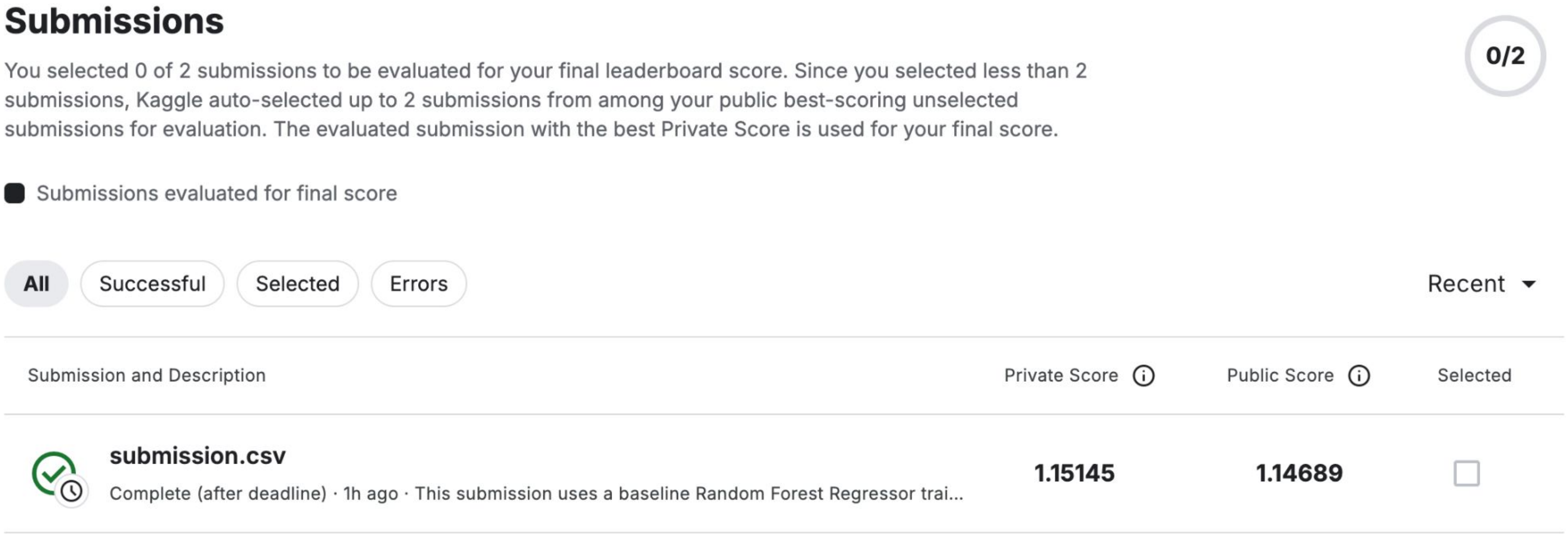
Kaggle Competition & Conclusion

Final Prediction and Kaggle Submission

After training the Random Forest model on the preprocessed training data, I used it to predict premium amounts for the unseen test data provided by Kaggle. Before prediction, I ensured the test data was cleaned and encoded using the exact same steps applied during training (e.g., handling missing values, encoding categorical variables, and extracting date features).

Once the predictions were generated, I created a submission file containing two columns: id and the predicted Premium Amount. This file matched the format required by Kaggle — a CSV with 800,000 rows. Finally, I uploaded the submission through the Kaggle competition page to receive the evaluation score based on the private and public leaderboard.

Kaggle Competition Score: ~1.15



Submissions

You selected 0 of 2 submissions to be evaluated for your final leaderboard score. Since you selected less than 2 submissions, Kaggle auto-selected up to 2 submissions from among your public best-scoring unselected submissions for evaluation. The evaluated submission with the best Private Score is used for your final score.

0/2

■ Submissions evaluated for final score

Submission and Description	Private Score	Public Score	Selected
submission.csv Complete (after deadline) · 1h ago · This submission uses a baseline Random Forest Regressor trai...	1.15145	1.14689	☐

Handling Data, Model, and Resource Challenges

Data Preprocessing Issues

- Handling missing values was a key challenge. Several features contained null entries, including Exercise Frequency, Occupation, and Credit Score. To address this, custom imputation strategies were applied—such as using the median for numerical features and mode for categorical ones.

Model Performance Limitations

- The baseline RandomForestRegressor showed poor performance, with an R^2 score of just 0.049. The model struggled to capture the variance in premium amounts, often predicting values close to the mean. This underperformance highlights the need for more advanced modeling techniques, feature engineering, or hyperparameter tuning in future iterations.

Computational Constraints

- Initial attempts to train the model using 100 trees (the default setting) consumed excessive RAM and took too long to run in the Google Colab environment. To overcome this, the number of trees was reduced to 50 and the maximum depth was decreased. While this may have contributed to underfitting, it allowed the model to successfully train on the large dataset within resource limits.

Thank You