



ITCS 6100 – Big Data for Competitive Advantage
Project Proposal Report – Loss Ratio Prediction for Insurance
Portfolios

TEAM- 1

Team Members:

NAME	STUDENT ID
Venkata Rajesh Alapati	801308822
Vineeth Kumar Aska	801328626
Riya Vinod Kalburgi	801338224
Pritesh Ambavane	801296733

Table of Contents

Introduction.....	3
Data Preparation	4
Data Exploration.....	7
Predictive Modeling	13
Findings	16

Introduction

In the complex landscape of insurance, where balancing premiums and claims is vital for a company's financial health, the ability to predict loss ratios accurately has emerged as a crucial challenge. Insurance companies rely on this fundamental metric – the ratio of claims paid out to premiums earned – to make strategic decisions about their pricing strategies. This delicate equilibrium, however, is influenced by a myriad of factors ranging from demographic details to external conditions like weather and economic fluctuations.

In this context, modern technology and data-driven methodologies offer a glimmer of hope. The integration of past customer data and various attributes associated with them has paved the way for predictive analytics. Enter this project, a pioneering initiative that aims to leverage the power of data science to unravel the complexities of loss ratio prediction.

At its core, this endeavor seeks to develop a robust model capable of accurately foreseeing the loss ratio within a portfolio of auto insurance policies. The dataset, rich with information about individual policies, including key attributes and financial parameters such as Annual Premium and Loss Amount, serves as the foundation for this predictive endeavor. The goal is not just a numerical prediction; it's about empowering insurance companies to make informed decisions, thereby enhancing their profitability and understanding of their customer base.

The journey begins with meticulous data preprocessing. Rigorous examinations will be conducted to identify and rectify any missing or null values, ensuring the dataset's integrity. Delving deeper, the project will sift through the data to pinpoint the most significant input features, discarding irrelevant columns that do not contribute substantially to the loss ratio prediction. A critical aspect lies in understanding the potential of each feature, assessing their impact on the model's predictive prowess. To avoid skewed interpretations, these features will be meticulously scaled, ensuring uniform weightage in the modeling process.

The heart of this project lies in the development and training of a predictive model. By harnessing the training data, the model will learn the intricate patterns within the datasets. Through iterative testing and validation, the model's performance will be rigorously evaluated. Only when the predicted values align closely with actual outcomes will the model be deemed ready for real-world application.

In essence, this project represents a synergy of cutting-edge technology, mathematical precision, and industry insight. By enabling insurance companies to peer into the future – albeit probabilistically – it holds the potential to revolutionize how premiums are priced and claims are managed. As we embark on this data-driven odyssey, the end goal remains crystal clear: empowering the insurance industry with predictive intelligence, one loss ratio at a time.

Data Preparation

Data Preparation is Divided into Following Steps

- i. Null value elimination
- ii. Removing Trailing and Leading Spaces in categorical columns
- iii. Replacing values of columns with appropriate values
- iv. Replacing categorical variables with dummy numeric variables

1. Null value elimination

We initially examined the data to identify any missing (null) values. After this examination, we identified two columns that contained null values. To address this, we replaced these null values with the respective modes (the most frequently occurring values) for those columns.

```
[5] missing_columns = data.columns[data.isnull().any()].tolist()
     print(missing_columns)

['Vehicle_Bodily_Injury_Limit', 'EEA_Prior_Bodily_Injury_Limit']

[6] for col in missing_columns:
     mode_value = data[col].mode()[0] # Calculate the mode
     data[col].fillna(mode_value, inplace=True)

[7] missing_columns = data.columns[data.isnull().any()].tolist()
     print(missing_columns)

[]
```

Figure 1: Modifying null values with mode

2. Removing Trailing and Leading Spaces in categorical columns

In our data cleaning, we made sure that the words in our categories don't have any extra spaces at the beginning or end. We did this by using the code below to remove these extra spaces.

```
cat_cols = data.select_dtypes(include='object')
print("Number of categorical_columns: ", cat_cols.shape[1])
```

Number of categorical_columns: 27

```
data[cat_cols.columns] = data[cat_cols.columns].apply(lambda x: x.str.strip())
```

Figure 2: Removing trailing and leading spaces in categorical data

3. Replacing values of columns with unknown values

We determined the columns that contained "unknown" values and displayed the value counts for these columns. This allowed us to assess and identify a more accurate estimate for replacing the "unknown" values.

```
# Replace "unknown" with the actual string that represents unknown values in your dataset
unknown_value = "unknown"

# Find columns with "unknown" values (case-insensitive search)
unknown_columns = [col for col in data.columns if data[col].astype(str).str.contains(unknown_value, case=False).any()]

columns_with_issues = list(unknown_columns)

# Get value counts for columns with issues
value_counts_for_columns_with_issues = {}

for col in columns_with_issues:
    counts = data[col].value_counts()
    value_counts_for_columns_with_issues[col] = counts

# Print the value counts for columns with issues
for col, counts in value_counts_for_columns_with_issues.items():
    print(f"Value counts for column '{col}':")
    print(counts)
    print()
```

