Predicting the Price of Used Cars Using Machine Learning Algorithms

Shin Le, Jeongyeon Kim, Benjamin Horvath, Nicolas Reategui, Paul Giglio
Florida State University

I. An introduction to the problem in question including potential applications

In this project we will apply various machine learning regression algorithms to a dataset of used car sales, determining if a suitable model can be made from information commonly available to consumers to predict the sales price of the car. We will be using Multiple Linear Regression, Decision Trees, Random Forest, Gradient Boosting, and Support Vector Machine, each with the goal of building a suitable model for predicting sales price. Should multiple models be found to be successful we will seek to determine which model is best suited to the task of predicting used car sales prices.

II. A thorough literature survey with appropriate references

Attempting to predict the sales prices of used cars by applying data mining techniques has been a popular subject for many years now, with many papers written on the effectiveness of various models in predicting the sales price. Pudaruth[1] attempted to find a suitable model for predicting the sales price of used cars in Mauritius by using multiple linear regression, k-nearest neighbors, naïve Bayes, and decision tree algorithms. Pudaruth found that naïve Bayes and decision tree techniques were ineffective at predicting the sales price of used cars (with between 60-70% accuracy), primarily due to the need to discretize the final predicted sales price which increased the scale of inaccuracies in predictions. Further, the study suffered overall from a very small collection of data (97 records), which hindered the effectiveness of all of the techniques applied.

Another analysis of the effectiveness of various methods in predicting the sales price of used cars by Chen et al.[2] attempted to determine the effectiveness of random forest models against linear regression models on a dataset of over 100,000 used car purchases in China. They concluded that while linear regression models are accurate within car series, random forests were more accurate when building a universal model, and the added workload of splitting large datasets by individual car series could both make the models inaccurate due to low sample sizes within car series, as well as adding a significant workload to the creation of the model. However, while random forests were concluded to be superior to linear regression models, it was acknowledged that the computational cost of random forests when applied to large datasets was significantly higher than the cost to build a universal linear regression model.

Monburinon et al.[3] once again attempted to compare the effectiveness of various regression models in predicting used car sales prices by comparing multiple linear regression, random forest regression, and gradient-boosted regression trees. Monburinon et al. created models for each of the regression techniques to be used and applied them to the same dataset, and using Mean Absolute Error as their criterion for success found that random forests were superior to the multiple linear regression model (MAE=0.35 vs. MAE=0.55), and that gradient boosted regression trees performed the best with a MAE=0.28. The reason why the random forest is superior to the multiple linear regression model is that the MAE is lower. When comparing the models, the lower MAE index is considered better in terms of accuracy. They concluded that gradient-boosted regression trees were the recommended method for developing prediction models for used car prices.

Noor and Jan[4] attempted to create an accurate model for predicting the sales price of used cars, focusing their efforts solely on creating a suitable multiple linear regression model. Data was collected from “PakWheels”, a popular online company for reselling cars in Pakistan, and gathered 1699 records after preprocessing. With this suitably large dataset and clever variable selection, they achieved a surprisingly high R^2 value above 98%, proving the potential effectiveness of multiple linear regression when applied to an adequate dataset.

III. Description of the methodology used

- **Multiple Linear Regression:** We started with the Sequential Feature Selection method to choose some significant variables for our model. With multiple linear regression, we can see the changes in the model and the relative contribution each independent variable has to the total variance. We started with this more complex model with the intent to simplify it with later models.

- **Decision Tree:** The goal of the decision tree is to take our many data features, learn simple decision rules inferred by those features, and predict our target variable, in this case, our car price. Using Mean Absolute Error (MAE) and Mean Square Error (MSE), we went with a Pruned Tree, as Pruning reduces the size of decision trees by removing parts of the tree that do not provide power to classify instances. Decision trees are the most susceptible out of all the machine learning algorithms to overfitting and effective pruning can reduce this likelihood.

