Data 621 HW4-Auto Insurance Claim Prediction

(Group 4) Yina Qiao, Mohamed Hassan-El Seraf, Chun Shing Leung, Keith Colella, Eddie Xu

Introduction

In the auto insurance industry, accurately predicting the likelihood of a crash and estimating potential claim amounts are critical for effective risk management and pricing strategies. Insurers use historical data and predictive models to assess individual risk profiles, allowing them to set premiums that reflect each customer's likelihood of filing a claim. This project focuses on building robust models to (1) predict the probability of a crash and (2) estimate the expected claim amount, leveraging a range of factors such as demographics, vehicle characteristics, and driving history.

To achieve these goals, we develop and evaluate multiple models for both tasks: Binary Logistic Regression (BLR) models for crash probability prediction and Multiple Linear Regression (MLR) models for claim amount estimation. By addressing key challenges such as data skewness, class imbalance, and multicollinearity, we aim to create models that offer accurate, interpretable, and production-ready predictions. The outcome of this project provides a foundational approach for insurance companies to assess risk, improve premium accuracy, and enhance customer satisfaction.

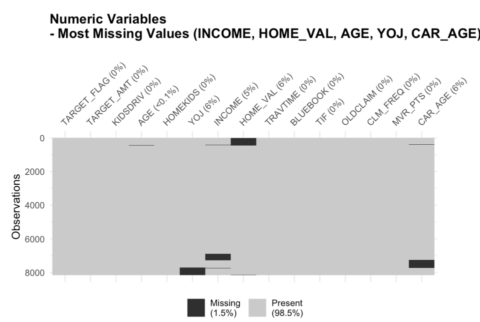
Data Exploration

In this section, we explore the dataset to understand its structure, variables, and summary statistics. We use specialized plots (from the visdat package) to identify missing values in both numeric and categorical variables, visualizations such as box plots, histograms, and bar plots to reveal distributions, a heatmap to show correlations. The goal is to provide a clear overview of the data, highlighting key findings that will guide the next steps in data transformation

Dataset

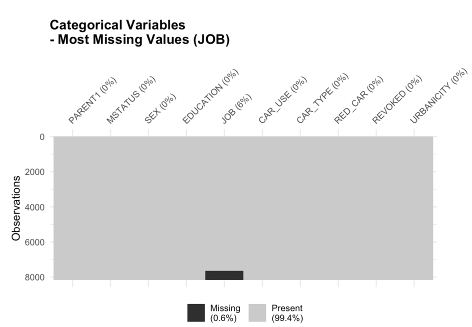
The dataset comprises 8,161 records across 25 columns, indicating a likely high-dimensional structure with a mix of numeric and categorical data types. The TARGET_FLAG variable indicates accident cases and shows a class imbalance, with accident cases comprising about 26% of the data. Key numeric fields, including INCOME, HOME_VAL, OLDCLAIM, and BLUEBOOK, contain numeric data represented as characters (e.g., $, z_, and <) and require cleaning. Additionally, there are many missing values in 6 fields, including INCOME, HOME_VAL, and JOB etc, totally 1879 numeric value missing and 526 categorical values missing. Many variables exhibit right skewness, particularly TARGET_AMT, suggesting outliers. Variables such as MVR_PTS and CLM_FREQ show moderate positive correlations with accident likelihood and claim amounts, while HOME_VAL and INCOME correlate slightly negatively with accident likelihood.

```
## 'data.frame':    8161 obs. of  25 variables:
##  $ TARGET_FLAG: int  0 0 0 0 0 1 0 1 1 0 ...
##  $ TARGET_AMT : num  0 0 0 0 0 ...
##  $ KIDSDRIV   : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ AGE        : int  60 43 35 51 50 34 54 37 34 50 ...
##  $ HOMEKIDS   : int  0 0 1 0 0 1 0 2 0 0 ...
##  $ YOJ        : int  11 11 10 14 NA 12 NA NA 10 7 ...
##  $ INCOME     : chr  "$67,349" "$91,449" "$16,039" "" ...
##  $ PARENT1    : chr  "No" "No" "No" "No" ...
##  $ HOME_VAL   : chr  "$0" "$257,252" "$124,191" "$306,251" ...
##  $ MSTATUS    : chr  "z_No" "z_No" "Yes" "Yes" ...
##  $ SEX        : chr  "M" "M" "z_F" "M" ...
##  $ EDUCATION  : chr  "PhD" "z_High School" "z_High School" "<High School" ...
##  $ JOB        : chr  "Professional" "z_Blue Collar" "Clerical" "z_Blue Collar" ...
##  $ TRAVTIME   : int  14 22 5 32 36 46 33 44 34 48 ...
##  $ CAR_USE    : chr  "Private" "Commercial" "Private" "Private" ...
##  $ BLUEBOOK   : chr  "$14,230" "$14,940" "$4,010" "$15,440" ...
##  $ TIF        : int  11 1 4 7 1 1 1 1 1 7 ...
##  $ CAR_TYPE   : chr  "Minivan" "Minivan" "z_SUV" "Minivan" ...
##  $ RED_CAR    : chr  "yes" "yes" "no" "yes" ...
##  $ OLDCLAIM   : chr  "$4,461" "$0" "$38,690" "$0" ...
##  $ CLM_FREQ   : int  2 0 2 0 2 0 0 1 0 0 ...
##  $ REVOKED    : chr  "No" "No" "No" "No" ...
##  $ MVR_PTS    : int  3 0 3 0 3 0 0 10 0 1 ...
##  $ CAR_AGE    : int  18 1 10 6 17 7 1 7 1 17 ...
##  $ URBANICITY : chr  "Highly Urban/ Urban" "Highly Urban/ Urban" "Highly Urban/ Urban" "Highly Urban/ Urban"
...
```



The count of missing records

```
##      AGE      YOJ   INCOME HOME_VAL  CAR_AGE
##        6      454      445      464      510
```



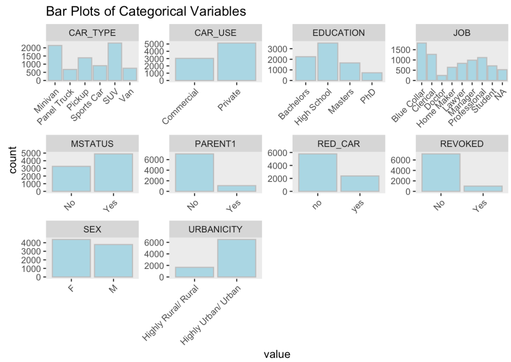
The count of missing records

```
## JOB
## 526
```

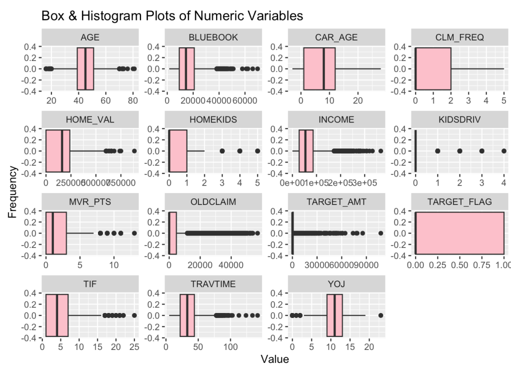
```
##   rows columns discrete_columns continuous_columns all_missing_columns
## 1 8161      25               10                 15                   0
##   total_missing_values complete_rows total_observations memory_usage
## 1                 1879          6448             204025      1313528
```

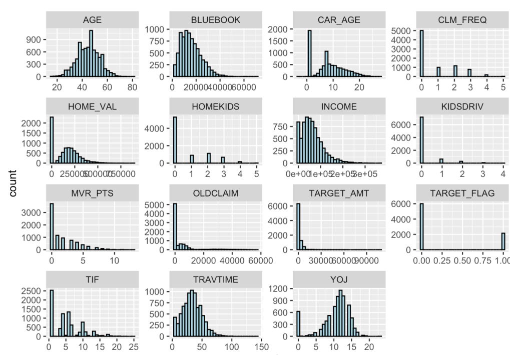
Metrics

The bar plots for categorical variables highlight clear distributions across categories. Some categories like JOB and EDUCATION have diverse entries.