Figure 3: Identifying columns with Unknown values

We determined that replacing the values with the mode (most common value) was the most suitable strategy for our data. As a result, we proceeded to replace the values in this manner.

```
[11] unknown_columns
      ['Policy_Zip_Code_Garaging_Location',
       'Vehicle_Make_Description',
       'Vehicle_Annual_Miles',
       'Vehicle_Anti_Theft_Device',
       'Vehicle_Passive_Restraint',
       'EEA_Policy_Zip_Code_3']

[12] for col in unknown_columns:
      mode_value = data[col].mode()[0] # Calculate the mode
      data[col] = data[col].str.replace(f'(?i){unknown_value}', mode_value, regex=True)

[13] # Find columns with "unknown" values (case-insensitive search)
      unknown_columns = [col for col in data.columns if data[col].astype(str).str.contains(unknown_value, case=False).any()]
      print(unknown_columns)

      ['Vehicle_Annual_Miles']
```

Vehicle_Annual_Miles, can be dropped as majority having unknown value.

Figure 4: Changing unknown values to mode.

4. Replacing Categorical Values with Numeric Values:

We've used the LabelEncoder from Scikit-Learn to convert all categorical columns in our DataFrame into numeric values. We looped through the selected categorical columns, applied the label encoding transformation to each column, and stored the resulting numeric values in the same DataFrame. The LabelEncoder helps us convert categorical data into a format suitable for machine learning algorithms or other data analysis tasks.

```
from sklearn.preprocessing import LabelEncoder

# Assuming 'data' is your DataFrame and 'categorical_columns' contains the names of categorical columns
categorical_columns = data.select_dtypes(include=['object']).columns

label_encoders = {} # Create a dictionary to store label encoders for each column

for column in categorical_columns:
    label_encoder = LabelEncoder()
    data[column] = label_encoder.fit_transform(data[column])
    label_encoders[column] = label_encoder

[50] data.head(5)
```

	Policy_Company	Policy_Installment_Term	Policy_Billing_Code	Policy_Method_Of_Payment	Policy_Reinstatement_Fee_Indicator	Policy_Zip_Code_Garaging
0	1	6	0	1	0	
1	1	6	0	1	0	
2	1	6	0	1	0	
3	1	6	0	1	0	
4	1	6	0	1	0	

5 rows x 47 columns

Figure 5: Replacing Categorical Values with Dummy Numeric Values

sData Exploration

1. Distribution of Annual Premiums and Loss Amount

As Annual_Premium and Loss_Amount determine the Loss_Ratio, plotting both graphs.

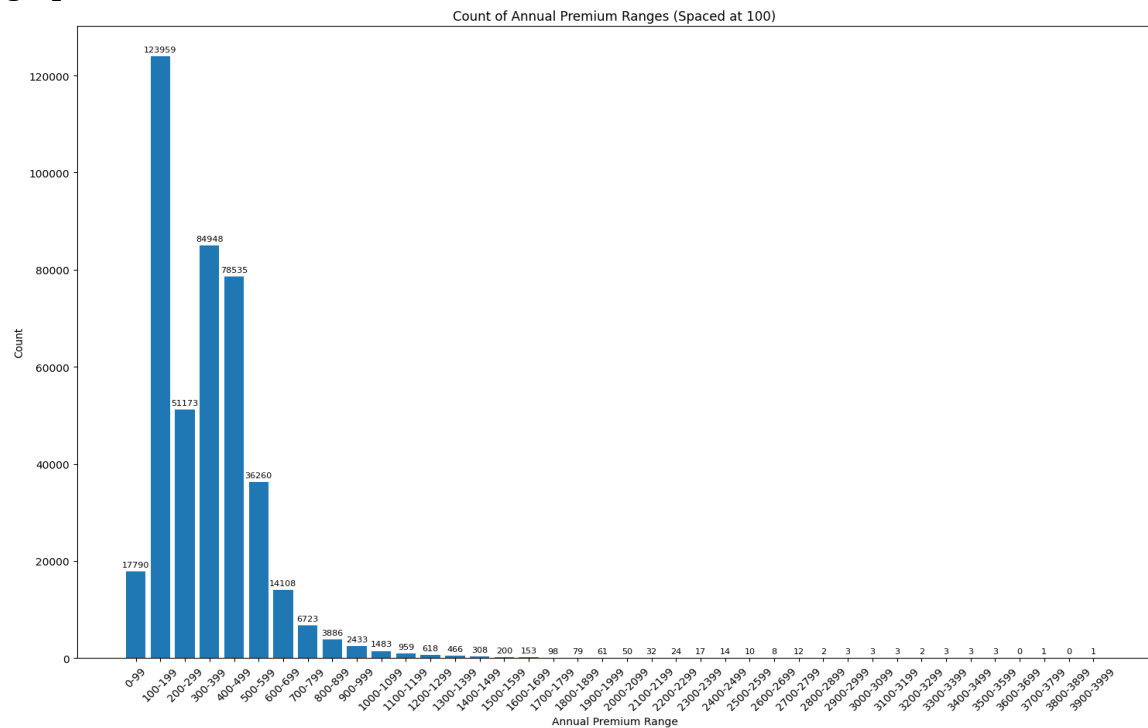


Figure 6: Distribution of Annual Premiums

Greater than 90% of the premiums are between 0-1000, with max between 100 - 200.