- **Ensemble Models: Random Forest Regression:** Random Forest is a supervised machine-learning algorithm made up of decision trees. Using the bagging technique to collect various samples, we make many trees to form the bootstrap samples, which are all combined to make the Random Forest.

- **Ensemble Models: Gradient Boosting Regression:** Gradient Boosting is a variant of Ensemble Models where we use multiple weak models to combine them to get better performance as a whole. Using MAE and MSE, we improve the model with each iteration as we run it, get the MAE and MSE, and alter the model using those parameters to improve it each time.

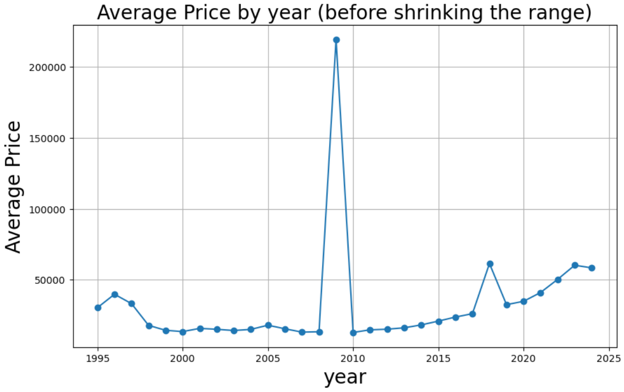
- **Support Vector Machine (SVM):** Support Vector Machine is a type of supervised learning algorithm used for machine learning, which is usually used to solve classification and regression tasks. The model graphs the samples in an n-dimensional space, where n is the number of features, and attempts to create a hyperplane that separates groups of features. For our dataset, we elected to use the linear kernel due to our larger number of records and features.

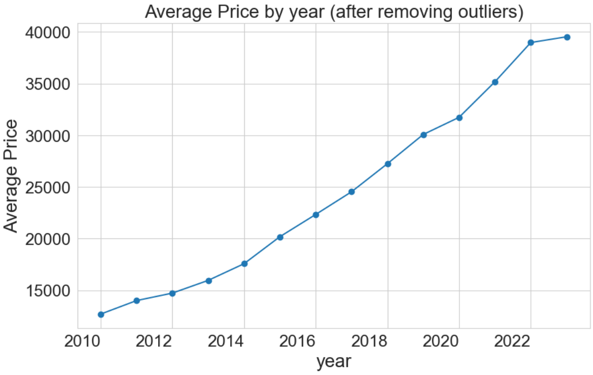
IV. Description of the implementation, potential problems, and assumptions

1. **Data collection:**
   Our data was sourced from the “Used Cars Dataset” by Andrei Novikov published on Kaggle. The dataset contains sales data from 762,091 cars sold on cars.com, collected in April 2023, with 20 features related to the car or information about the sale.

2. **Data preprocessing:**
   We split the ‘mpg’ column into its city mpg and highway mpg components and then filled missing values with the mean of the column. We then dropped all rows still containing missing values. For categorical features with lots of rare values, we identified the most common values and dropped all rows that had rare values for these features. Once we had the most common categorical values we used label encoding to convert the values in place into their numeric representation. We then identified outliers in our numeric columns and removed rows containing these outliers. To identify many of these outliers and rare values we created visualizations of the content of each feature, visually identifying problems within our dataset before cleaning our data using identical methods across features.

- The plot shows us that there are a lot of outliers and the average price in 2009 is abnormal. We removed cars with unusually high sales prices and shrunk our range of cars from 1995-2023 to 2010-2023 due to the comparatively small number of records in the years 1995-2009.

