DATA 1030 Final Report

Health Insurance Cross Sale Prediction

Yuyan Fan

12/14/2024

Brown DSI

https://github.com/yuu930/1030_Final.git

I. Introduction

Cross-selling is a sales strategy where a business encourages customers to purchase additional products or services that are related or complementary to the primary purchase. The goal of my project is to predict whether an existing health insurance customer will purchase vehicle insurance or not. Building a high-accuracy model for this is important because it allows the company to accordingly plan its communication strategy to reach out to those customers and optimize its business model and revenue. It is a binary classification problem. The data set is from kaggle[1]. My data set contains 1 binary target variable 'Response' (1 = interested, 0 = not interested). There are 11 predictor features and 381109 rows of data points.

Feature Name	**Type**	**Description and unit**
Age	continuous	Customer age. unit:years
ID	categorical	Unique identifier for the Customer.
Gender	categorical	0: male, 1: female
Driving license	categorical	0:not licensed, 1: licensed
region_code	categorical	Unique code for the region of the customer
previously_insuranced	categorical	0: no, 1: yes
Vehicle_Age	ordinal	3 age groups: < 1 Year, 1-2 Year, > 2 Years
Vehicle_Damage	categorical	1: yes, 0: no
Annual_Premium	continuous	The amount the customer needs to pay as premium in the year. unit:dollar
Policy_Sales_Channel	categorical	Anonymized Code for the channel of outreaching to the customer ie. Over Mail, In Person, etc

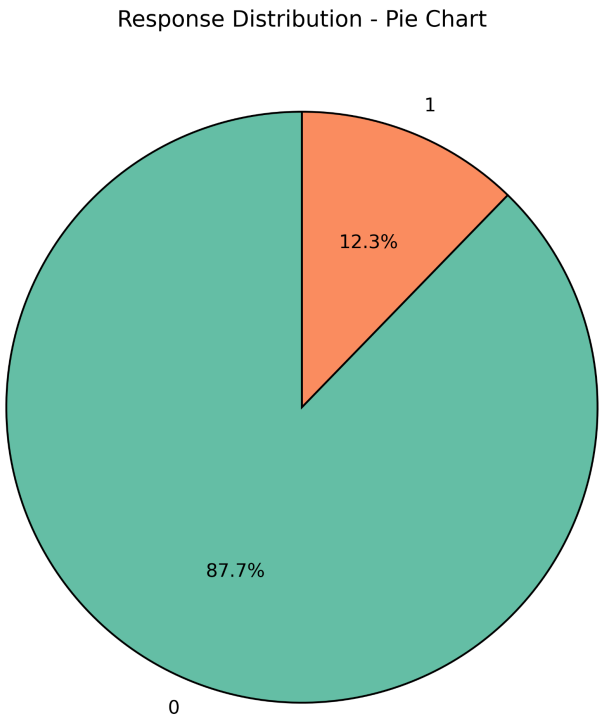
Vintage	continuous	Number of days, customer has been associated with the company.

Table 1. Dataset features and descriptions.

Previous research has explored similar problems in the health insurance domain, focusing on cross-sell predictions. One study by Khulekani Mavundla et al applied Random Forest, K-Nearest Neighbors, Logistic Regression, and XGBoost to predict cross-selling opportunities for health insurance products. The Random Forest model achieved the highest accuracy (99%) and an F1 score of 1.00 [2]. Similarly, a study by Manoj Patil utilized machine learning techniques, including logistic regression and normalization, to predict vehicle insurance interest [3], highlighting the role of preprocessing and algorithm selection in achieving reliable results.These studies demonstrate that machine learning models, particularly Random Forest and XGBoost, are highly effective in this domain, achieving high accuracy and F1 scores. However, each model's predictive power is contingent on preprocessing steps, feature selection, and handling of imbalanced data.