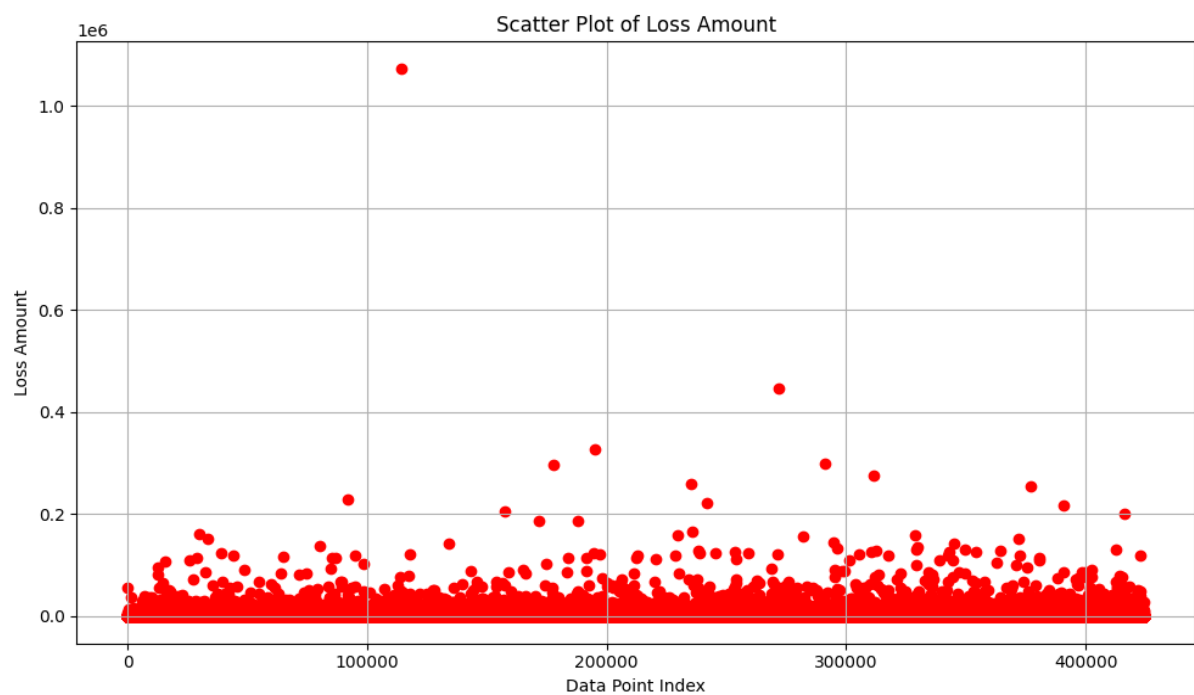
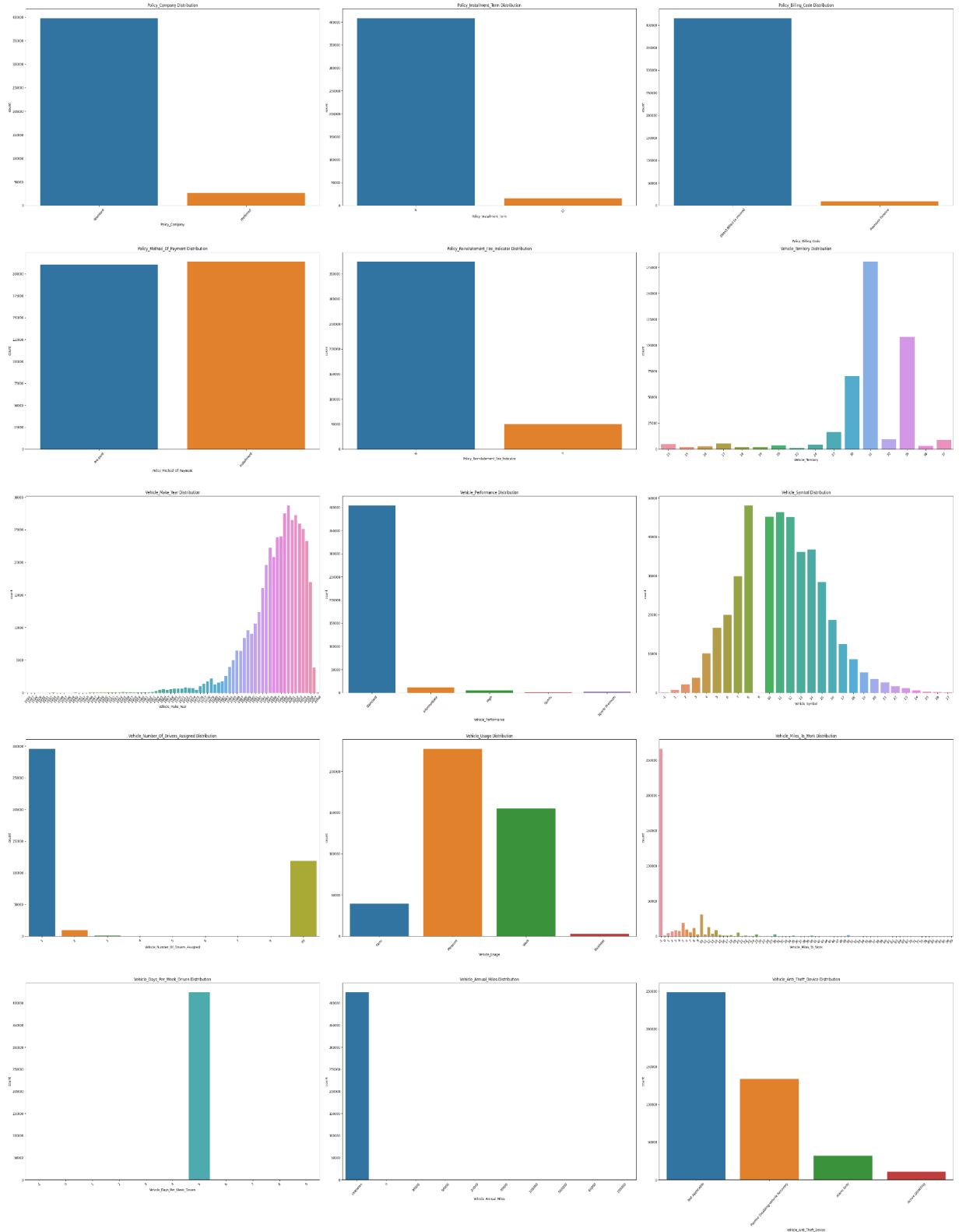


