Team 4 Project

Vehicle Insurance Prediction

Ammar Al-baadani, Erwin Linares, Huu Dang Nguyen, Sean Cairns

The University of Texas at Dallas

ITSS 4382 – Applied Artificial Intelligence/Machine Learning

Professor Lidong Wu

December 1, 2023

Table of Contents

Problem Statement

With the rising number of vehicles that are sold annually, and the growing number of new drivers has led to a substantial demand for vehicular insurance. The domain of insurance has been experiencing a rising cost of vehicle insurance which raises concerns for both insurance providers and policyholders. Inflation, coupled with various socio-economic factors, has led to significant fluctuations in insurance premium rates making the market unpredictable and sometimes not fair to some people. The objective of this project is to assess the impact of inflation, social impact, and other different factors on vehicle insurance prices and create a model that can accurately predict the premium of individuals so that it is affordable, fairness, and allows for competitiveness for both the providers and the policyholders.

Project Purpose

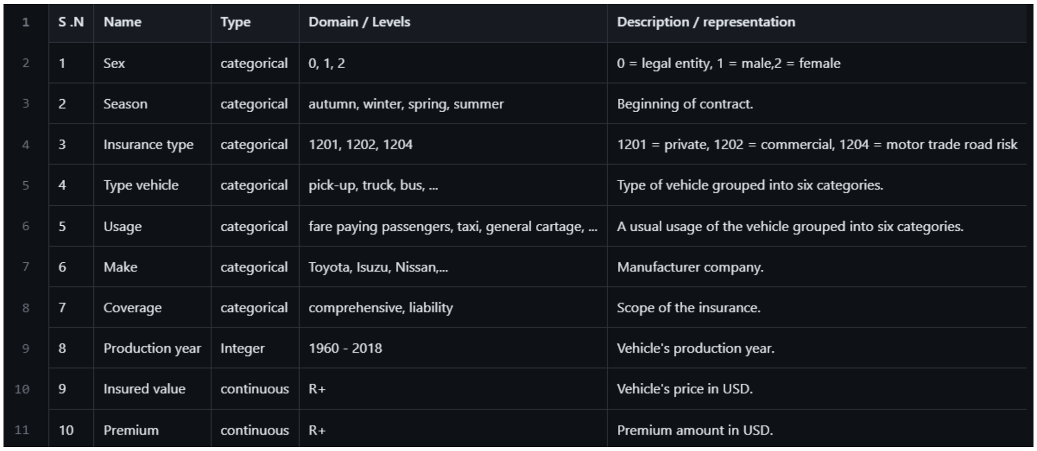
Develop a predictive model for auto insurance premiums to address rising costs. The goal is to offer insurance companies and policyholders a tool for navigating economic fluctuations, fostering stability, competitiveness, and affordability in the insurance market.

Tools selection

- Tableau – data visualization for EDA
- Alteryx – models and features selection
- Jupiter Notebook (Python) - data cleansing, EDA, modeling, and evaluation

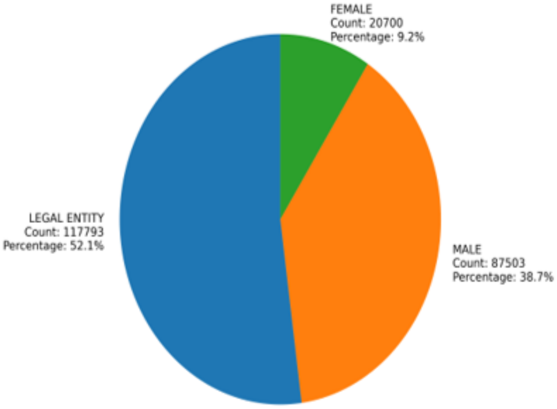
Exploratory Data Analysis (EDA)