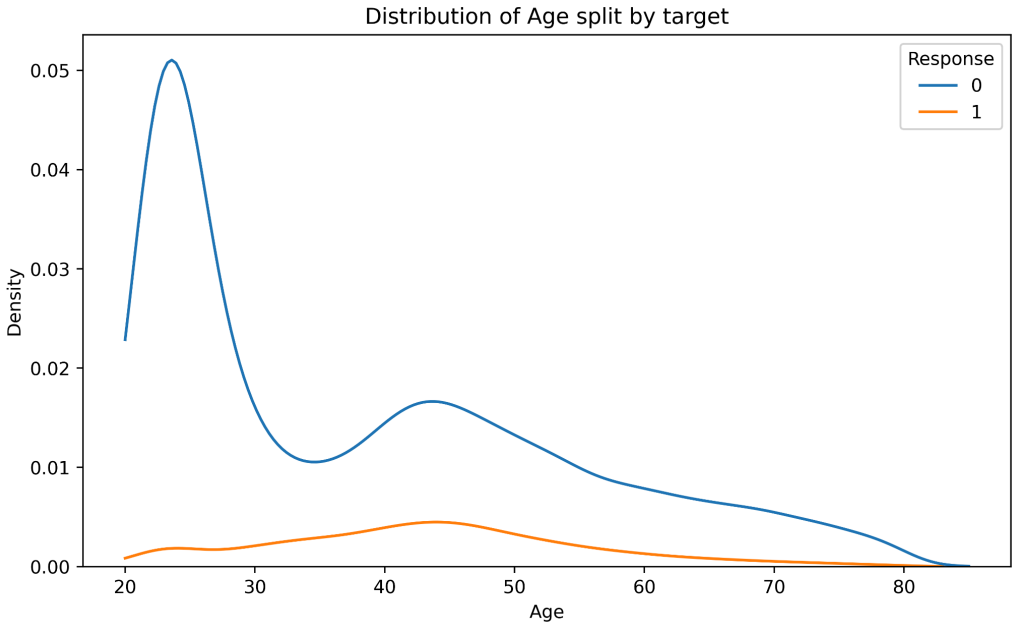
II. EDA

The dataset does not contain any missing values. The biggest challenge is the data size, since it has over 380k rows of data points. The dataset is IID. Each row is the information of an independent user, and there is no structured data or time series data here. During the EDA process, I found something interesting. The data set is highly imbalanced. As shown in the plot below, 87.7% of the data belongs to class 0. This will bring a big impact on my following pipeline and processing.



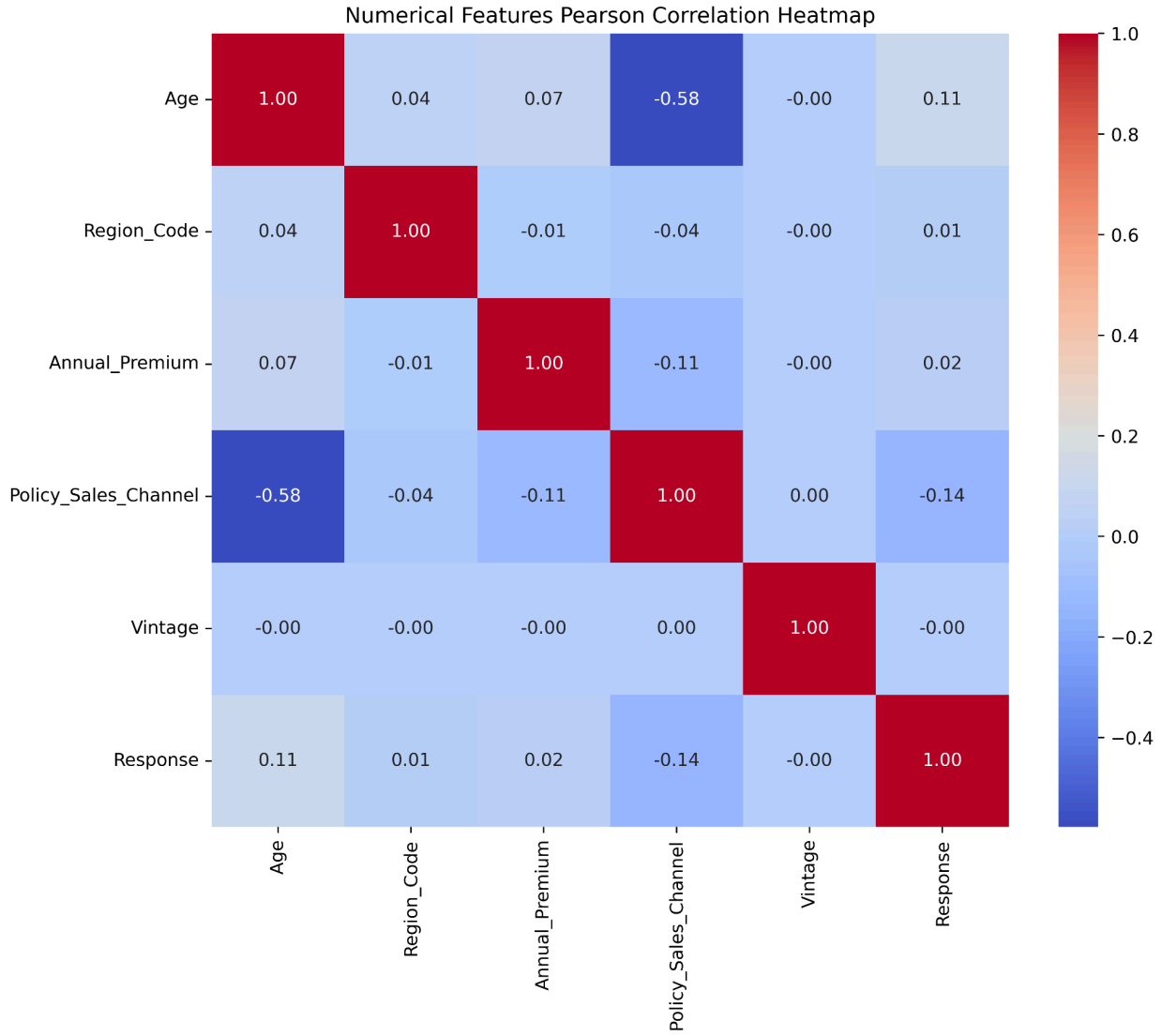
Plot 1. Pie chart showing the distribution of the target variable, “Response.”