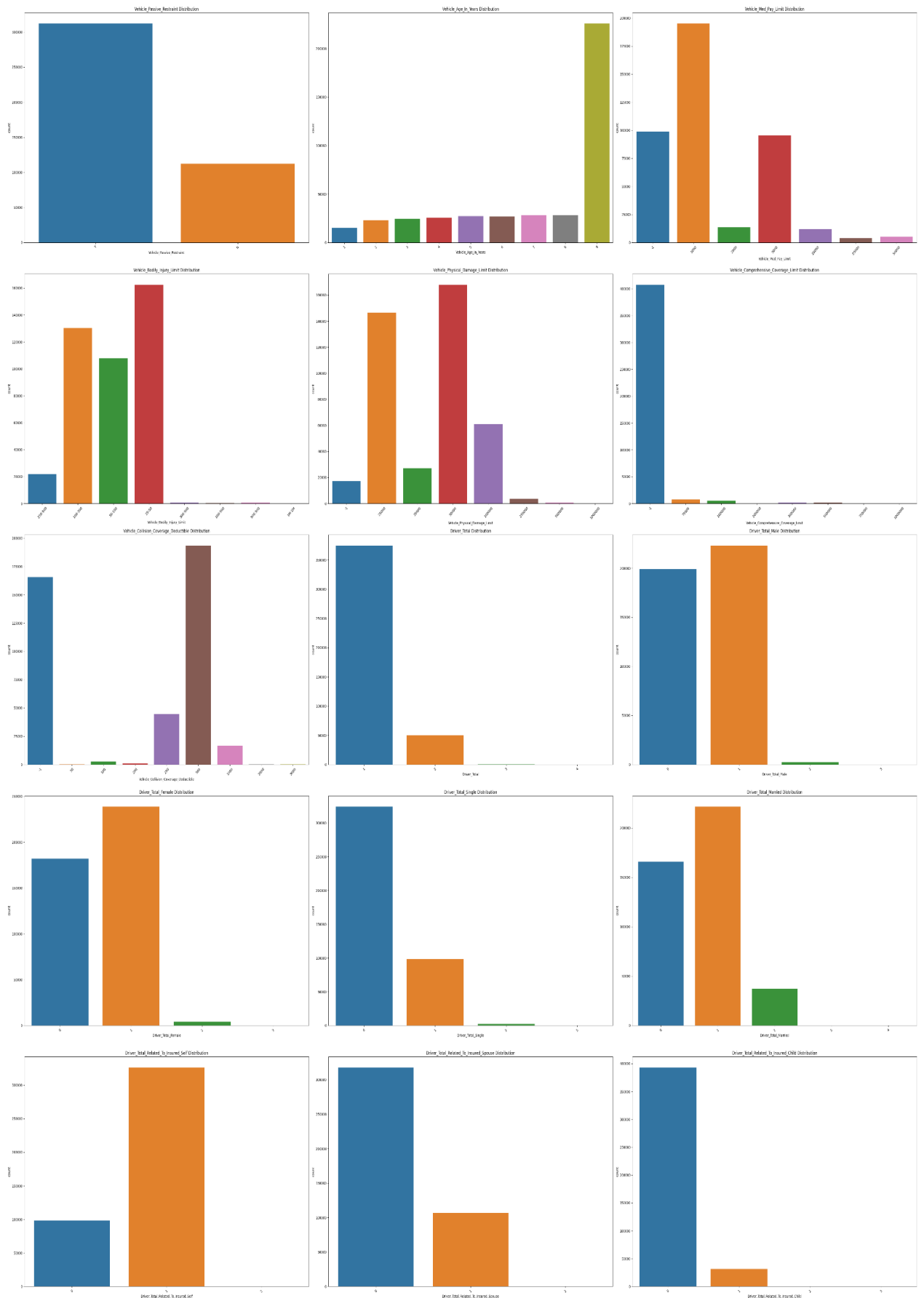
Figure 7: Distribution of the Loss Amount