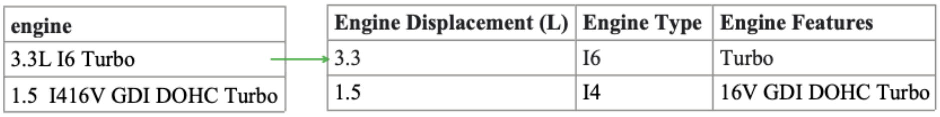




3. Exploratory Data Analysis (EDA):

a. Filtering Data:

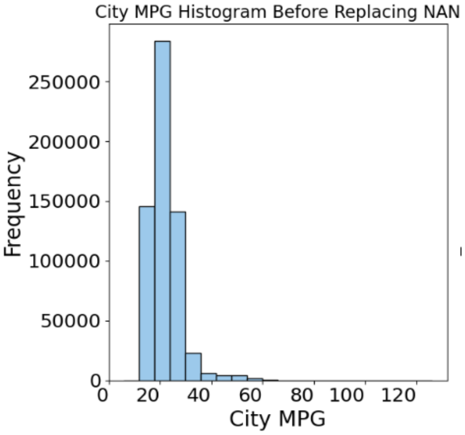
- The engine feature was divided into its subset features, ‘Engine Displacement’, ‘Engine Type’, and ‘Engine Features’, and dropped the original engine column.

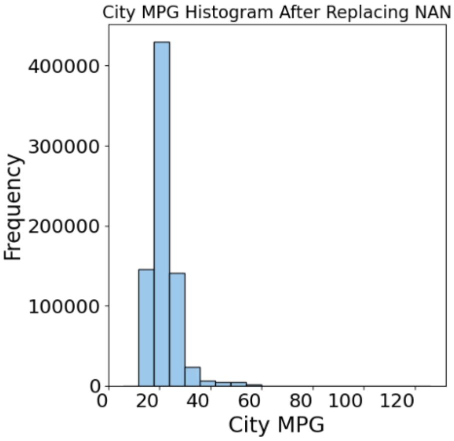


engine
3.3L I6 Turbo
1.5 I416V GDI DOHC Turbo

Engine Displacement (L)	Engine Type	Engine Features
3.3	I6	Turbo
1.5	I4	16V GDI DOHC Turbo

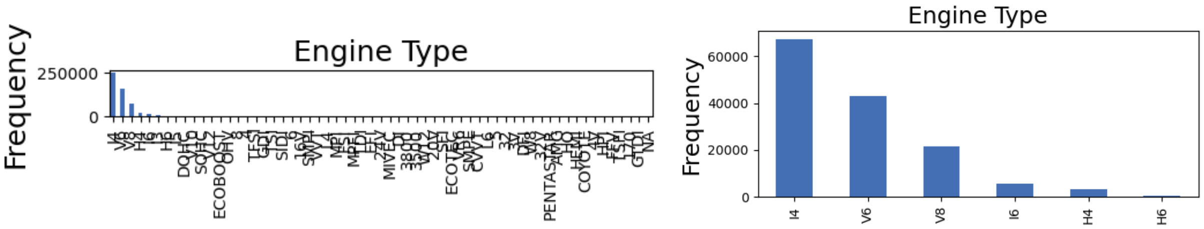
- Missing value:
  For numerical values, we replaced missing values with the mean. Our goal is to keep the initial property of data after replacing the missing values. For example,





b. Detecting Outliers:

- For *numerical features*, we can detect the outlier by using the interquartile range (IQR) to identify and remove outliers. We decided to drop the rows that contained outliers as we still had over 300,000 samples, and doing so still preserved the integrity of our dataset, whereas attempting to replace the outliers across all of these features could have reduced the quality of our data.
- For *categorical features*, We can detect the outliers by plotting the distinct values of each feature against the frequency of that value. We kept the most frequent values for each feature and removed all records from our dataset which had rare features.