I also found that there is an age preference contrast during the EDA process. According to the age distribution based on target variable split, we can see that higher density of response 0 in younger age groups, with a decreasing trend as age increases.



Plot 2. Density plot of age distribution split by target response.

The heatmap reveals weak linear correlations between most numerical features and the target variable (Response). Among them, Age shows a slight positive correlation (0.11), indicating older customers are marginally more likely to respond, while Policy_Sales_Channel exhibits a small negative correlation (-0.14), suggesting certain sales channels are less effective for positive responses. Additionally, Policy_Sales_Channel and Age have a strong negative correlation (-0.58), highlighting a potential interaction effect. Overall, the numerical features exhibit minimal direct linear relationships with the target variable.



Plot 3. Heatmap for the Pearson correlation between numerical features and target variable.

III. Methods

The data splitting strategy began with an 80-20 train-test split, employing stratified sampling to ensure that the class distributions in the test and 'other' sets remained consistent with the overall data. To further assess the model's robustness and account for uncertainty due to data splitting, the process was repeated for five different random states. Additionally, within each 'other' set, Stratified K-Fold Cross-Validation (with 4 folds) was applied, allowing consistent performance evaluation while mitigating bias and variance. For data preprocessing, ColumnTransformer was used to handle different types of features effectively. Ordinal Encoding was applied to the 'Vehicle_Age' feature, maintaining its natural hierarchy (< 1 Year, 1-2 Year, > 2 Years). One-Hot Encoding was used for categorical variables like 'Gender' and 'Vehicle_Damage,' while Min-Max Scaling was applied to the 'Age' feature to normalize it between 0 and 1. Finally, Standard Scaling was employed for numerical features such as 'Region_Code,' 'Annual_Premium,' 'Policy_Sales_Channel,' and 'Vintage' to ensure they had zero mean and unit. variance, enabling consistent treatment across all numeric inputs.