Loss Amount is zero for the majority of the Premiums.

2. Observing the Data Distribution

We checked the distribution of various columns to identify and drop the columns that could be represented by other columns.





- **Driver_Minimum_Age** and **Driver_Maximum_Age**: These can be replaced with one column, **Driver_Age**, by taking the median of minimum and maximum age, as the majority of the rows have minimum and maximum age the same. This change allows us to remove the following columns, as all of them represent age.

Driver_Total_Teenager_Age_15_19, **Driver_Total_College_Ages_20_23,**
Driver_Total_Young_Adult_Ages_24_29,
Driver_Total_Low_Middle_Adult_Ages_30_39,
Driver_Total_Middle_Adult_Ages_40_49,
Driver_Total_Adult_Ages_50_64, **Driver_Total_Senior_Ages_65_69,**
Driver_Total_Upper_Senior_Ages_70_plus,
Vehicle_Youthful_Driver_Indicator.

3. Correlation between columns

We plotted the correlation heat map to understand the relationships between variables in the data, by displaying the pairwise correlations between them.

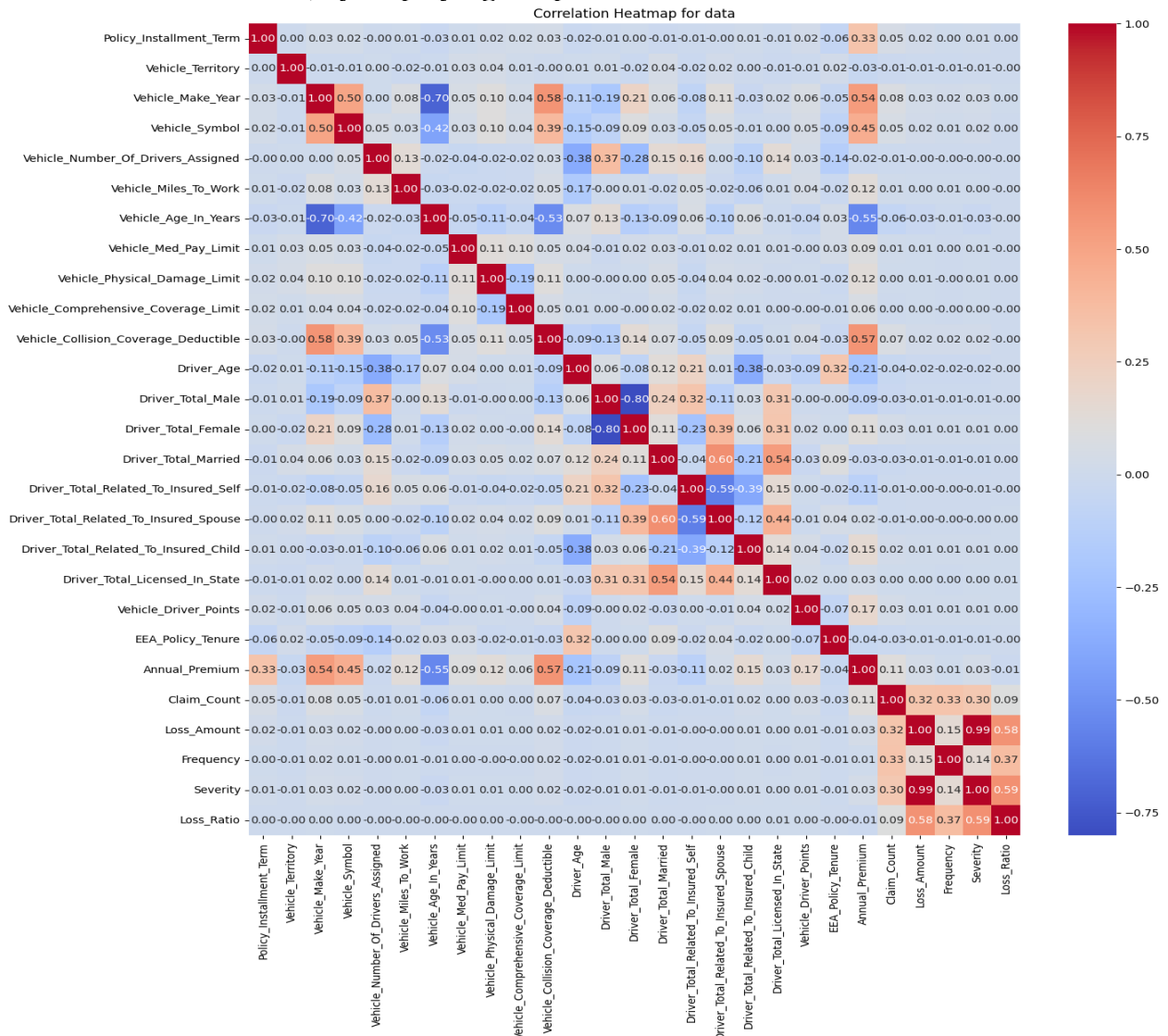


Figure 8: Correlation between the columns

Following are the observations:

Vehicle_Age_In_Years: Vehicle_make_year has similarities with Vehicle_Age_In_Years, so Vehicle_Age_In_Years can be dropped.

Severity: Severity is 99% correlated with loss_amount; loss_amount can represent severity and frequency combined.

Predictive Modeling

In the intricate realm of insurance, predicting loss ratios stands as a pivotal endeavor, influencing the very foundation of an insurer's financial stability. Loss ratio, denoting the proportion of incurred losses and adjustment expenses to earned premiums, serves as a vital indicator of an insurer's fiscal health. A high loss ratio hints at a precarious balance, where claims surpass premiums, potentially leading to substantial financial setbacks if not managed astutely.