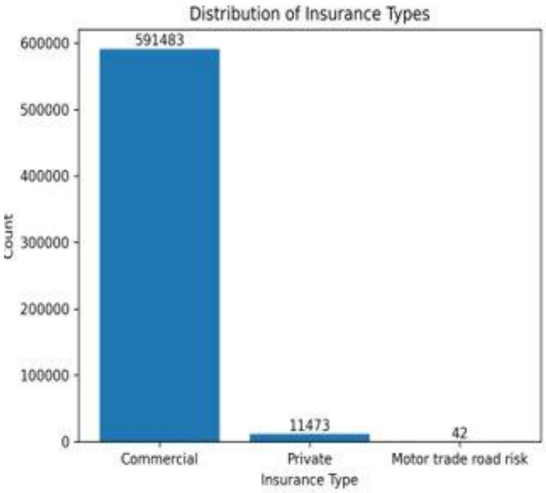
- The dataset was obtained from the Ethiopian Insurance Corporation
- Containing 16 fields and 802,036 records, were saved as 2 separate csv files.
- Dataset URL - https://www.kaggle.com/datasets/imtkaggleteam/vehicle-insurance-data
- Here is Metadata with some features that we want to highlight

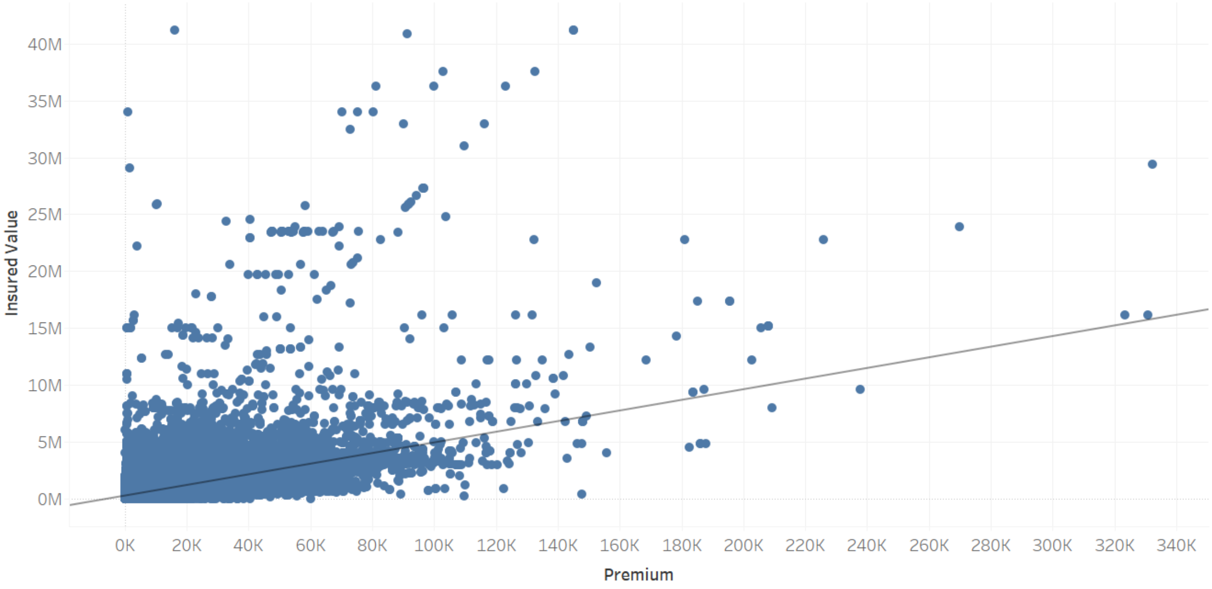


S .N	Name	Type	Domain / Levels	Description / representation
1	Sex	categorical	0, 1, 2	0 = legal entity, 1 = male,2 = female
2	Season	categorical	autumn, winter, spring, summer	Beginning of contract.
3	Insurance type	categorical	1201, 1202, 1204	1201 = private, 1202 = commercial, 1204 = motor trade road risk
4	Type vehicle	categorical	pick-up, truck, bus, ...	Type of vehicle grouped into six categories.
5	Usage	categorical	fare paying passengers, taxi, general cartage, ...	A usual usage of the vehicle grouped into six categories.
6	Make	categorical	Toyota, Isuzu, Nissan,...	Manufacturer company.
7	Coverage	categorical	comprehensive, liability	Scope of the insurance.
8	Production year	Integer	1960 - 2018	Vehicle's production year.
9	Insured value	continuous	R+	Vehicle's price in USD.
10	Premium	continuous	R+	Premium amount in USD.

Here are some data visualizations that our team did to show the overall picture of the data







- We have 3 groups in the “Sex” features which are Legal Entity (52.1%), Female (9.2%) and Male (38.7%)
- We have three types of insurance which are labeled as shown in Metadata. The distribution is 591483 for commercial insurance, 11473 for private insurance, and 42 for motor trade road risk type of insurance.