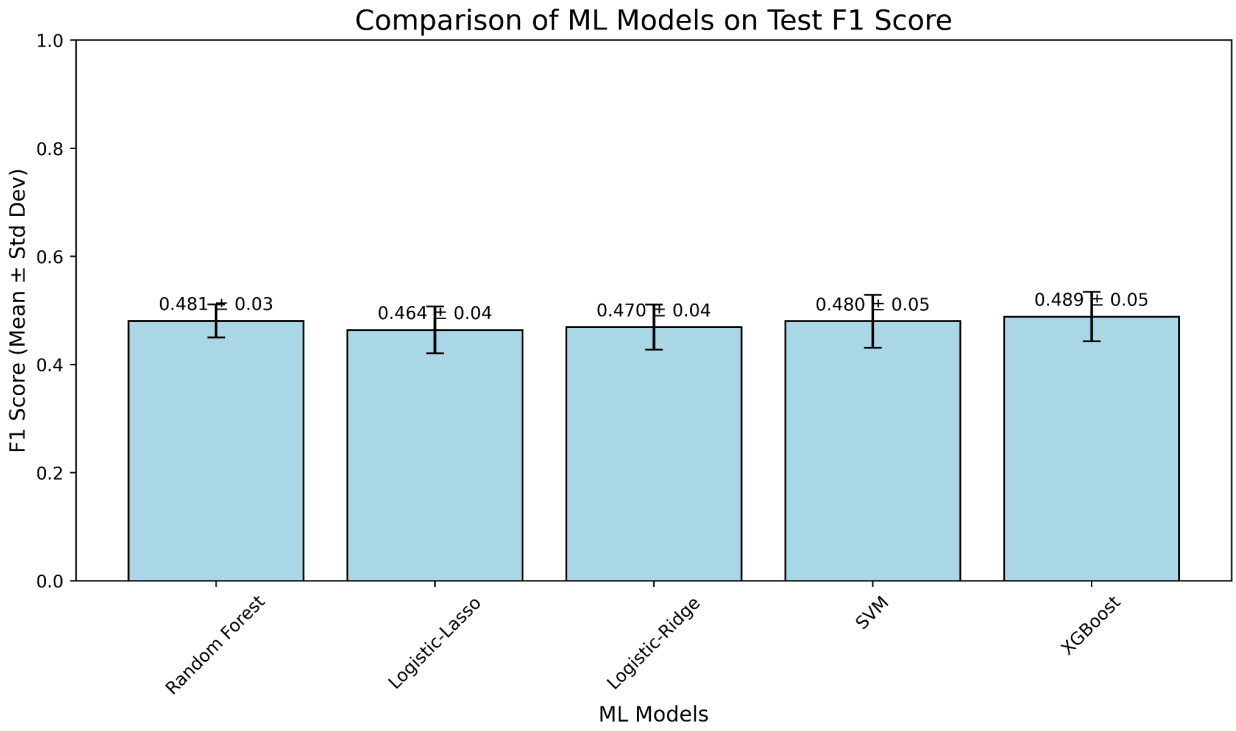
The pipeline combined preprocessing with Grid Search Cross-Validation (GridSearchCV) to tune hyperparameters for multiple machine learning algorithms. I tried 5 different algorithms and tuned their parameters respectively. The table below summarizes the algorithms and their candidate parameters I tried. The best parameters are colored red.

Algorithms	**Parameters**
Logistic - l1	'ML_algo__C': [0.1, 1, 10, 100,200]
Logistic - l2	'ML_algo__C': [0.1, 1, 10, 100,200]
SVM	'ML_algo__C': [1e-1, 1e0, 1e1, 1e2,1e3], 'ML_algo__gamma': [1e-2, 1e-1, 1e0, 1e1]
Random Forest	'ML_algo__max_depth': [3,5,7,10,1], 'ML_algo__max_features': [0.2,0.4,0.6,0.8,1.0]
XGBoost	'ML_algo__max_depth': [1,3,5, 10], 'ML_algo__learning_rate': [0.001,0.01, 0.1, 0.2], 'ML_algo__n_estimators': [50,75, 100, 150, 200]