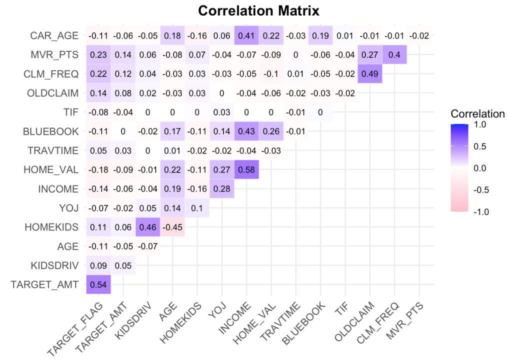


Below plots along with summary statistics shows that most numeric variables like TARGET_AMT, MVR_PTS and TRAVTIME,etc exhibit right-skewed distributions and outliers. Summary statistics also highlight that only 26% of cases involve crashes, indicating class imbalance. Both suggest potential data transformation needed for modeling.






```
##              vars      mean        sd median  min      max  skew kurtosis
## TARGET_FLAG     1      0.26      0.44      0    0      1.0  1.07    -0.85
## TARGET_AMT      2   1504.32   4704.03      0    0 107586.1  8.71   112.29
## KIDSDRIV        3      0.17      0.51      0    0      4.0  3.35    11.78
## AGE             4     44.79      8.63     45   16     81.0 -0.03    -0.06
## HOMEKIDS        5      0.72      1.12      0    0      5.0  1.34     0.65
## YOJ             6     10.50      4.09     11    0     23.0 -1.20     1.18
## INCOME          7  61898.09  47572.68  54028    0 367030.0  1.19     2.13
## PARENT1*        8      1.13      0.34      1    1      2.0  2.17     2.73
## HOME_VAL        9 154867.29 129123.77 161160    0 885282.0  0.49    -0.02
## MSTATUS*       10      1.60      0.49      2    1      2.0 -0.41    -1.83
## SEX*           11      1.46      0.50      1    1      2.0  0.14    -1.98
## EDUCATION*     12      2.11      0.91      2    1      4.0  0.50    -0.51
## JOB*           13      4.83      2.62      5    1      9.0  0.13    -1.46
## TRAVTIME       14     33.49     15.91     33    5    142.0  0.45     0.66
## CAR_USE*       15      1.63      0.48      2    1      2.0 -0.53    -1.72
## BLUEBOOK       16  15709.90   8419.73  14440 1500  69740.0  0.79     0.79
## TIF            17      5.35      4.15      4    1     25.0  0.89     0.42
## CAR_TYPE*      18      3.34      1.76      3    1      6.0 -0.10    -1.43
## RED_CAR*       19      1.29      0.45      1    1      2.0  0.92    -1.16
## OLDCLAIM       20   4037.08   8777.14      0    0  57037.0  3.12     9.86
## CLM_FREQ       21      0.80      1.16      0    0      5.0  1.21     0.28
## REVOKED*       22      1.12      0.33      1    1      2.0  2.30     3.30
## MVR_PTS        23      1.70      2.15      1    0     13.0  1.35     1.38
## CAR_AGE        24      8.33      5.70      8   -3     28.0  0.28    -0.75
## URBANICITY*    25      1.80      0.40      2    1      2.0 -1.46     0.15
```



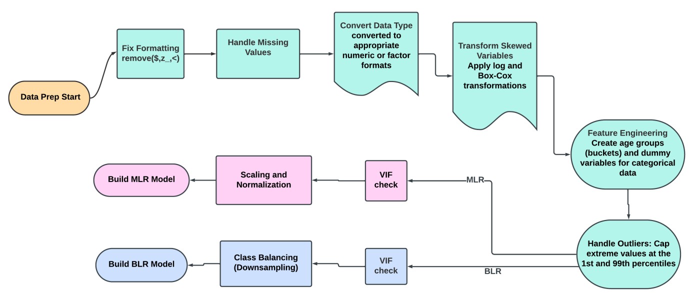
A few key variable relationships observed from the correlation plot are summarized below:

Variable Pairs	Correlation	Insights
TARGET_FLAG vs. TARGET_AMT	0.54	Higher accident likelihood corresponds to increased claim amounts.
TARGET_FLAG vs. MVR_PTS,CLM_FREQ, HOME_VAL, INCOME	0.23, 0.22, -0.18, -0.14	Higher accident likelihood is moderately linked to more traffic violations and claim frequency, while HOME_VAL and INCOME have a slight negative effect.
TARGET_AMT vs. MVR_PTS, CLM_FREQ	0.14, 0.12	More traffic violations and claim frequency are slightly associated with higher claim amounts.
INCOME vs. HOME_VAL, BLUEBOOK	0.58, 0.43	Strong positive correlation suggests that higher income often aligns with higher home value and more expensive cars.
KIDSDRIV vs. HOMEKIDS	0.46	More children in the household are linked to having more young drivers at home.

Next steps will focus on variable cleaning/transformation, imputation for missing values, handling skewness, outliers, vif check, class imbalance to build models.

Data Prep

This flowchart provides an overview of the comprehensive data preparation process applied to ensure consistency and accuracy throughout the dataset. Given the high dimensionality, numerous missing values, and a mix of numeric and categorical variables, a unified approach was crucial to maintain data integrity and reliability for modeling. This workflow highlights the extensive effort put into transforming the raw data into a form suitable for building robust and accurate models.

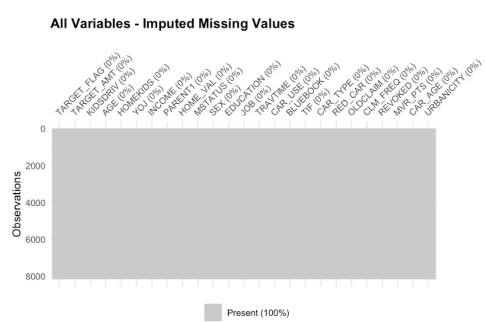


Data Prep Process Flow

By following this structured process, each model (MLR and BLR) is built upon a well-prepared dataset, allowing for consistent preprocessing, improved model interpretability, and minimized biases. Below are key data preparation steps with visual summaries

Missing Values

This plot shows the distribution of missing values across variables, where all missing values were treated to ensure a complete dataset for model reliability. Numeric variables were imputed using regression-based methods, which preserve relationships between variables, enhancing model accuracy. Categorical variables were imputed using mode-based imputation to retain the most common category, ensuring consistency. Treating missing data minimizes bias and improves the model's predictive performance by leveraging all available information.

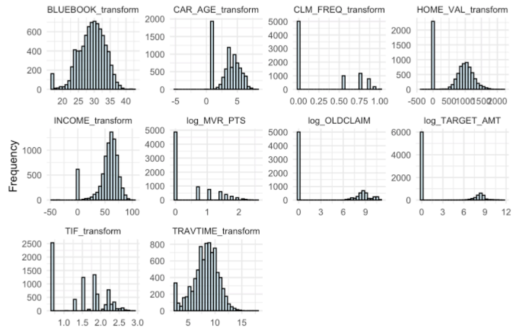


Missing Value Check Plot

Log/Box-Cox Transformation

This plot displays the distribution of transformed variables, where log and Box-Cox transformations were applied to correct right skewness. Log transformations were applied to highly skewed variables (TARGET_AMT, OLDCLAIM, MVR_PTS), and Box-Cox transformations were applied to moderately skewed variables (BLUEBOOK, CAR_AGE, HOME_VAL, INCOME, TIF, TRAVTIME, CLM_FREQ).

Right-skewed distributions can distort model predictions, especially in regression models, where extreme values can unduly influence outcomes. Compared to the original distributions in the data exploration section, these transformations have significantly normalized the distributions, making them closer to normal and thereby enhancing the stability and accuracy of the model.

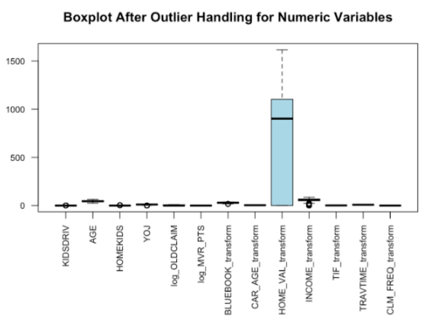


Log and Box Transformations

Treating Outliers

This boxplot shows the effect of outlier handling on numeric variables, where extreme values were capped at the 1st and 99th percentiles. Treating outliers before model building reduces the skew from extreme values, enhancing model reliability by preventing overfitting to unusual data points. Most variables now fall within a reasonable range, making the data well-suited for modeling.

The HOME_VAL_transform variable, however, stands out with a higher range, reflecting the natural variability in property values. While outliers have been capped, property values inherently span a wider range than other variables. This spread is expected and acceptable, as the primary outliers have been controlled, ensuring that HOME_VAL_transform remains within a manageable range for modeling.