Harnessing the power of machine learning models, insurers can not only foresee their future losses but also calibrate their pricing strategies in response. By deciphering the intricate interplay between various factors and loss ratios, insurers gain invaluable insights. This analytical prowess enables them to tailor underwriting and pricing strategies, mitigating risk exposure effectively.

Several methodologies come to the fore when predicting loss ratios, each offering distinct advantages:

1) Linear Regression:

Linear regression, a stalwart of statistical modeling, establishes a linear relationship between input variables (like policy type, age, and location) and the output variable (loss ratio). Employing a linear equation, this method predicts loss ratios by finding the best-fitting line through the data points. This approach assumes a linear connection between inputs and outputs, providing a foundational understanding of the relationships in play.

2) Decision Trees:

Decision trees, a versatile choice, handle both categorical and numerical input variables. Operating like a flowchart, the algorithm partitions data into subsets based on input variables until each subset shares similar loss ratios. The resulting tree translates into a set of if-then rules, mapping inputs to predicted loss ratios. Decision trees excel in capturing complex, non-linear relationships within the data, making them invaluable in nuanced insurance scenarios.

3) Gradient Boosting Regression:

Gradient boosting regression trees leverage the concept of ensemble methods, originating from individual decision trees. These trees, characterized by their hierarchical branching structure, start from the root node, follow specific conditions, and culminate in prediction outcomes at the leaves. However, a drawback surfaces when the tree hierarchy becomes excessively deep—overfitting on test data becomes a concern. To counter this issue, the concept of ensemble methods steps in. Instead of relying solely on a single decision tree, this technique combines the strengths of multiple decision trees, mitigating the risk of overfitting and enhancing predictive accuracy.