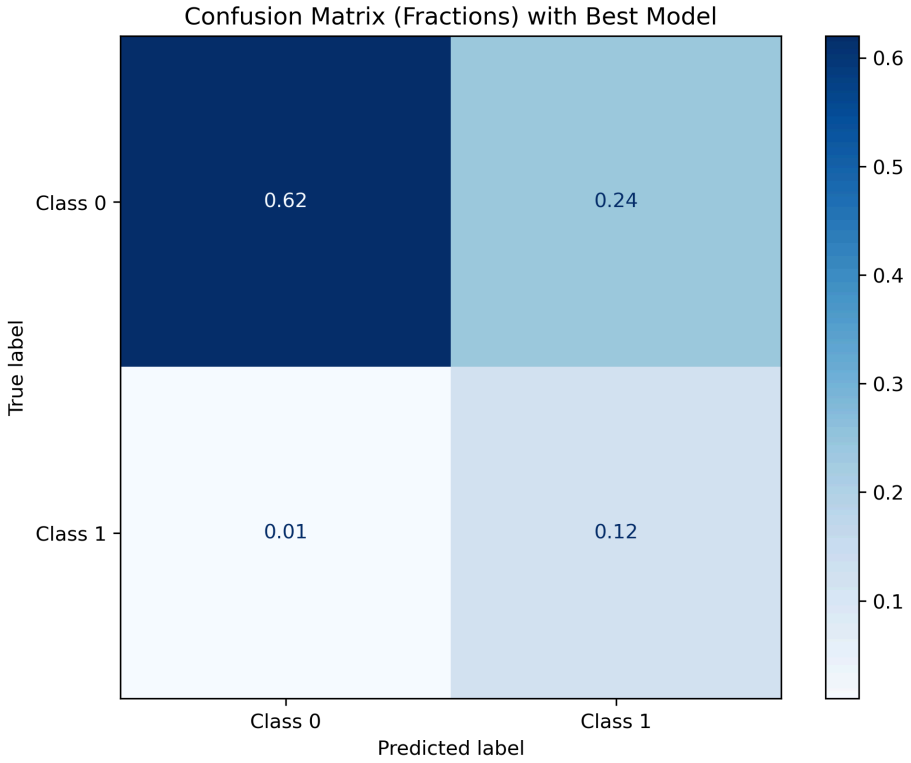
The evaluation of model performance was centered on the F1-Score, a balanced measure of precision and recall, which is particularly important for imbalanced datasets where accuracy alone can be misleading. To handle the imbalance data, I set the class_weight='balanced' here for all algorithms except the XGBoost. I set its own parameter ‘scale_pos_weight’ to be 9, since the ratio of class 0 to class 1 is almost 9:1. Finally, a Confusion Matrix was presented in normalized form to provide intuitive insights into model performance across classes, and class-specific metrics such as precision, recall, and F1-scores were reported to evaluate performance comprehensively. To assess the randomness, I will calculate the mean and standard deviation of the test scores for each algorithm.

IV. Results

The baseline F1 score for class 0 is undefined, for class 1 is 0.219.As shown in plot 4, the model performance summary plot, the XGBoost model, which achieved the highest F1 score of 0.489 ± 0.05, significantly outperformed the class 1 baseline of 0.219. This represents 5.4 standard deviations above the baseline F1 score for class 1 and its ability in identifying the minority class. Moreover, XGBoost also outperformed other models like Random Forest (0.481) and SVM (0.480), demonstrating its ability to effectively handle the class imbalance and capture complex feature interactions. Logistics regressions have the worst predictive power. The weaker performance of logistic regression can be attributed to its reliance on linear decision boundaries.

