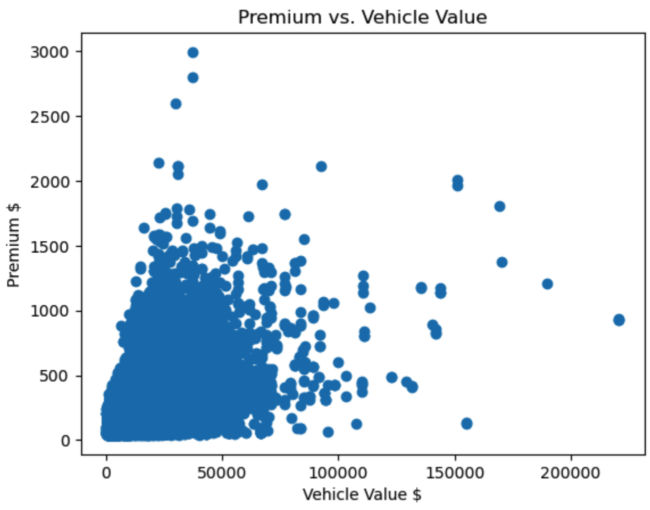
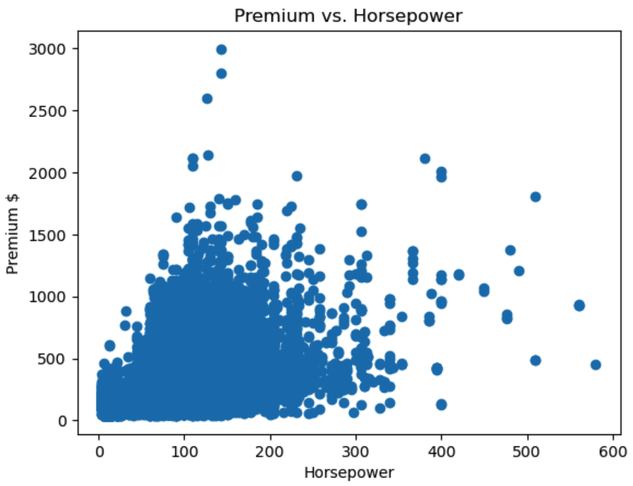
Car Insurance Predictive Model

The automotive industry is a massive market that many people interact with everyday. Largely, these interactions are driven by the purchase and sale of cars. With purchasing a new car, there are lots of obvious decisions surrounding the make, model, color, and other features based on customer preferences. However, one of the less contemplated aspects is understanding the subsequent costs that come with your car. The price is one thing, but an often overlooked cost inherent with owning a car is the insurance associated. I intend to build out a model that can predict insurance costs based on a variety of features within a car. This will assist any customer looking to understand their all in costs for a car, ultimately allowing them to make a more informed decision and avoid unexpected costs.

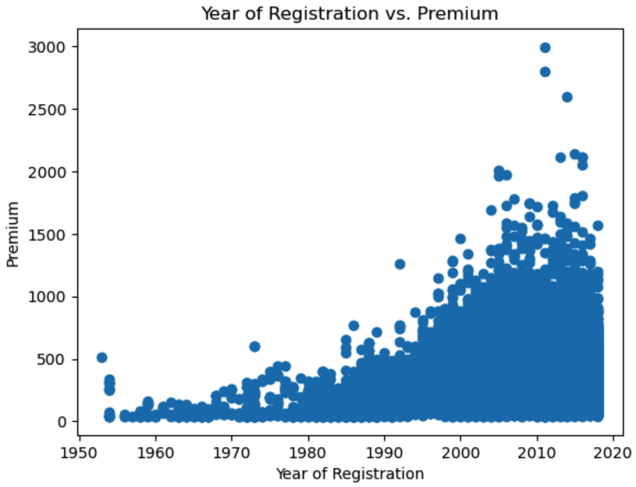
The dataset I will be using to build this model is one from a motor vehicle insurance portfolio, accessible through Mendeley Data with over 100k entries. It includes features such as Insurance Premium, Number and Cost of claims, renewal date, value of the vehicle, power, and other car features/ policy details.