Outlier Handling

VIF Check

This set of VIF (Variance Inflation Factor) checks for both the MLR (Multiple Linear Regression) and BLR (Binary Logistic Regression) models helps identify multicollinearity—situations where variables are highly correlated with each other. High multicollinearity can distort model estimates and reduce interpretability, so addressing it is essential for achieving accurate and reliable model results.

Criteria: Variables with VIF scores above 5 were flagged for removal, as values above this threshold typically suggest problematic multicollinearity.

For MLR: The log_CLM_FREQ variable was dropped because it was less relevant for predicting claim amounts. In contrast, log_OLDCLAIM was retained, as it directly reflects past claim history, providing a more meaningful predictor of future claim amounts.

For BLR: The log_OLDCLAIM variable was excluded, while log_CLM_FREQ was retained, as claim frequency is more indicative of future claims; individuals with frequent past claims are more likely to file again.

As shown in the “Before” and “After” VIF plots for both models, high-VIF variables were successfully removed, resulting in VIF scores below the threshold. This confirms that the final variable sets are well-suited for modeling, with minimized multicollinearity, ensuring each variable contributes uniquely and meaningfully to the model.

KIDSDRIV	1.378668	AGE	3.941766
HOMEKIDS	2.163478	YOJ	1.873150
log_OLDCLAIM	14.212348	log_MVR_PTS	1.292822
BLUEBOOK_transform	1.780145	CAR_AGE_transform	1.872712
HOME_VAL_transform	2.066119	INCOME_transform	4.185036
TIF_transform	1.006331	TRAVTIME_transform	1.032125
CLM_FREQ_transform	13.895263	PARENT1_Yes	1.858020
MSTATUS_Yes	2.136203	SEX_M	3.203730
`EDUCATION_High School`	2.307614	EDUCATION_Masters	2.326119
EDUCATION_PhD	2.008367	JOB_Clerical	1.849708
JOB_Doctor	1.734730	`JOB_Home Maker`	2.308049
JOB_Lawyer	2.351996	JOB_Manager	1.688606
JOB_Professional	1.735397	JOB_Student	2.265551
CAR_USE_Private	2.259492	`CAR_TYPE_Panel Truck`	1.969397
CAR_TYPE_Pickup	1.584292	`CAR_TYPE_Sports Car`	1.841540
CAR_TYPE_SUV	2.469684	CAR_TYPE_Van	1.457785
RED_CAR_yes	1.815937	REVOKED_Yes	1.076566
`URBANICITY_Highly Urban/ Urban`	1.256642	AGE_GROUP_Older	2.776847
AGE_GROUP_Young	1.407656		

MLR VIF Before

KIDSDRIV	1.377925	AGE	3.941710
HOMEKIDS	2.163422	YOJ	1.872409
log_OLDCLAIM	1.395306	log_MVR_PTS	1.292466
BLUEBOOK_transform	1.779625	CAR_AGE_transform	1.872662
HOME_VAL_transform	2.065960	INCOME_transform	4.184896
TIF_transform	1.006317	TRAVTIME_transform	1.031215
PARENT1_Yes	1.858020	MSTATUS_Yes	2.136044
SEX_M	3.203470	`EDUCATION_High School`	2.307505
EDUCATION_Masters	2.326106	EDUCATION_PhD	2.008312
JOB_Clerical	1.849452	JOB_Doctor	1.734470
`JOB_Home Maker`	2.307950	JOB_Lawyer	2.351737
JOB_Manager	1.688095	JOB_Professional	1.735381
JOB_Student	2.265521	CAR_USE_Private	2.258728
`CAR_TYPE_Panel Truck`	1.969279	CAR_TYPE_Pickup	1.584156
`CAR_TYPE_Sports Car`	1.841239	CAR_TYPE_SUV	2.469146
CAR_TYPE_Van	1.457739	RED_CAR_yes	1.815665
REVOKED_Yes	1.028698	`URBANICITY_Highly Urban/ Urban`	1.253768
AGE_GROUP_Older	2.776340	AGE_GROUP_Young	1.407656

MLR VIF After

```
                 KIDSDRIV                        AGE
                 1.425083                   4.111548
                 HOMEKIDS                        YOJ
                 2.253667                   1.989959
             log_OLDCLAIM                log_MVR_PTS
                11.685569                   1.237436
       BLUEBOOK_transform          CAR_AGE_transform
                 1.835622                   1.894077
       HOME_VAL_transform           INCOME_transform
                 1.928412                   4.340426
            TIF_transform         TRAVTIME_transform
                 1.010776                   1.032632
       CLM_FREQ_transform                PARENT1_Yes
                11.459528                   1.934339
              MSTATUS_Yes                      SEX_M
                 2.221207                   3.513644
 `EDUCATION_High School`           EDUCATION_Masters
                 2.296231                   2.317717
            EDUCATION_PhD               JOB_Clerical
                 1.817456                   1.776358
               JOB_Doctor             `JOB_Home Maker`
                 1.489774                   2.355835
               JOB_Lawyer                JOB_Manager
                 2.186764                   1.393763
         JOB_Professional                JOB_Student
                 1.600690                   2.377885
          CAR_USE_Private    `CAR_TYPE_Panel Truck`
                 2.217844                   2.142679
          CAR_TYPE_Pickup     `CAR_TYPE_Sports Car`
                 1.796621                   2.154584
             CAR_TYPE_SUV               CAR_TYPE_Van
                 2.932501                   1.637792
              RED_CAR_yes                REVOKED_Yes
                 1.832314                   1.069666
URBANICITY_Highly Urban/ Urban`          AGE_GROUP_Older
                 1.143935                   2.829658
          AGE_GROUP_Young
                 1.474601
```

BLR VIF Before

```
                 KIDSDRIV                               AGE
                 1.424653                          4.111485
                 HOMEKIDS                               YOJ
                 2.253672                          1.988912
              log_MVR_PTS                BLUEBOOK_transform
                 1.217812                          1.835024
        CAR_AGE_transform                HOME_VAL_transform
                 1.893890                          1.928311
         INCOME_transform                     TIF_transform
                 4.338801                          1.010777
       TRAVTIME_transform                CLM_FREQ_transform
                 1.032333                          1.236405
              PARENT1_Yes                       MSTATUS_Yes
                 1.934280                          2.221650
                    SEX_M           `EDUCATION_High School`
                 3.514257                          2.296124
        EDUCATION_Masters                     EDUCATION_PhD
                 2.317425                          1.817830
             JOB_Clerical                        JOB_Doctor
                 1.775154                          1.489564
         `JOB_Home Maker`                        JOB_Lawyer
                 2.354775                          2.186343
              JOB_Manager                  JOB_Professional
                 1.393130                          1.600584
              JOB_Student                   CAR_USE_Private
                 2.377554                          2.217962
   `CAR_TYPE_Panel Truck`                   CAR_TYPE_Pickup
                 2.142865                          1.796503
    `CAR_TYPE_Sports Car`                      CAR_TYPE_SUV
                 2.154681                          2.932890
             CAR_TYPE_Van                       RED_CAR_yes
                 1.637719                          1.831986
              REVOKED_Yes `URBANICITY_Highly Urban/ Urban`
                 1.006725                          1.143189
          AGE_GROUP_Older                   AGE_GROUP_Young
                 2.829077                          1.474466
```

BLR VIF After

Class Imbalance

This table illustrates the impact of addressing class imbalance for the BLR (Binary Logistic Regression) model. Initially, Class 0 had nearly three times as many observations as Class 1, which could cause the model to be biased toward Class 0. To improve performance, especially in predicting the minority class (Class 1), downsampling was applied, balancing both classes at 1,723 observations each. Balancing classes is like giving equal practice time to both teams in a game, helping the model "learn" both sides fairly and reducing bias towards the majority class. This balanced dataset enhances the model's ability to detect patterns for both classes effectively.

	Class 0	Class 1
Original Count	4807	1723
Downsampled Count	1723	1723

Class Imbalance in BLR (Before and After)

Build Model

Model Building for Multiple Linear Regression (MLR) and Binary Logistic Regression (BLR)

With the training data explored and preprocessed, 2 MLR and 3 BLR models were developed to determine the best-performing models for each task.

Multiple Linear Regression (MLR) Models

Since `TARGET_AMT` is highly right-skewed (skewness = 8.71) with notable outliers, a log transformation was applied to normalize its distribution. Both MLR models were trained on `log_TARGET_AMT`, which stabilized the model and reduced the influence of extreme values.