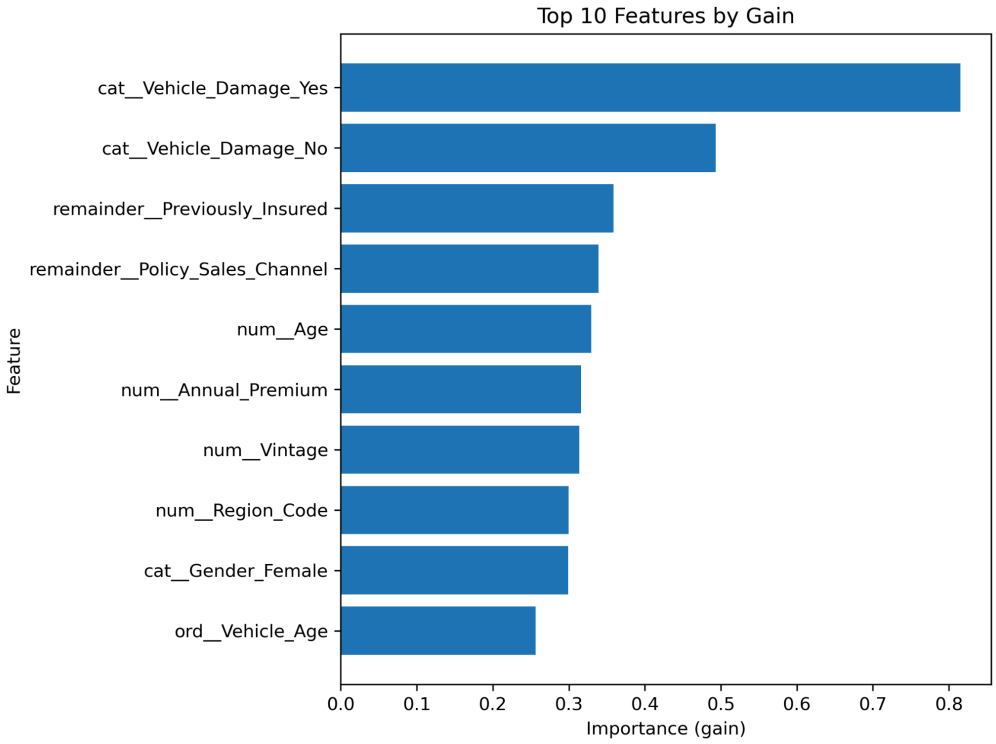


Plot 4. Bar plot of models performance summary

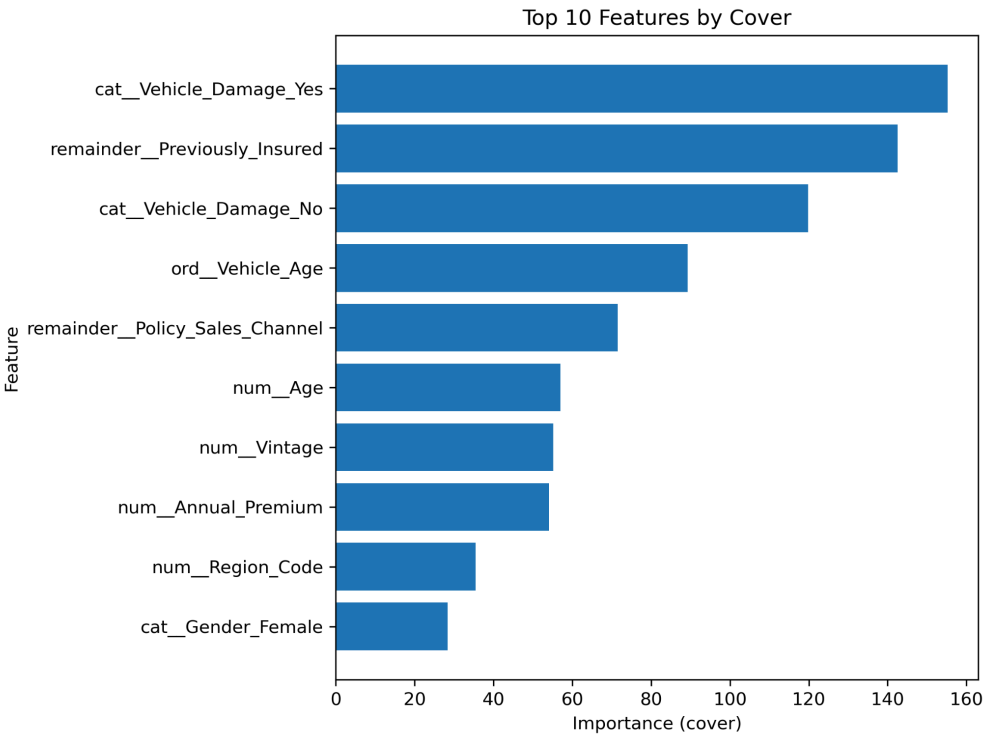


Plot 5. Confusion Matrix for the best model

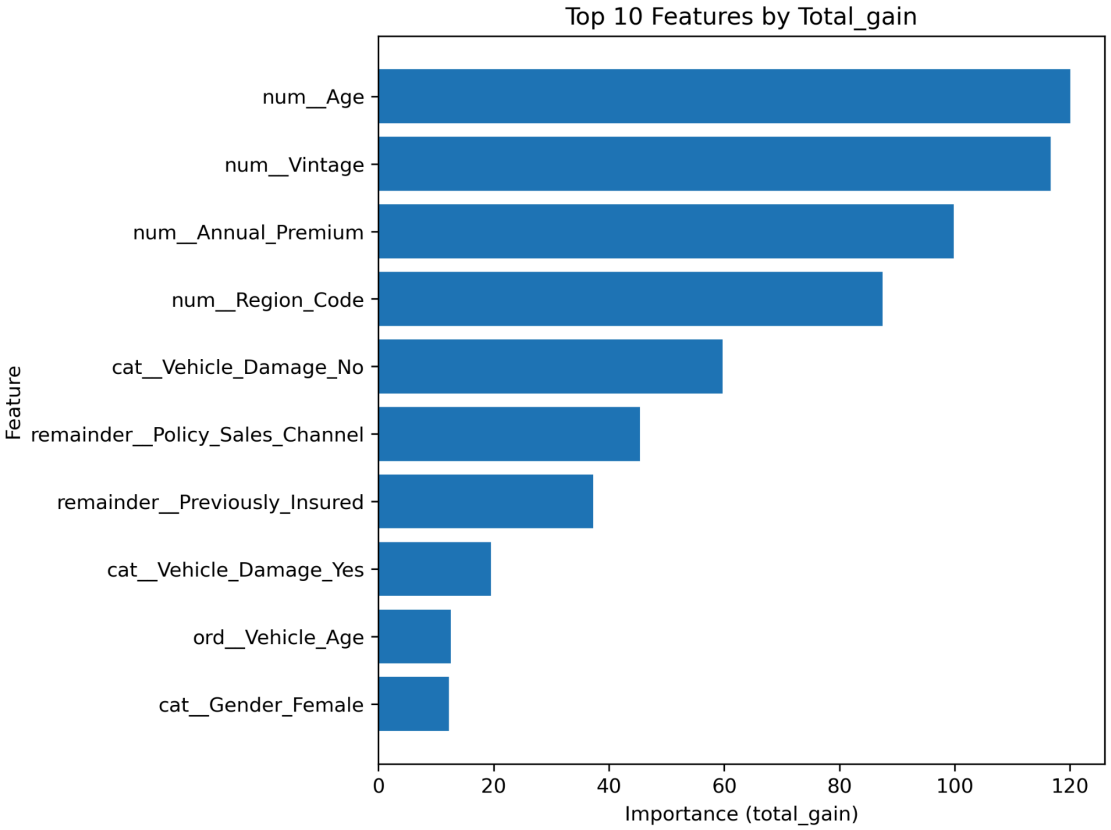
As shown in plot 6,7,8, using three metrics—cover, gain, and total gain—the analysis reveals that features such as cat__Vehicle_Damage_Yes, remainder__Previously_Insured, and cat__Vehicle_Damage_No are consistently among the most significant predictors. However, nuanced patterns emerge for features like num__Age, which ranks lower in terms of cover but higher in total gain.