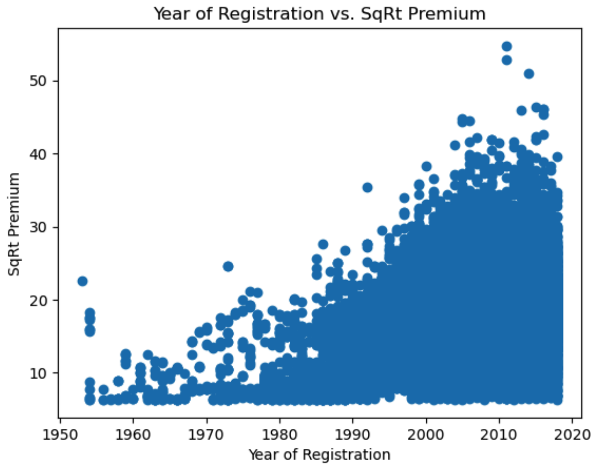
After importing, my first task was to clean it to ensure all data types were consistent with what the feature was representing and there were no missing values. I also sought to add any features based off the information I had that I thought may be good predictors of my target variable (Premium). One of the features I added was the age of the customer as this is sometimes a factor in coming to the premium. After this step, I was left with approximately 30 columns and just about 100k rows. I began analyzing the data and quickly realized the Premium variable was quite dispersed with a right skew. Running the correlation function on my dataset made it clear that there were few variables that were directly correlated with the Premium. Below are the two most correlated features, Vehicle Value, and Horsepower:





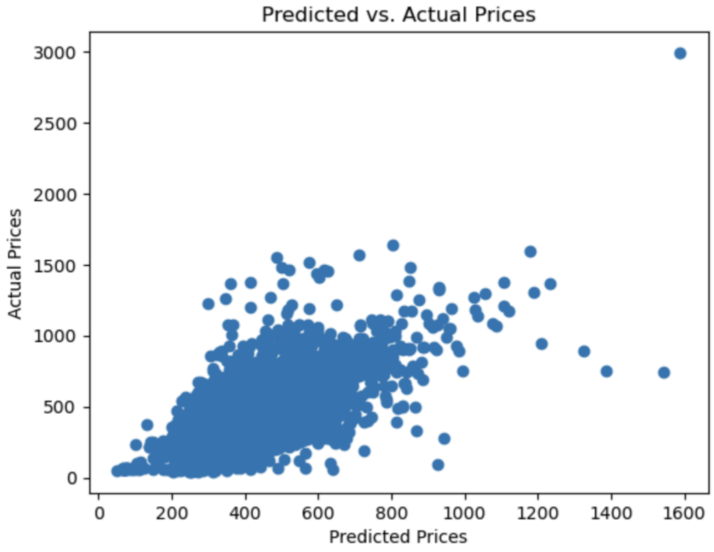
As you can see, neither of these features appeared to show strong correlation with a pearson correlation coefficient of approximately .35. Another feature that I found interesting was the Year of the Registration appeared to be exponentially correlated with insurance, however taking the square root did not improve this correlation enough to be effective:





It appeared there were many outliers, so I began running analysis on keeping only the instances where Premium lied within a 5%-95% interval although this also did not appear to be effective in increasing correlations.

I decided to move forward with my analysis of Linear Regression thinking if I had multiple features with OK correlation, maybe I could still get some results. The error metrics I used to evaluate my model included the r2_score, and RMSE as this would tell me on average how far off in price I was from actuals. I first ran an OLS and Linear Regression model, inclusive of all features. Both of these performed relatively poorly with r2_scores in the .25 range. I then ran a Ridge Regression model and Lasso model to see if weighing the features helped improve scores at all. With these two models, I did a GridSearchCV to evaluate the best hyperparameters to include for the model. Even with this, my model was still coming out to approximately .25 r2_score, with a RMSE of almost $120. The last model I ran was a Random Forest model that yielded much different results. Out of the box, this model was calculating a training r2_score of almost .90 and a RMSE of $30. I decided to further evaluate this and run a GridSearch as well as a for loop to determine the different training and test scores based on different hyperparameters. I was quite pleased to see the jump in training data predictive power, however when it came to the test data, my model was coming up around .57, or $89 RMSE. Below is a scatterplot included from the final model:



The final hyperparameters included using a Random Forest Regressor used n_estimators at 1000, and a max depth of 25. Additionally, after looking at the GridSearch, I determined the most important features were the value of the vehicle, the year of matriculation, and the Age of the insured.

I do not necessarily believe this model is useful for customers at this point, but I do believe further feature engineering/ modeling can be done to increase the predictive power of a Tree based model. One potential further evaluation would be to run a clustering algorithm on the dataset to try and find similarities in car types and see how that affects the premium. Additionally, a different Tree based model can and should be tested on this dataset to see if it produces any better results than the Random Forest method.