- **Full Model**: This model included all relevant predictors, providing a comprehensive view of the factors affecting the target amount and serving as a baseline for comparison.
- **Stepwise Model**: A stepwise selection process was applied, iteratively adding or removing variables based on significance. This method identified a more efficient model with fewer predictors, enhancing both interpretability and prediction accuracy.

Binary Logistic Regression (BLR) Models

To address class imbalance, each BLR model was trained on a balanced dataset achieved through downsampling, allowing for unbiased learning between classes.

- **Null Model**: A baseline model with only an intercept, serving as a benchmark for assessing improvements in models with predictors.
- **Full Model**: This model included all predictor variables related to `TARGET_FLAG`, helping identify comprehensive factors linked to claim likelihood.
- **Stepwise Model**: This model used an alternative approach to traditional stepwise selection by incorporating additional preprocessing. After downsampling, near-zero variance predictors and highly correlated features were removed, achieving a refined predictor set without the typical AIC process. This approach reduced redundancy and multicollinearity, enhancing interpretability and reducing overfitting, much like clearing clutter to focus on key information.

This model-building strategy effectively handled skewness in `TARGET_AMT` for the MLR models and optimized predictor selection in the BLR models.

Select Model

Based on the metrics and plot below the selection of the best models for predicting `TARGET_AMT` (MLR) and `TARGET_FLAG` (BLR) is as follows:

Multiple Linear Regression (MLR) - Targeting `TARGET_AMT`

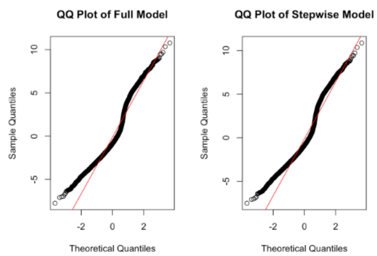
- **Metrics Comparison**: The MSE, RMSE, and R-squared values are closely matched between the Full and Stepwise models, indicating similar predictive accuracy. However, the Stepwise model shows a slightly higher Adjusted R-squared (0.224), suggesting it balances prediction accuracy and model simplicity more effectively.

- **Residual Analysis**: The QQ plots for both the Full and Stepwise models show residuals that closely follow the line of normality, indicating both models handle residuals similarly well. However, the Stepwise model, with fewer predictors (25 vs. 36), achieves this efficiency with less complexity.

Selection: The Stepwise MLR model is chosen for its comparable accuracy with fewer predictors, improving interpretability while maintaining strong prediction performance for `TARGET_AMT`.

Metric	Full Model	Stepwise Model
MSE	10.41	10.40
RMSE	3.23	3.23
R-squared	0.228	0.229
Adjusted R-squared	0.221	**0.224**
F-statistic	33.20	48.06
Residual Std. Error (Training)	3.24	3.24
Number of Predictors	36	25

MLR Model Metrics



MLR Residual Plot

Binary Logistic Regression (BLR) - Targeting `TARGET_FLAG`

- **Model Metrics**: For the BLR models, the Stepwise model achieves the highest AUC (0.814), indicating the strongest ability to distinguish between crash and no-crash cases. It also has the highest Kappa (0.390, Kappa is analogous to a grading system that accounts for both correct and lucky guesses), showing better overall agreement in classification performance beyond chance.

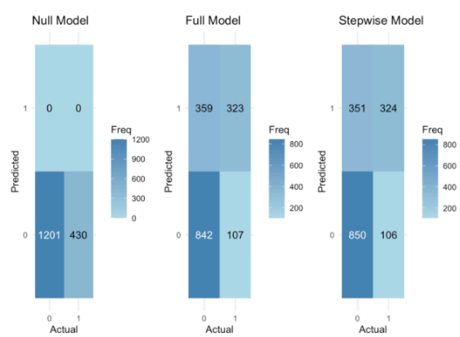
- **Confusion Matrix**: The confusion matrix for the Stepwise model reveals balanced sensitivity and specificity, with improved performance over the Null model, which lacks predictors, and the Full model, which includes all predictors. This balance suggests the Stepwise model generalizes well to both classes without overfitting to one.

Selection: The Stepwise BLR model is selected due to its superior AUC and Kappa scores, indicating better classification performance with an efficient subset of predictors, enhancing model interpretability and effectiveness in predicting `TARGET_FLAG`.

Overall, these selected models (Stepwise MLR and Stepwise BLR) provide robust and interpretable solutions, balancing accuracy and complexity for predicting both claim amount and crash likelihood.

Metric	Null Model	Full Model	Stepwise Model
Accuracy	0.736	0.714	0.720
Error Rate	0.264	0.286	0.280
Kappa	0.000	0.381	**0.390**
Precision	N/A	0.474	0.480
Sensitivity	0.000	0.751	0.753
Specificity	1.000	0.701	0.708
F1 Score	N/A	0.581	0.586
AUC	0.500	0.813	**0.814**

BLR Model Metrics



Confusion Matrix Grid

Predictions on Test Data

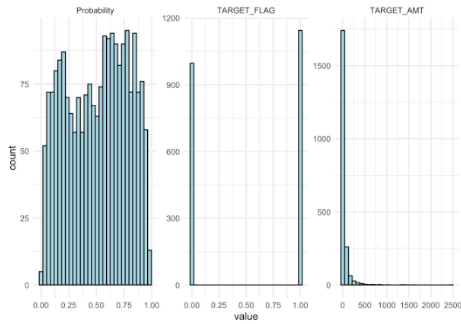
The selected Stepwise MLR and Stepwise BLR models were retrained on the entire labeled dataset to enhance robustness and accuracy for final predictions on the test data. The test data, which lacks labels for `TARGET_FLAG` and `TARGET_AMT`, underwent the same preprocessing steps used in model training. Predictions were then generated using the retrained models.

The plot below displays the predictions for the probability of a crash (`TARGET_FLAG`) and the expected claim amount (`TARGET_AMT`).

- **Probability of Crash (`TARGET_FLAG`)**: The left panel shows the distribution of predicted probabilities for a crash. Most probabilities are spread across the entire range, with a notable concentration near 1, indicating a strong confidence in predicting crash occurrences for some cases.

- **Predicted Claim Amount (`TARGET_AMT`)**: The right panel shows the distribution of predicted claim amounts. The distribution is highly skewed to the right, with most values near zero, indicating that the model generally predicts lower claim amounts, which is common when using a log-transformed model. While log models often produce slightly lower predictions, this approach prioritizes prediction accuracy, particularly for more typical, smaller claims. For extreme cases, adjustments can be made in post-processing if necessary.

In summary, the selected models provide a reliable approach for identifying likely crash cases and estimating associated claim amounts on new data. The distribution of predictions aligns well with expectations, capturing both the probability of a crash and the typically smaller but occasionally high claim amounts.