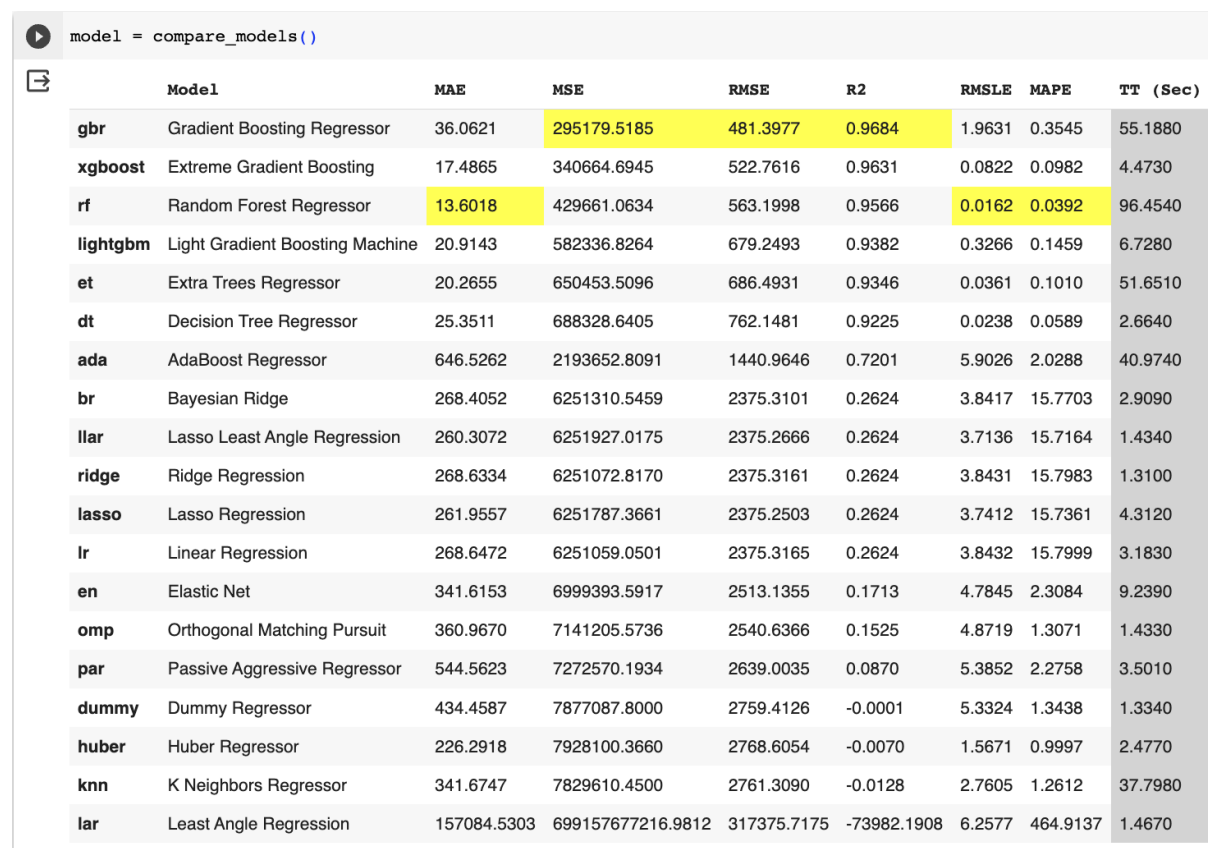
4) Random Forest:

Random forest, a pinnacle of ensemble learning, elevates predictive accuracy and robustness. This method amalgamates multiple decision trees, each trained on distinct data subsets. The amalgamation process hones in on the collective wisdom of individual trees, resulting in a prediction that outstrips the capabilities of individual models. Key to its prowess is:

- **Bootstrap Aggregation (Bagging):** Trees train on unique subsets of the data, reducing overfitting and enhancing generalizability.
- **Feature Subsampling:** The algorithm assesses only a subset of input variables at each split, curbing correlation between trees and bolstering overall model reliability.

Advantages of Random Forest in Loss Ratio Prediction:

- **Enhanced Accuracy:** By amalgamating diverse decision trees, random forest captures intricate input-output relationships, enhancing predictive accuracy.
- **Reduced Overfitting:** Bagging and feature subsampling curb overfitting, ensuring the model generalizes effectively to unseen data.
- **Robustness:** Random forest's resilience to outliers and missing data renders it particularly apt for the inherently variable nature of insurance datasets.



The image shows a screenshot of a Jupyter Notebook cell with the code `model = compare_models()` and a table of model comparison results. The table has columns for Model, MAE, MSE, RMSE, R2, RMSLE, MAPE, and TT (Sec). The Random Forest Regressor (rf) is highlighted as the best model with the lowest MAE (13.6018) and RMSLE (0.0162).