Data Cleansing

This meticulous data cleansing process ensures the integrity and reliability of our dataset for subsequent modeling and analysis phases.

In the initial phase of our project, we extracted a zip file from Kaggle of two distinct insurance datasets originating from an Ethiopian insurance company, each spanning a four-year period (2011-2014 and 2014-2018). The datasets shared identical attributes, encompassing fields such as SEX, INSR_begin, INSR_END, Effective_yr, INSR_Type, INSURED_VALUE, PREMIUM, OBJECT_ID, PROD_YR, SEATS_NUM, CARRYING_CAPACITY, TYPE_VEHICLE, CCM_TON, MAKE, USAGE, and CLAIM_PAID. Our objective was to concatenate these datasets using Pandas dataframes, resulting in a combined dataframe named 'Combined.'

Handling Null Values: Identification and management of null values are imperative for dataset integrity. The 'combined.isnull().sum()' function exposes attributes with substantial null values, notably 'CLAIM_PAID' and 'CARRYING_CAPACITY.' The former, associated with insurance claims, was addressed by replacing null values with zeros, considering the likelihood that many policyholders had no claims. The latter, pertaining to vehicle carrying capacity, underwent nuanced treatment. Initial attempts at null value deletion were followed by a reevaluation that prompted imputation based on specific criteria. Median values, specific to car makes and usages, were employed to ensure a more accurate representation of carrying capacity.

Data Cleansing and Imputation Strategies: The code details various strategies employed to cleanse and impute data. Correcting typographical errors in the 'MAKE' column, filtering and imputing values based on conditions (e.g., 'FIAT' buses for carrying capacity), and addressing anomalous entries (e.g., 'MERCEEDES') all contribute to refining the dataset.

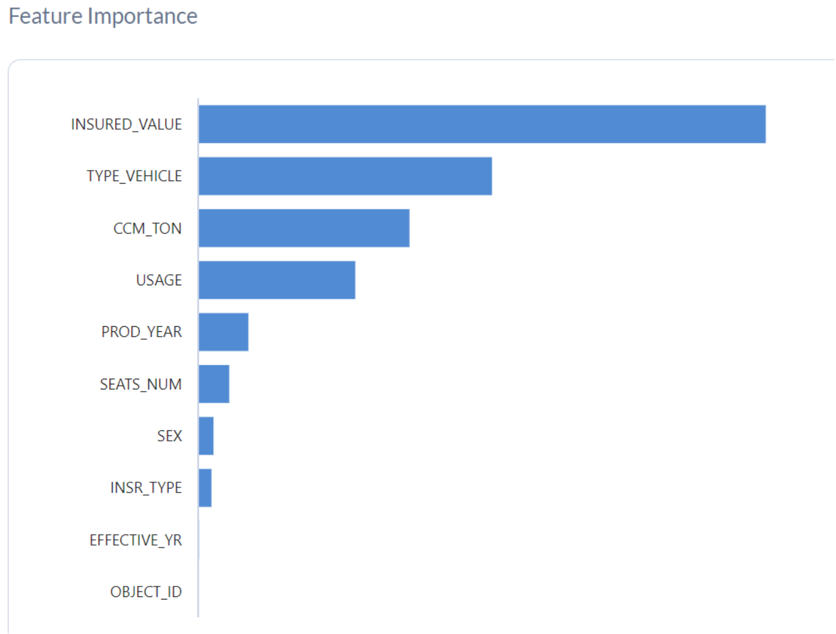
Multicollinearity Assessment: The essay underscores the significance of assessing multicollinearity concerns through scatter plots, which visualize relationships between variables. The absence of significant dependencies among variables is crucial for model robustness. The decision not to drop any attributes based on multicollinearity reflects a judicious approach, considering the potential impact on the model's predictive power.

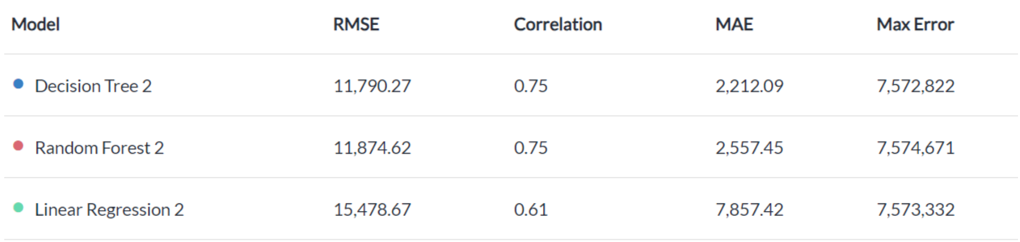
Data Validation and String Corrections: The code introduces data validation steps, such as dropping records with minimal impact on the model and correcting string values in the 'EFFECTIVE_YR' column. The correction of typographical errors in the 'MAKE' column, such as 'MERCEEDES,' ensures consistency and accuracy in subsequent analyses.