Predictions on Test Data

Appendix

```
---
title: "Data 621 HW4-Auto Insurance Claim Prediction"
author: (Group 4) Yina Qiao, Mohamed Hassan-El Seraf, Chun Shing Leung, Keith Colella, Eddie Xu
output:
  html_document:
    toc: true
    toc_depth: 2
---

This project aims to predict the probability that a person will crash their car and estimate the potential claim
amount based on various factors.

# DATA EXPLORATION
In this section, we explore the dataset to understand its structure, variables, and summary statistics. We us esp
ecialized plots (from the visdat package) to identify missing values in both numeric and categorical variables, v
isualizations such as box plots, histograms, and bar plots to reveal distributions, a heatmap to show correlation
s.  The goal is to provide a clear overview of the data, highlighting key findings that will guide the next steps
in data transformation

# ---
# Exploratory Data Analysis
library(tidyverse)
library(dplyr)
library(DataExplorer)
library(visdat)
library(ggplot2)
library(psych)
library(reshape2)
# Handling Missing Values
library(mice)

# Creating Dummy Variables
library(caret)
library(fastDummies)
# Multicollinearity Check
library(car)
# Scaling & Normalization
library(scales)
# Class Balancing
library(tidymodels)
library(themis)

library(leaps)
library(caret)
library(MASS)
library(forecast)
library(yardstick)
library(pROC)
library(gridExtra)



df_ins_raw <- read.csv("https://raw.githubusercontent.com/yinaS1234/Auto-Insurance-Regression/refs/heads/main/dat
a/insurance_training_data.csv?token=GHSAT0AAAAAACYZCXU2KALCPQONYEWJER42ZZL2RWA")
df_ins_raw <- subset(df_ins_raw, select = -c(INDEX))

# ---

## Dataset

The dataset comprises 8,161 records across 25 columns, indicating a likely high-dimensional structure with a mix
of numeric and categorical data types. The TARGET_FLAG variable indicates accident cases and shows a class imbala
nce, with accident cases comprising about 26% of the data.  Key numeric fields, including INCOME, HOME_VAL, OLDCL
AIM, and BLUEBOOK, contain numeric data represented as characters (e.g., $, z_, and <) and require cleaning.  Add
itionally, there are missing values in 6 fields, including INCOME, HOME_VAL, and JOB etc, affecting approximately
1.5% of observations. Many variables exhibit right skewness, particularly TARGET_AMT, suggesting outliers. Variab
les such as MVR_PTS and CLM_FREQ show moderate positive correlations with accident likelihood and claim amounts,
while HOME_VAL and INCOME correlate slightly negatively with accident likelihood.

# ---
str(df_ins_raw)
# ---



# ---
df_ins_raw <- df_ins_raw %>%
  mutate(
    INCOME = as.numeric(gsub("[$,]", "", INCOME)),
    HOME_VAL = as.numeric(gsub("[$,]", "", HOME_VAL)),
    OLDCLAIM = as.numeric(gsub("[$,]", "", OLDCLAIM)),
    BLUEBOOK = as.numeric(gsub("[$,]", "", BLUEBOOK))
  )

df_ins_raw <- df_ins_raw %>%
  mutate(across(where(~ !is.numeric(.)),
                ~ str_replace_all(., c("z_" = "", "<" = ""))))
# ---



# ---
num_vars <- df_ins_raw %>% select_if(where(is.numeric))
vis_miss(num_vars, cluster = TRUE) +
  ggtitle("Numeric Variables \n- Most Missing Values (INCOME, HOME_VAL, AGE, YOJ, CAR_AGE)") +
  theme(
    plot.title = element_text(face = "bold"),
    plot.margin = unit(c(1, 2, 1, 1), "cm")
  )
# ---

The count of missing records
# ---
print(colSums(is.na(num_vars))[colSums(is.na(num_vars)) > 0])
# ---

# ---
cat_vars <- df_ins_raw %>% select_if(~ !is.numeric(.))
cat_vars <- cat_vars %>%
  mutate(across(everything(), ~na_if(., "")))
vis_miss(cat_vars, cluster = TRUE) +
  ggtitle("Categorical Variables \n- Most Missing Values (JOB)") +
  theme(
    plot.title = element_text(face = "bold"),
    plot.margin = unit(c(1, 2, 1, 1), "cm")
  )
# ---

The count of missing records
# ---
print(colSums(is.na(cat_vars))[colSums(is.na(cat_vars)) > 0])
# ---


# ---
introduce(df_ins_raw )

# ---

Metrics





The bar plots for categorical variables highlight clear distributions across categories.  Some categories like JO
B and EDUCATION have diverse entries.

# ---


cat_vars%>%
  gather() %>%
  ggplot(aes(value)) +
  geom_bar(fill = "lightblue", color="grey") +
  facet_wrap(~ key, scales = "free", ncol = 4) +
  theme(
    panel.grid = element_blank(),
    axis.text.x = element_text(angle = 45, hjust = 1)
```

```
  ) +
  labs(title = "Bar Plots of Categorical Variables")

# ---
```

Below plots along with summary statistics shows that most numeric variables like TARGET_AMT, MVR_PTS and TRAVTIM E,etc exhibit right-skewed distributions and outliers. Summary statistics also highlight that only 26% of cases involve crashes, indicating class imbalance. Both suggest potential data transformation needed **for** modeling.

```
# ---


# Boxplot
num_vars %>%
  gather() %>%
  ggplot(aes(value)) +
  facet_wrap(~ key, scales = "free") +
  geom_boxplot(fill = "pink") +
  labs(title = "Box & Histogram Plots of Numeric Variables", x = "Value", y = "Frequency")



ggplot(gather(num_vars), aes(value)) +
  facet_wrap(~ key, scales = "free") +
  geom_histogram(bins = 30, fill = "lightblue", color = "black")
# ---




# ---
full_summary <- describeBy(df_ins_raw)
print(full_summary[,c(1, 3, 4, 5, 8, 9, 11, 12)])
# ---



# ---
cor_matrix <- cor(df_ins_raw %>% select_if(where(is.numeric)), use = "complete.obs")

cor_long <- melt(cor_matrix)
cor_long <- cor_long[as.numeric(cor_long$Var1) > as.numeric(cor_long$Var2), ]


ggplot(cor_long, aes(Var2, Var1, fill = value)) +
  geom_tile(color = "white") +
  geom_text(aes(label = ifelse(value != 0, round(value, 2), "")),
            color = "black", size = 3, face="bold") +  # Show only significant values
  scale_fill_gradient2(low = "pink", high = "blue", mid = "white",
                       midpoint = 0, limit = c(-1, 1),
                       space = "Lab", name = "Correlation") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size = 10),  # Adjust x-axis label
    axis.text.y = element_text(size = 10),                                   # Adjust y-axis label
    axis.title = element_blank(),                                            # Remove axis titles
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14)          # Center plot title
  ) +
  ggtitle("Correlation Matrix")

# ---
```

A few key variable relationships observed from the correlation plot are summarized below:

`![Insight Table](https://github.com/yinaS1234/Auto-Insurance-Regression/raw/main/Resources/insighttable.png)`