An example of reducing categorical features to manageable numbers by removing rare features

c. *Labeling Encode*

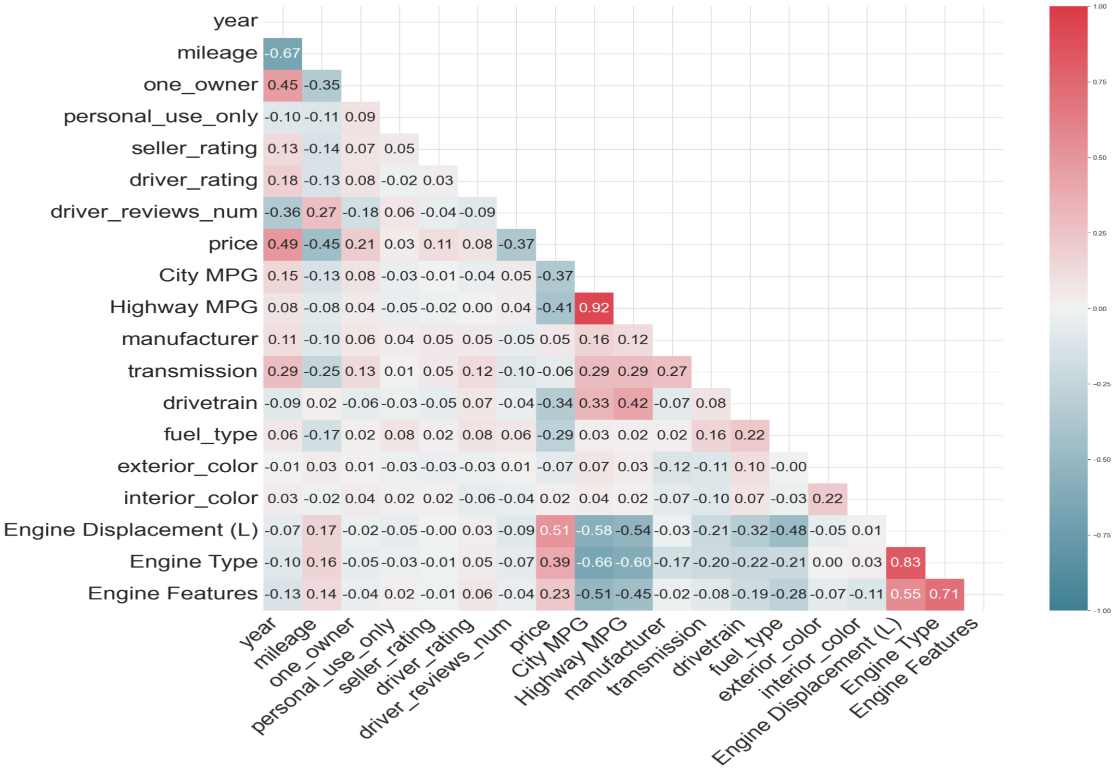
Our dataset has many categorical features that contain many classes. For many of scikit-learn's models to be able to work with the contents of our dataset these categorical feature values needed to be converted into numerical values. For this, we decided to use label encoding, which replaced the distinct values in each feature with a number representing that feature. For example, we label the values in "**transmission":**

(before label encoding) transmission=['Automatic CVT', '6-Speed Manual', '8-Speed Automatic', 'Automatic', '9-Speed Automatic', '10-Speed Automatic', '8-Speed Automatic with Auto-Shift', 'Manual', '10-Speed Automatic with Overdrive', '6-Speed Automatic', '6-Speed Electronically Controlled Automatic with O', '7-Speed Automatic', 'Automatic Xtronic CVT', '7-Speed', '7-Speed Automatic with Auto-Shift', '5-Speed Manual', '4-Speed Automatic', '6-Speed', '5-Speed Automatic', '6-Speed Automatic with Auto-Shift']