Imputing Values for Specific Conditions: The essay delves into detailed strategies for imputing values under specific conditions. For instance, it describes the replacement of zero values with Na for certain vehicle makes and types, followed by the calculation and application of medians to ensure a more accurate representation of these attributes.

Models and Features Selection

Our team runs the data through Assited Machine Learning Modeling in Alteryx to see the importance of features and which models should be selected and tested on.

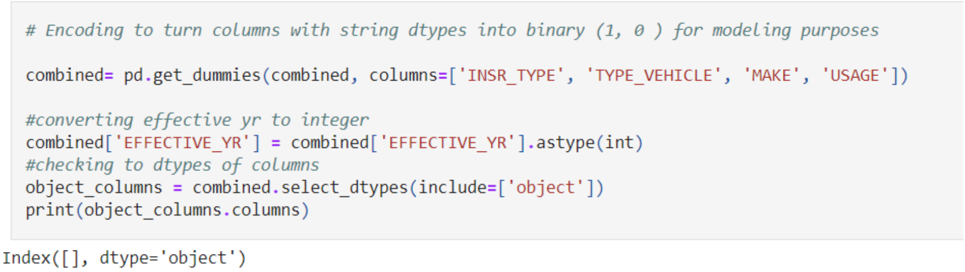




Model	RMSE	Correlation	MAE	Max Error
● Decision Tree 2	11,790.27	0.75	2,212.09	7,572,822
● Random Forest 2	11,874.62	0.75	2,557.45	7,574,671
● Linear Regression 2	15,478.67	0.61	7,857.42	7,573,332

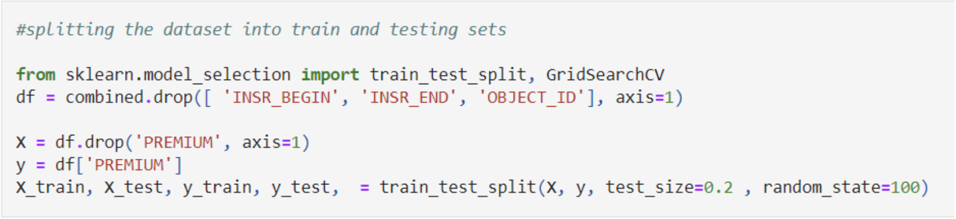
Machine Learning Modeling

- First, we prepared data ready for modeling



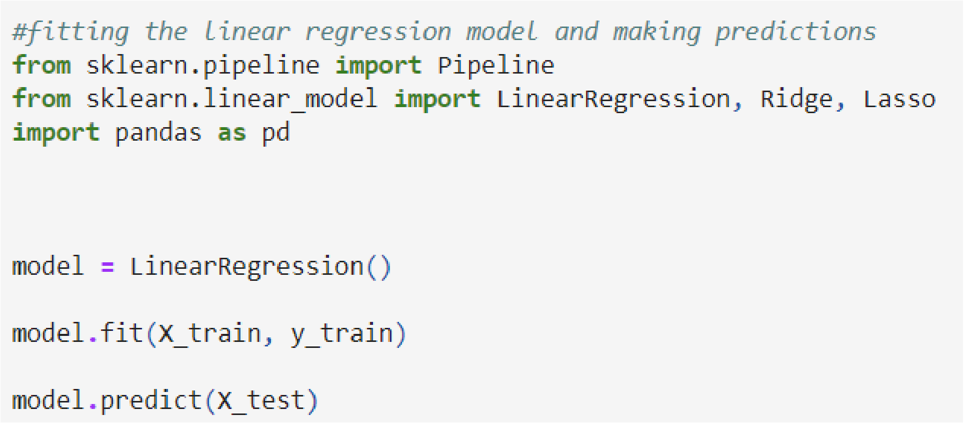
```
# Encoding to turn columns with string dtypes into binary (1, 0 ) for modeling purposes

combined= pd.get_dummies(combined, columns=['INSR_TYPE', 'TYPE_VEHICLE', 'MAKE', 'USAGE'])

#converting effective yr to integer
combined['EFFECTIVE_YR'] = combined['EFFECTIVE_YR'].astype(int)
#checking to dtypes of columns
object_columns = combined.select_dtypes(include=['object'])
print(object_columns.columns)
```

```
Index([], dtype='object')
```