Next steps will focus on variable cleaning/transformation, imputation **for** missing values, handling skewness, outliers, vif check, class imbalance to build reliable models.

```
# Data Prep
# ---
df_ins_prep<-df_ins_raw
str(df_ins_prep)
describeBy(df_ins_prep)
# ---


# ---
# Create Missing Flags for Numeric Variables
missing_flags_num <- df_ins_prep %>%
  dplyr::select_if(is.numeric) %>%
  summarise(across(everything(), ~ sum(is.na(.)) > 0)) %>%
  dplyr::select(where(~ . == TRUE)) %>%
  names()

df_ins_prep <- df_ins_prep %>%
  mutate(across(all_of(missing_flags_num), ~ ifelse(is.na(.), 1, 0), .names = "{.col}_MISSING_FLAG"))

# Create Missing Flags for Categorical Variables
df_ins_prep <- df_ins_prep %>%
  mutate(across(where(~ !is.numeric(.)), ~ na_if(., "")))

missing_flags_cat <- df_ins_prep %>%
  dplyr::select_if(~ !is.numeric(.)) %>%
  summarise(across(everything(), ~ sum(is.na(.)) > 0)) %>%
  dplyr::select(where(~ . == TRUE)) %>%
  names()

df_ins_prep <- df_ins_prep %>%
  mutate(across(all_of(missing_flags_cat), ~ ifelse(is.na(.), 1, 0), .names = "{.col}_MISSING_FLAG"))

# Numeric variables: Regression-based imputation
impute_mice <- mice(df_ins_prep, method = "norm.predict", m = 1, remove.collinear = FALSE)
df_ins_prep <- complete(impute_mice)

# Categorical Variables: Mode-based Imputation
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

df_ins_prep <- df_ins_prep %>%
  mutate(across(where(~ !is.numeric(.)), ~ replace_na(., getmode(.))))

# Select all columns except missing flags for visualization
vars_without_flags <- df_ins_prep %>% dplyr::select(-contains("_MISSING_FLAG"))

# Visualize Missing Values (All Variables After Imputation)
vis_miss(vars_without_flags, cluster = TRUE) +
  ggtitle("All Variables - Imputed Missing Values") +
  theme(
    plot.title = element_text(face = "bold"),
    plot.margin = unit(c(1, 2, 1, 1), "cm")
  )

missing_flag_counts <- df_ins_prep %>%
  dplyr::select(contains("_MISSING_FLAG")) %>%
  summarise(across(everything(), sum))

# Display the count of 1s in each missing flag column
print(missing_flag_counts)

sapply(df_ins_prep, class)

# Define the list of columns to be converted to factors
factor_cols <- c("TARGET_FLAG", "PARENT1", "MSTATUS", "SEX", "EDUCATION", "JOB",
                 "CAR_USE", "CAR_TYPE", "RED_CAR", "REVOKED", "URBANICITY")
```

```
# Convert the specified columns to factors and the rest to numeric
df_ins_prep <- df_ins_prep %>%
  mutate(across(all_of(factor_cols), as.factor)) %>%
  mutate(across(!all_of(factor_cols), as.numeric))

sapply(df_ins_prep, class)
```

```
# Define variables for conditional log transformation and Box-Cox transformation
log_vars <- c("TARGET_AMT", "OLDCLAIM", "MVR_PTS")
boxcox_vars <- c("BLUEBOOK", "CAR_AGE", "HOME_VAL", "INCOME", "TIF", "TRAVTIME", "CLM_FREQ")

# Apply conditional log transformation for high-skew variables with many zeros
df_ins_prep <- df_ins_prep %>%
  mutate(across(all_of(log_vars), ~ if_else(. == 0, 0, log(.)), .names = "log_{.col}"))

# Apply Box-Cox transformation using forecast package for other moderately skewed variables
df_ins_prep <- df_ins_prep %>%
  mutate(across(
    all_of(boxcox_vars),
    ~ forecast::BoxCox(. + 1, lambda = forecast::BoxCox.lambda(. + 1)), # Adding 1 to handle zeros
    .names = "{.col}_transform"
  ))

# Remove original variables except 'TARGET_AMT' to prevent redundancy
df_ins_prep <- df_ins_prep %>%
  dplyr::select(-all_of(c(log_vars[log_vars != "TARGET_AMT"], boxcox_vars)))
```

```
# Select only transformed variables for plotting
transformed_vars <- df_ins_prep %>% dplyr::select(dplyr::matches("log_|_transform$"))

# Plot histograms for transformed variables using DataExplorer
DataExplorer::plot_histogram(
  data = transformed_vars,
  geom_histogram_args = list(alpha = 0.5, fill = "lightblue", color = "black"),
  ggtheme = theme_minimal()
)

 #Feature Engineering
# Bucketing AGE
df_ins_prep <- df_ins_prep %>%
  mutate(AGE_GROUP = case_when(
    AGE < 30 ~ "Young",
    AGE >= 30 & AGE < 50 ~ "Middle-Aged",
    AGE >= 50 ~ "Older"
  ))


# Create Dummy Variables
df_ins_prep <- df_ins_prep %>%
  dplyr::select(-TARGET_FLAG) %>%
  dummy_cols(remove_first_dummy = TRUE) %>%
  bind_cols(df_ins_prep %>% dplyr::select(TARGET_FLAG))

# Define the list of original categorical columns
cat_cols <- c("PARENT1", "MSTATUS", "SEX", "EDUCATION", "JOB",
              "CAR_USE", "CAR_TYPE", "RED_CAR", "REVOKED", "URBANICITY", "AGE_GROUP")

# Remove the original categorical columns
df_ins_prep <- df_ins_prep %>%
  dplyr::select(-all_of(cat_cols))

# Check the structure to confirm the dummies
str(df_ins_prep)
describeBy(df_ins_prep)
```

```
# Handle Outliers
# Define a function for capping outliers using the 1st and 99th percentiles
cap_outliers <- function(x) {
  lower_bound <- quantile(x, 0.01, na.rm = TRUE)
  upper_bound <- quantile(x, 0.99, na.rm = TRUE)
  x <- ifelse(x < lower_bound, lower_bound, x)
  x <- ifelse(x > upper_bound, upper_bound, x)
  return(x)
}

# Select numeric variables, excluding target variables and missing flags
numeric_vars <- df_ins_prep %>%
  dplyr::select(where(~ is.numeric(.x) && !is.integer(.x)),
                -log_TARGET_AMT,
                -TARGET_AMT,
                -TARGET_FLAG,
                -contains("_MISSING_FLAG"))

# Apply the capping function to each numeric variable
df_ins_prep <- df_ins_prep %>%
  mutate(across(all_of(names(numeric_vars)), cap_outliers))

describeBy(df_ins_prep)
```

```
# Adjust plot layout for a larger boxplot
par(mar = c(8, 4, 4, 2) + 0.1, cex.axis = 0.8, las = 2)  # Increase bottom margin and axis text size

#Plot boxplot with enhanced settings
boxplot(df_ins_prep %>% dplyr::select(all_of(names(numeric_vars))),
        main = "Boxplot After Outlier Handling for Numeric Variables",
        col = "lightblue")
```

```
mlr_vars <- df_ins_prep %>%
  dplyr::select(-TARGET_AMT,
                -TARGET_FLAG,
                -contains("_MISSING_FLAG"))

mlr_vif_model <- lm(log_TARGET_AMT ~ ., data = mlr_vars)
print(vif(mlr_vif_model))

mlr_vars <- df_ins_prep %>%
  dplyr::select(-TARGET_AMT,
                -TARGET_FLAG,
                -contains("_MISSING_FLAG"),
                -CLM_FREQ_transform)

mlr_vif_model <- lm(log_TARGET_AMT ~ ., data = mlr_vars)
print(vif(mlr_vif_model))

mlr_scaled <- mlr_vars %>%
  dplyr::select(-log_TARGET_AMT) %>%
  mutate(across(where(~ is.numeric(.x) && !is.integer(.x)), scale))

mlr_scaled <- mlr_scaled %>%
  bind_cols(df_ins_prep %>% dplyr::select(TARGET_AMT, log_TARGET_AMT))

describeBy(mlr_scaled)

# ---


# Model Building

## MLR
# ---
# Set seed for reproducibility
set.seed(123)
trainIndex <- createDataPartition(mlr_scaled$log_TARGET_AMT, p = 0.5, list = FALSE)
mlr_train <- mlr_scaled[trainIndex, ]
mlr_valid <- mlr_scaled[-trainIndex, ]
```

```
# Fit full MLR model, using dplyr::select() to avoid conflicts with MASS::select()
mlr_full_model <- lm(log_TARGET_AMT ~ ., data = mlr_train %>% dplyr::select(-TARGET_AMT))
summary(mlr_full_model)
mlr_eval_pred_log <- predict(mlr_full_model, newdata = mlr_valid %>% dplyr::select(-TARGET_AMT))



stepwise_model <- stepAIC(mlr_full_model, direction = "both", trace = FALSE)
summary(stepwise_model)
mlr_stepwise_pred_log <- predict(stepwise_model, newdata = mlr_valid %>% dplyr::select(-TARGET_AMT))




# Define a function to calculate relevant metrics
eval_metrics <- function(true_values, predictions, num_predictors) {
  n <- length(true_values)
  rss <- sum((true_values - predictions)^2)
  tss <- sum((true_values - mean(true_values))^2)
  rsq <- 1 - rss / tss
  mse <- mean((true_values - predictions)^2)
  rmse <- sqrt(mse)

  # Calculate Adjusted R-squared
  adj_rsq <- 1 - ((1 - rsq) * (n - 1) / (n - num_predictors - 1))

  # F-statistic calculation
  f_stat <- (rsq / (1 - rsq)) * ((n - num_predictors - 1) / num_predictors)