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
gbr	Gradient Boosting Regressor	36.0621	295179.5185	481.3977	0.9684	1.9631	0.3545	55.1880
xgboost	Extreme Gradient Boosting	17.4865	340664.6945	522.7616	0.9631	0.0822	0.0982	4.4730
rf	Random Forest Regressor	13.6018	429661.0634	563.1998	0.9566	0.0162	0.0392	96.4540
lightgbm	Light Gradient Boosting Machine	20.9143	582336.8264	679.2493	0.9382	0.3266	0.1459	6.7280
et	Extra Trees Regressor	20.2655	650453.5096	686.4931	0.9346	0.0361	0.1010	51.6510
dt	Decision Tree Regressor	25.3511	688328.6405	762.1481	0.9225	0.0238	0.0589	2.6640
ada	AdaBoost Regressor	646.5262	2193652.8091	1440.9646	0.7201	5.9026	2.0288	40.9740
br	Bayesian Ridge	268.4052	6251310.5459	2375.3101	0.2624	3.8417	15.7703	2.9090
llar	Lasso Least Angle Regression	260.3072	6251927.0175	2375.2666	0.2624	3.7136	15.7164	1.4340
ridge	Ridge Regression	268.6334	6251072.8170	2375.3161	0.2624	3.8431	15.7983	1.3100
lasso	Lasso Regression	261.9557	6251787.3661	2375.2503	0.2624	3.7412	15.7361	4.3120
lr	Linear Regression	268.6472	6251059.0501	2375.3165	0.2624	3.8432	15.7999	3.1830
en	Elastic Net	341.6153	6999393.5917	2513.1355	0.1713	4.7845	2.3084	9.2390
omp	Orthogonal Matching Pursuit	360.9670	7141205.5736	2540.6366	0.1525	4.8719	1.3071	1.4330
par	Passive Aggressive Regressor	544.5623	7272570.1934	2639.0035	0.0870	5.3852	2.2758	3.5010
dummy	Dummy Regressor	434.4587	7877087.8000	2759.4126	-0.0001	5.3324	1.3438	1.3340
huber	Huber Regressor	226.2918	7928100.3660	2768.6054	-0.0070	1.5671	0.9997	2.4770
knn	K Neighbors Regressor	341.6747	7829610.4500	2761.3090	-0.0128	2.7605	1.2612	37.7980
lar	Least Angle Regression	157084.5303	699157677216.9812	317375.7175	-73982.1908	6.2577	464.9137	1.4670

Figure 9: Model comparison using pycaret for predictive modeling

In the realm of insurance, where precision can dictate financial prosperity, the choice of predictive modeling techniques is pivotal. Through the synergy of linear regression, decision trees, and the formidable random forest, this project is poised to unravel the complexities of loss ratio prediction. By embracing these sophisticated methodologies, the team endeavors to provide insurers with actionable insights, shaping a future where financial prudence and data-driven decisions coalesce seamlessly.

Findings

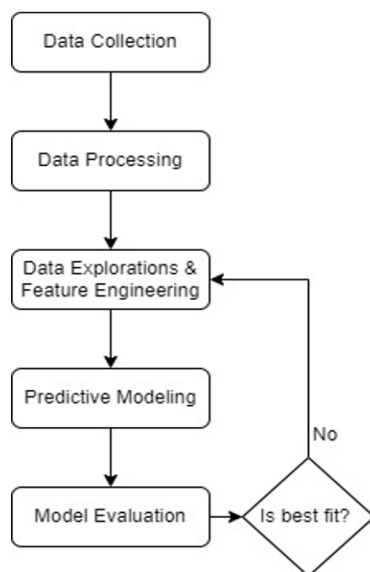


Figure 10: Project Workflow

The progression of our project is akin to a meticulously crafted tapestry, where each stage intricately weaves together insights and data-driven decisions. At its core, understanding the intricacies of the provided data was paramount. Careful data preprocessing was conducted to lay a robust foundation for the subsequent phases, particularly the pivotal data exploration stage.

Our journey commenced with a comprehensive comprehension of the provided information, ensuring a judicious selection of data preprocessing techniques. Addressing the challenge of missing data, we employed sophisticated methods such as imputation and data interpolation, particularly for numerical attributes.

A significant aspect of our preprocessing endeavor involved the strategic modification of attributes.

Attributes were meticulously interpolated using existing data from other related characteristics, ensuring a nuanced approach tailored to the unique attributes of the dataset. A vivid example of this innovative interpolation was demonstrated in the transformation of the `Driver_Age` column.

Upon completing the preprocessing phase, the features in the dataset were comprehensively analyzed, utilizing tools like bar graphs and histograms. These visual representations not only shed light on the diverse characteristics of the data but also spotlighted any discrepancies that could potentially influence model training.

During the data exploration stage, our focus intensified on feature extraction, a pivotal step in shaping the project's outcomes. Plotting intricate graphs unearthed nuanced insights into features, illuminating their underlying patterns and distributions. However, this stage presented a unique challenge in the form of multicollinearity—a complex web of interconnections between attributes. To tackle this challenge, correlation analyses were meticulously executed. Beyond evaluating the relationship between individual features and the target variable, we delved into the correlations between attributes themselves. This nuanced evaluation allowed us to discern superior features within highly correlated pairs, ensuring meticulous feature engineering and the precise creation of results.

In essence, our approach transcends the conventional boundaries of data exploration and predictive modeling, where, as an initial testament to the predictive power, we have implemented the `pycaret` run on the preprocessed data to identify how different models perform on the given dataset. The models Gradient Boosting Regression and Random Forest Regression outperform other regression models. Through this holistic methodology, we not only explore the data's depths but also chart a precise course toward generating insights that are not just valuable but transformative in the realm of insurance loss ratio prediction.