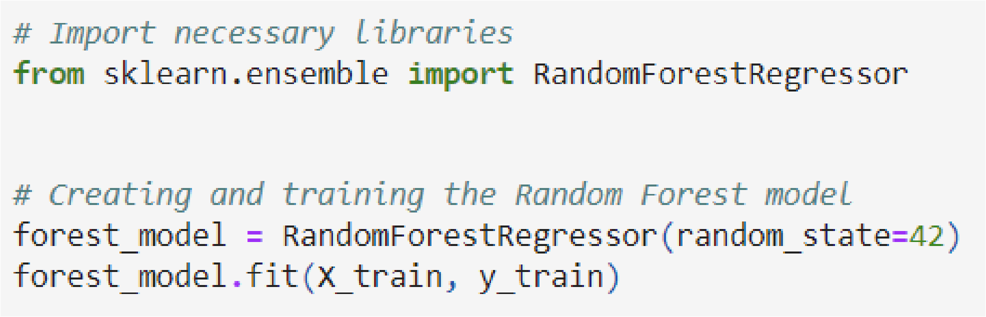
```
#splitting the dataset into train and testing sets

from sklearn.model_selection import train_test_split, GridSearchCV
df = combined.drop([ 'INSR_BEGIN', 'INSR_END', 'OBJECT_ID'], axis=1)

X = df.drop('PREMIUM', axis=1)
y = df['PREMIUM']
X_train, X_test, y_train, y_test,  = train_test_split(X, y, test_size=0.2 , random_state=100)
```

Linear Regression



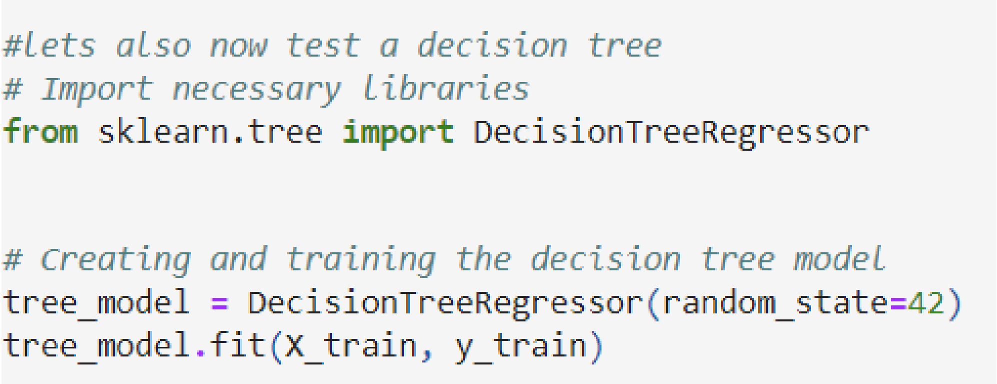
```
#fitting the linear regression model and making predictions
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LinearRegression, Ridge, Lasso
import pandas as pd



model = LinearRegression()

model.fit(X_train, y_train)

model.predict(X_test)
```

Random Forest



```
# Import necessary libraries
from sklearn.ensemble import RandomForestRegressor


# Creating and training the Random Forest model
forest_model = RandomForestRegressor(random_state=42)
forest_model.fit(X_train, y_train)
```