  return(list(mse = mse, rmse = rmse, rsq = rsq, adj_rsq = adj_rsq, f_stat = f_stat))
}

# Calculate metrics for Full Model on validation set
metrics_full <- eval_metrics(mlr_valid$log_TARGET_AMT, mlr_eval_pred_log, num_predictors = length(coef(mlr_full_m
odel)) - 1)

# Calculate metrics for Stepwise Model on validation set
metrics_stepwise <- eval_metrics(mlr_valid$log_TARGET_AMT, mlr_stepwise_pred_log, num_predictors = length(coef(st
epwise_model)) - 1)

# Display results for both models
cat("Full Model Metrics:\n")
print(metrics_full)

cat("\nStepwise Model Metrics:\n")
print(metrics_stepwise)


# Calculate residuals for the full model on the validation set
mlr_eval_residuals_full <- mlr_valid$log_TARGET_AMT - mlr_eval_pred_log

# Calculate residuals for the stepwise model on the validation set
mlr_eval_residuals_stepwise <- mlr_valid$log_TARGET_AMT - mlr_stepwise_pred_log

# Set up the plotting area for side-by-side plots
par(mfrow = c(1, 2))

# Full model QQ plot
qqnorm(mlr_eval_residuals_full, main = "QQ Plot of Full Model")
qqline(mlr_eval_residuals_full, col = "red")

# Stepwise model QQ plot
qqnorm(mlr_eval_residuals_stepwise, main = "QQ Plot of Stepwise Model")
qqline(mlr_eval_residuals_stepwise, col = "red")

# Reset plotting area to default
par(mfrow = c(1, 1))




# Variable importance for Full Model
varImp_full <- varImp(mlr_full_model) %>%
  as.data.frame() %>%
  rownames_to_column("Variable") %>%
  top_n(6, wt = Overall)

# Variable importance for Stepwise Model
varImp_stepwise <- varImp(stepwise_model) %>%
  as.data.frame() %>%
  rownames_to_column("Variable") %>%
  top_n(6, wt = Overall)


# Plotting Full Model Variable Importance
ggplot(varImp_full, aes(x = reorder(Variable, Overall), y = Overall)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  coord_flip() +
  labs(title = "Top 6 Variables - Full Model", x = "Variable", y = "Importance") +
  theme_minimal()

# Plotting Stepwise Model Variable Importance
ggplot(varImp_stepwise, aes(x = reorder(Variable, Overall), y = Overall)) +
  geom_bar(stat = "identity", fill = "salmon") +
  coord_flip() +
  labs(title = "Top 6 Variables - Stepwise Model", x = "Variable", y = "Importance") +
  theme_minimal()


mlr_eval__stepwise_pred <- exp(mlr_stepwise_pred_log)

comparison_data <- data.frame(
  Actual = mlr_valid$TARGET_AMT,
  Predicted = mlr_eval__stepwise_pred
)

describeBy(comparison_data)

# Plot histograms for transformed variables using DataExplorer
DataExplorer::plot_histogram(
  data = comparison_data,
  geom_histogram_args = list(alpha = 0.5, fill = "lightblue", color = "black"),
  ggtheme = theme_minimal()
)


# ---

## Retrain selected MLR on full data


Retraining on the entire dataset allows the model to capture all available information, which generally improves
its robustness for future, unseen data predictions.

# ---
set.seed(123)
# Fit the stepwise model on the full dataset
final_mlr_stepwise_model <- stepAIC(
  lm(log_TARGET_AMT ~ ., data = mlr_scaled %>% dplyr::select(-TARGET_AMT)),
  direction = "both",
  trace = FALSE
)

# ---

## BLR


# ---

# Select variables for BLR VIF calculation
blr_vars <- df_ins_prep %>%
  dplyr::select(-TARGET_AMT,
                -log_TARGET_AMT,
                -contains("_MISSING_FLAG"))

# Fit the initial model for VIF calculation
```

```
blr_vif_model <- glm(TARGET_FLAG ~ ., data = blr_vars, family = binomial)
print(vif(blr_vif_model))



blr_vars <- df_ins_prep %>%
  dplyr::select(-TARGET_AMT,
                -log_TARGET_AMT,
                -contains("_MISSING_FLAG"),
                -log_OLDCLAIM)

blr_vif_model <- glm(TARGET_FLAG ~ ., data = blr_vars, family = binomial)
print(vif(blr_vif_model))

set.seed(123)
trainIndex <- createDataPartition(blr_vars$TARGET_FLAG, p = 0.8, list = FALSE)
blr_train <- blr_vars[trainIndex, ]
blr_val <- blr_vars[-trainIndex, ]

# Defining a recipe is like writing down steps to make a balanced cake (downsampled data).
# Baking is actually following steps to create the cake (balanced dataset).
table(blr_train$TARGET_FLAG)

downsample_recipe <- recipe(TARGET_FLAG ~ ., data = blr_train) %>%
  step_downsample(TARGET_FLAG) %>%
  prep()

downsampled_train <- bake(downsample_recipe, new_data = NULL)
table(downsampled_train$TARGET_FLAG)


logit_spec <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")

null_model <- logit_spec %>%
  fit(TARGET_FLAG ~ 1, data = downsampled_train)

summary(null_model$fit)


full_model <- logit_spec %>%
  fit(TARGET_FLAG ~ ., data = downsampled_train)

summary(full_model$fit)


# For the stepwise model, we're adding 2 specific feature selection steps that aren't applied to the null and ful
l models
stepwise_recipe <- recipe(TARGET_FLAG ~ ., data = downsampled_train) %>%
  step_nzv(all_predictors()) %>%
  step_corr(all_numeric_predictors(), threshold = 0.9) %>%
  prep()

stepwise_train <- bake(stepwise_recipe, new_data = NULL)

stepwise_model <- logit_spec %>%
  fit(TARGET_FLAG ~ ., data = stepwise_train)

summary(stepwise_model$fit)




# Updated evaluate_model to return the plot
evaluate_model <- function(pred_data, truth_col, pred_col, prob_col) {
  cm <- confusionMatrix(factor(pred_data[[pred_col]], levels = c("0", "1")),
                        reference = factor(pred_data[[truth_col]], levels = c("0", "1")),
                        positive = "1")

  auc_value <- roc(pred_data[[truth_col]], pred_data[[prob_col]], levels = c("0", "1"), direction = "<")$auc

  results <- list(
    Accuracy = cm$overall["Accuracy"],
    Error_Rate = 1 - cm$overall["Accuracy"],
    Kappa = cm$overall["Kappa"],
    Precision = cm$byClass["Precision"],
    Sensitivity = cm$byClass["Sensitivity"],
    Specificity = cm$byClass["Specificity"],
    F1_Score = cm$byClass["F1"],
    AUC = auc_value,
    Confusion_Matrix = cm$table
  )


  cm_plot <- as.data.frame(as.table(cm$table))
  cm_plot <- ggplot(cm_plot, aes(Reference, Prediction, fill = Freq)) +
    geom_tile() +
    geom_text(aes(label = Freq, color = Freq > 600), size = 4) +
    scale_fill_gradient(low = "lightblue", high = "steelblue") +
    scale_color_manual(values = c("black", "white"), guide = "none") +
    labs(title = "Confusion Matrix", x = "Actual", y = "Predicted") +
    theme_minimal()

  results$Confusion_Matrix_Plot <- cm_plot
  return(results)
}

# Generate predictions and results
null_preds <- predict(null_model, blr_val, type = "prob") %>%
  mutate(.pred_class = ifelse(.pred_1 >0.5, "1", "0"), TARGET_FLAG = blr_val$TARGET_FLAG)

full_preds <- predict(full_model, blr_val, type = "prob") %>%
  mutate(.pred_class = ifelse(.pred_1 > 0.5, "1", "0"), TARGET_FLAG = blr_val$TARGET_FLAG)

stepwise_preds <- predict(stepwise_model, blr_val, type = "prob") %>%
  mutate(.pred_class = ifelse(.pred_1 >0.5, "1", "0"), TARGET_FLAG = blr_val$TARGET_FLAG)

null_results <- evaluate_model(null_preds, "TARGET_FLAG", ".pred_class", ".pred_1")
full_results <- evaluate_model(full_preds, "TARGET_FLAG", ".pred_class", ".pred_1")
stepwise_results <- evaluate_model(stepwise_preds, "TARGET_FLAG", ".pred_class", ".pred_1")

list(
  Null_Model = null_results,
  Full_Model = full_results,
  Stepwise_Model = stepwise_results
)

# Arrange the plots side by side
grid.arrange(
  null_results$Confusion_Matrix_Plot + ggtitle("Null Model"),
  full_results$Confusion_Matrix_Plot + ggtitle("Full Model"),
  stepwise_results$Confusion_Matrix_Plot + ggtitle("Stepwise Model"),
  ncol = 3
)

# ---




## Retrain selected BLR on full data
Retraining on the entire dataset allows the model to capture all available information, which generally improves
its robustness for future, unseen data predictions.
# ---
set.seed(123)


downsample_recipe_final <- recipe(TARGET_FLAG ~ ., data = blr_vars) %>%
  step_downsample(TARGET_FLAG) %>%  # Downsample to handle class imbalance
  prep()

downsampled_final_data <- bake(downsample_recipe_final, new_data = NULL)