Plot 6. Global feature importance by XGBoost Gain

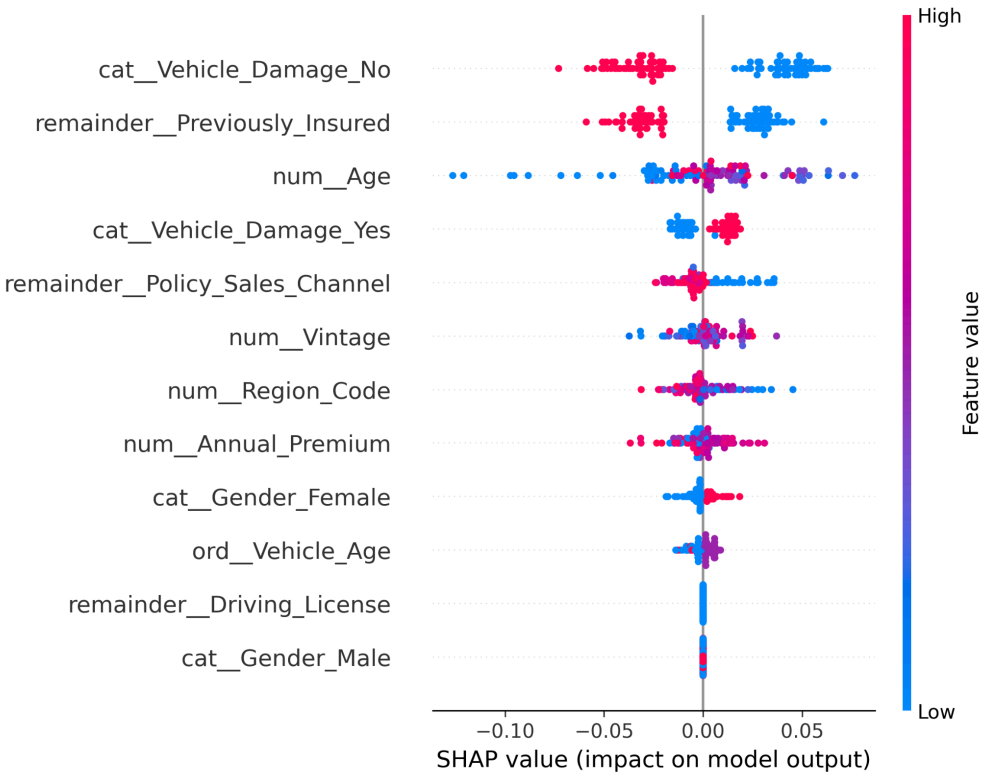


Plot 7. Global feature importance by XGBoost Cover



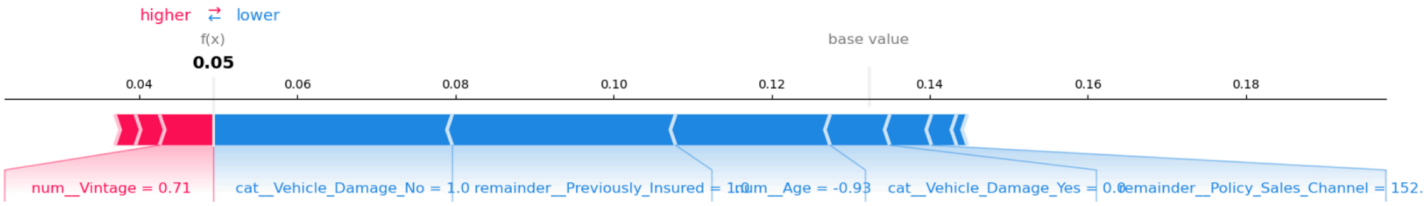
Plot 8. Global feature importance by XGBoost Total_Gain

Local feature importance, derived from SHAP values, complements the global perspective by explaining individual predictions. The SHAP summary plot shows that cat__Vehicle_Damage_No, remainder__Previously_Insured, and num__Age are the most impactful features across instances. Interestingly, num__Age exhibits both positive and negative SHAP values, suggesting that older customers are less likely to respond positively, while younger customers are more inclined to respond. Similarly, cat__Vehicle_Damage_No exhibits mixed SHAP values, indicating that the absence of vehicle damage interacts with other features in unexpected ways.



Plot 9. SHAP value for features with top impact on model

As shown in plot 10, The SHAP force plot shows that, for customer indice = 30, the prediction of 0.05, meaning the features collectively pull the model's output downward from the base value.. num__Vintage strongly increases the likelihood of a positive response, while cat__Vehicle_Damage_No and remainder__Previously_Insured significantly lower it.



Plot 10. SHAP force plot for specific customer.

V. Outlook

In order to handle the imbalance data, I already did the stratified splitting strategy and set the class weight to be "balanced," but there are still a lot of potential aspects that I can work on, and feature engineering is the part which hopefully I can dive deeper into in the future. In my current project, I did not do any feature engineering here. The weak correlations in the heatmap suggest that non-linear models like Random Forest or Gradient Boosting may perform better by identifying complex, non-linear patterns that logistic regression—being the simplest model used in this analysis—might fail to detect. Therefore, feature engineering and exploring advanced models are critical next steps for improving predictive performance. Another critical consideration is threshold optimization. Logistic regression and other probabilistic models produce probabilities rather than binary predictions by default, meaning that the classification threshold (commonly set at 0.5) can be adjusted to optimize performance.

VI. Reference

[1] https://www.kaggle.com/datasets/anmolkumar/health-insurance-cross-sell-prediction/data

[2]https://www.tandfonline.com/doi/pdf/10.1080/08874417.2024.2395913?utm_source=chatgpt.com

[3]https://github.com/Manojpatil123/Capstone-project-Supervised_machinelearning_classification_on_HEALTH-INSURANCE-CROSS-SELL-PREDICTION