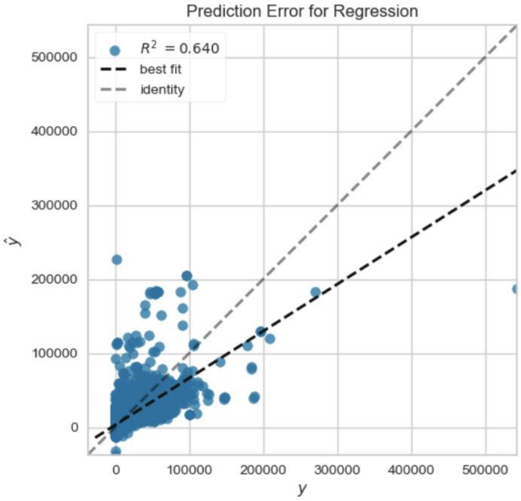
Decision Tree

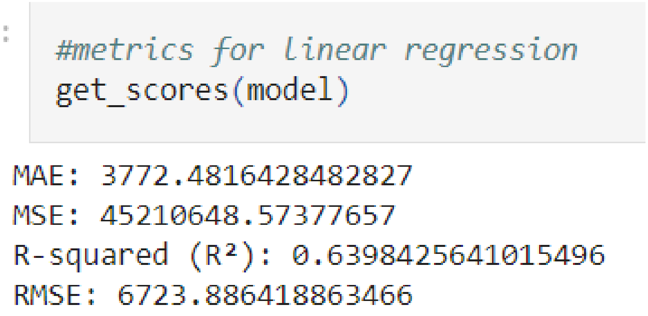


```
#lets also now test a decision tree
# Import necessary libraries
from sklearn.tree import DecisionTreeRegressor


# Creating and training the decision tree model
tree_model = DecisionTreeRegressor(random_state=42)
tree_model.fit(X_train, y_train)
```

Models Evaluation

Linear Regression



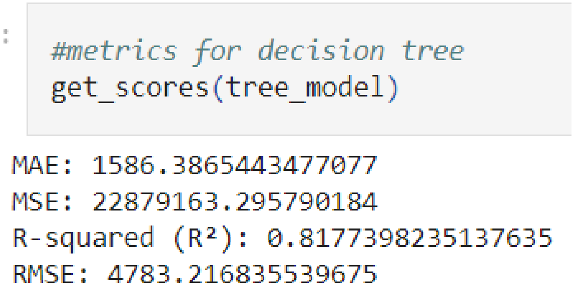


```
#metrics for linear regression
get_scores(model)
```

```
MAE: 3772.4816428482827
MSE: 45210648.57377657
R-squared (R²): 0.6398425641015496
RMSE: 6723.886418863466
```

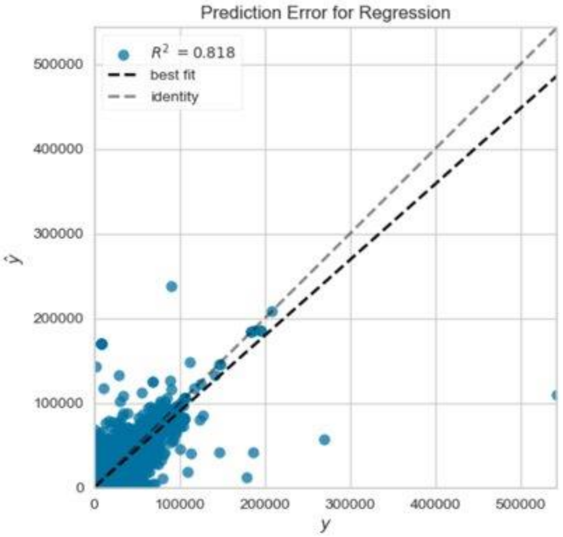
The Linear Regression model metrics indicate that the model explains around 64% for the variance in insurance premium. The average error was about $3772.48 in predicting individual premiums.

Decision Tree



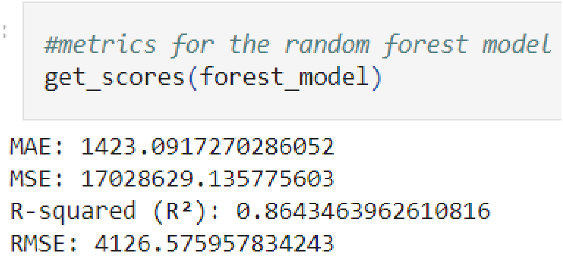
```
#metrics for decision tree
get_scores(tree_model)
```

```
MAE: 1586.3865443477077
MSE: 22879163.295790184
R-squared (R²): 0.8177398235137635
RMSE: 4783.216835539675
```