stepwise_recipe_final <- recipe(TARGET_FLAG ~ ., data = downsampled_final_data) %>%
  step_nzv(all_predictors()) %>%    # Remove near-zero variance predictors
  step_corr(all_numeric_predictors(), threshold = 0.9) %>%  # Remove highly correlated predictors
  prep()
```

```r
stepwise_final_data <- bake(stepwise_recipe_final, new_data = NULL)

final_blr_stepwise_model <- logit_spec %>%
  fit(TARGET_FLAG ~ ., data = stepwise_final_data)


# ---


# Prediction

# ---
test_data<- read.csv("https://raw.githubusercontent.com/yinaS1234/Auto-Insurance-Regression/refs/heads/main/data/
insurance-evaluation-data.csv")
index <- test_data$INDEX
str(test_data)
test_data<-test_data %>%dplyr::select(-TARGET_AMT, -TARGET_FLAG, -INDEX)


test_data <- test_data %>%
  mutate(
    INCOME = as.numeric(gsub("[$,]", "", INCOME)),
    HOME_VAL = as.numeric(gsub("[$,]", "", HOME_VAL)),
    OLDCLAIM = as.numeric(gsub("[$,]", "", OLDCLAIM)),
    BLUEBOOK = as.numeric(gsub("[$,]", "", BLUEBOOK))
  )

test_data <- test_data %>%
  mutate(across(where(~ !is.numeric(.)),
                ~ str_replace_all(., c("z_" = "", "<" = ""))))

DataExplorer::plot_histogram(
  data = test_data,
  geom_histogram_args = list(alpha = 0.5, fill = "lightblue", color = "black"),
  ggtheme = theme_minimal()
)


missing_flags_num <- test_data %>%
  dplyr::select_if(is.numeric) %>%
  summarise(across(everything(), ~ sum(is.na(.)) > 0)) %>%
  dplyr::select(where(~ . == TRUE)) %>%
  names()

test_data <- test_data %>%
  mutate(across(all_of(missing_flags_num), ~ ifelse(is.na(.), 1, 0), .names = "{.col}_MISSING_FLAG"))

# Create Missing Flags for Categorical Variables
test_data <- test_data %>%
  mutate(across(where(~ !is.numeric(.)), ~ na_if(., "")))



missing_flags_cat <- test_data %>%
  dplyr::select_if(~ !is.numeric(.)) %>%
  summarise(across(everything(), ~ sum(is.na(.)) > 0)) %>%
  dplyr::select(where(~ . == TRUE)) %>%
  names()

test_data <- test_data %>%
  mutate(across(all_of(missing_flags_cat), ~ ifelse(is.na(.), 1, 0), .names = "{.col}_MISSING_FLAG"))


# Visualize Missing Values
vis_miss(test_data, cluster = TRUE) +
  ggtitle("Missing Values") +
  theme(
    plot.title = element_text(face = "bold"),
    plot.margin = unit(c(1, 2, 1, 1), "cm")
  )


# Numeric variables: Regression-based imputation
impute_mice <- mice(test_data, method = "norm.predict", m = 1, remove.collinear = FALSE)
test_data <- complete(impute_mice)

# Categorical Variables: Mode-based Imputation
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

test_data <- test_data %>%
  mutate(across(where(~ !is.numeric(.)), ~ replace_na(., getmode(.))))

# Select all columns except missing flags for visualization
vars_without_flags <- test_data %>% dplyr::select(-contains("_MISSING_FLAG"))

# Visualize Missing Values (All Variables After Imputation)
vis_miss(vars_without_flags, cluster = TRUE) +
  ggtitle("All Variables - Imputed Missing Values") +
  theme(
    plot.title = element_text(face = "bold"),
    plot.margin = unit(c(1, 2, 1, 1), "cm")
  )

missing_flag_counts <- test_data %>%
  dplyr::select(contains("_MISSING_FLAG")) %>%
  summarise(across(everything(), sum))

# Display the count of 1s in each missing flag column
print(missing_flag_counts)

sapply(test_data, class)

# Define the list of columns to be converted to factors
factor_cols <- c("PARENT1", "MSTATUS", "SEX", "EDUCATION", "JOB",
                 "CAR_USE", "CAR_TYPE", "RED_CAR", "REVOKED", "URBANICITY")

# Convert the specified columns to factors and the rest to numeric
test_data <- test_data %>%
  mutate(across(all_of(factor_cols), as.factor)) %>%
  mutate(across(!all_of(factor_cols), as.numeric))

sapply(test_data, class)


# Define variables for conditional log transformation and Box-Cox transformation
log_vars <- c("OLDCLAIM", "MVR_PTS")
boxcox_vars <- c("BLUEBOOK", "CAR_AGE", "HOME_VAL", "INCOME", "TIF", "TRAVTIME", "CLM_FREQ")

# Apply conditional log transformation for high-skew variables with many zeros
test_data <- test_data %>%
  mutate(across(all_of(log_vars), ~ if_else(. == 0, 0, log(.)), .names = "log_{.col}"))

# Apply Box-Cox transformation using forecast package for other moderately skewed variables
test_data <- test_data %>%
  mutate(across(
    all_of(boxcox_vars),
    ~ forecast::BoxCox(. + 1, lambda = forecast::BoxCox.lambda(. + 1)), # Adding 1 to handle zeros
    .names = "{.col}_transform"
  ))

# Remove original variables
test_data <- test_data %>%
  dplyr::select(-all_of(c(log_vars, boxcox_vars)))



# Select only transformed variables for plotting
transformed_vars <- test_data %>% dplyr::select(dplyr::matches("log_|_transform$"))

# Plot histograms for transformed variables using DataExplorer
DataExplorer::plot_histogram(
  data = transformed_vars,
  geom_histogram_args = list(alpha = 0.5, fill = "lightblue", color = "black"),
  ggtheme = theme_minimal()
```

```r
)



 #Feature Engineering
# Bucketing AGE
test_data <- test_data %>%
  mutate(AGE_GROUP = case_when(
    AGE < 30 ~ "Young",
    AGE >= 30 & AGE < 50 ~ "Middle-Aged",
    AGE >= 50 ~ "Older"
  ))


# Create Dummy Variables
test_data <- test_data %>%
  dummy_cols(remove_first_dummy = TRUE)

# Define the list of original categorical columns
cat_cols <- c("PARENT1", "MSTATUS", "SEX", "EDUCATION", "JOB",
              "CAR_USE", "CAR_TYPE", "RED_CAR", "REVOKED", "URBANICITY", "AGE_GROUP")

# Remove the original categorical columns
test_data <- test_data %>%
  dplyr::select(-all_of(cat_cols))



# Handle Outliers
# Define a function for capping outliers using the 1st and 99th percentiles
cap_outliers <- function(x) {
  lower_bound <- quantile(x, 0.01, na.rm = TRUE)
  upper_bound <- quantile(x, 0.99, na.rm = TRUE)
  x <- ifelse(x < lower_bound, lower_bound, x)
  x <- ifelse(x > upper_bound, upper_bound, x)
  return(x)
}

# Select numeric variables, excluding target variables and missing flags
numeric_vars <- test_data %>%
  dplyr::select(where(~ is.numeric(.x) && !is.integer(.x)),  -contains("_MISSING_FLAG"))

# Apply the capping function to each numeric variable
test_data <- test_data %>%
  mutate(across(all_of(names(numeric_vars)), cap_outliers))

# Display summary statistics to confirm outlier handling
#describeBy(test_data)

# Adjust plot layout for a larger boxplot
par(mar = c(8, 4, 4, 2) + 0.1, cex.axis = 0.8, las = 2)  # Increase bottom margin and axis text size

# Plot boxplot with enhanced settings
boxplot(test_data,
        main = "Boxplot After Outlier Handling",
        col = "lightblue")

#str(test_data)


# Prepare MLR variables, excluding unwanted columns
mlr_vars <- test_data %>%
  dplyr::select(-contains("_MISSING_FLAG"), -CLM_FREQ_transform)

mlr_scaled <- mlr_vars %>%
  mutate(across(where(~ is.numeric(.x) && !is.integer(.x)), scale))

describeBy(mlr_scaled)


blr_vars <- test_data %>%
  dplyr::select(-contains("_MISSING_FLAG"),-log_OLDCLAIM)
describeBy(blr_vars)



# Predict the Probability of Crash (select second column if prob columns are unnamed)
blr_probabilities <- predict(final_blr_stepwise_model, new_data = blr_vars, type = "prob")[, 2]


TARGET_FLAG <- ifelse(blr_probabilities >= 0.5, 1, 0)

# Predict TARGET_AMT using the MLR model
mlr_log_predictions <- predict(final_mlr_stepwise_model, newdata = mlr_scaled)
TARGET_AMT <- ifelse(TARGET_FLAG == 1, exp(mlr_log_predictions), 0)

# Combine final results
results <- data.frame(
  Probability = blr_probabilities,
  TARGET_FLAG = TARGET_FLAG,
  TARGET_AMT = TARGET_AMT
)
colnames(results) <- c("Probability", "TARGET_FLAG", "TARGET_AMT")


# Reshape and plot
ggplot(melt(results), aes(value)) +
  geom_histogram(bins = 30, fill = "lightblue", color = "black") +
  facet_wrap(~ variable, scales = "free") +
  theme_minimal()





final_results <- cbind(index, results)
colnames(final_results)[1] <- "INDEX"

write.csv(final_results, "final_predictions.csv", row.names = FALSE)

# ---
```