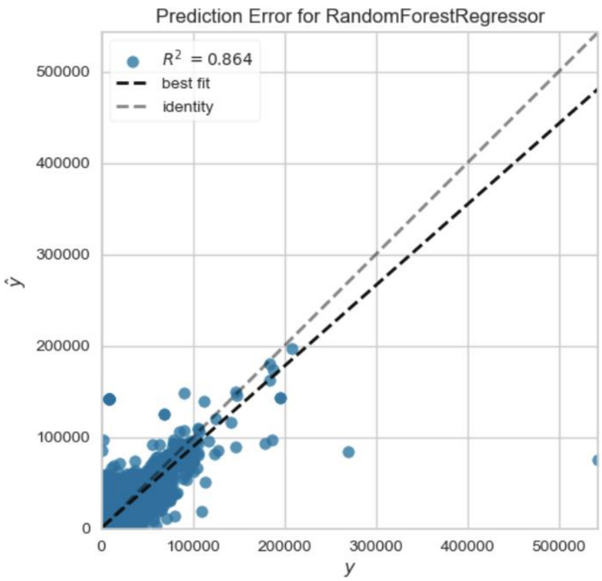
The Decision Tree model metrics show that the model fits better than the linear regression model. It explains around 82% of the variance in insurance premium with an average error of approximately $1586.38.

Random Forest



```
#metrics for the random forest model
get_scores(forest_model)
```

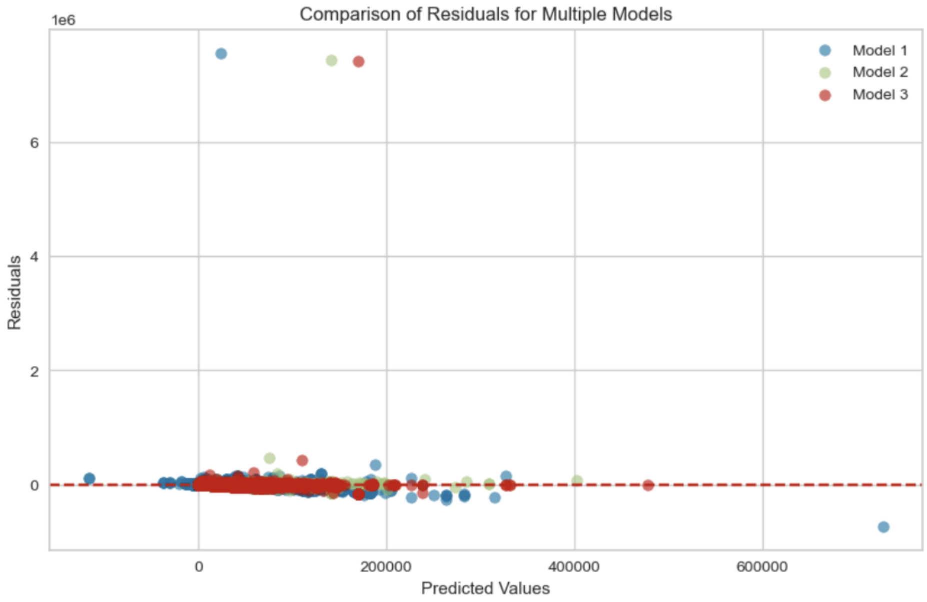
```
MAE: 1423.0917270286052
MSE: 17028629.135775603
R-squared (R²): 0.8643463962610816
RMSE: 4126.575957834243
```



Lastly the Random Forest Model showed superior performance with all metrics showing better performance. The model can explain 86% of the variance in insurance premium with an average error of $1423.09.

Conclusion

We can clearly see that in the three models, random forest model gave us the highest R-squared and the least errors. We went ahead and created a visualization to compare residuals of the three models together.



Our suggestions for further improvement

To enhance data quality in insurance premium prediction models, it is imperative to systematically address missing values, outliers, and skewed distributions within the dataset. This involves employing robust data preprocessing techniques to fill in missing information, identify and handle outliers, and normalize skewed variables, ensuring a more accurate and reliable input for the predictive model. Moreover, incorporating essential features such as mileage, safety ratings, and driver history further refines the precision of premium predictions, capturing nuanced aspects of risk assessment. To bolster predictive accuracy, a holistic approach involves amalgamating various prediction methodologies, allowing the model to harness the strengths of different algorithms. Lastly, continuous model updates are crucial for adaptability to evolving trends in the insurance landscape, enabling the system to stay relevant and effective in the face of changing variables and industry dynamics. This comprehensive strategy not only elevates the quality of data but also ensures a resilient and responsive insurance premium prediction model.