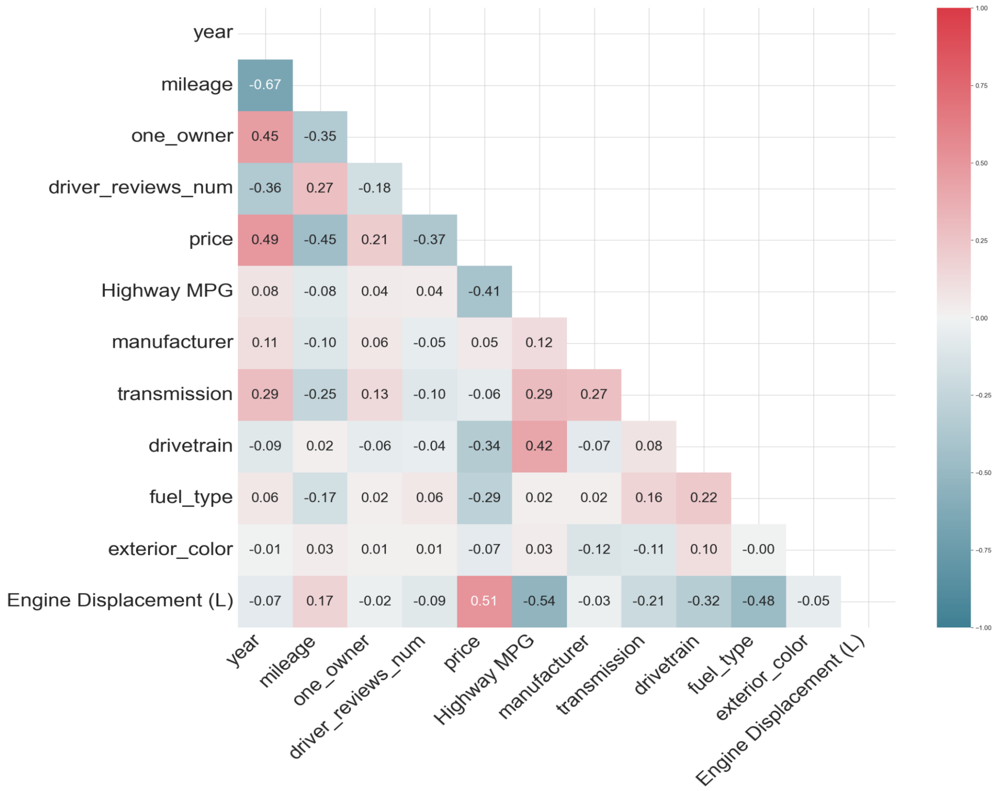
(after label encoding) transmission=[17, 9, 16, 15, 0, 19, 13, 1, 6, 8, 10, 11, 4, 12, 14, 18, 3, 5, 2, 7]

d. *Correlation Matrix*

The correlation matrix shows us the correlation coefficients between variables. A correlation coefficient close to 1 indicates a strong positive relationship, meaning when one variable increases, the other tends to increase. A coefficient near -1 signifies a strong negative relationship, indicating that as one variable decreases, the other decreases. A coefficient near 0 suggests a weak or no linear relationship between the variables.



Based on the correlation matrix, we can see that our target “price” has a strong relationship with many of our remaining features. However, the variables outside our target exhibit multicollinearity, such as City MPG - Highway MPG, year - one owner, and Engine features - Engine Type, etc. We removed the features that display unusually high correlations with other features while keeping the features that have a strong relationship with our target.

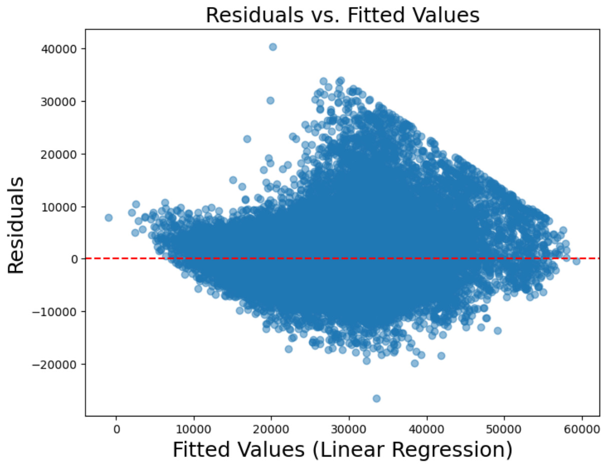


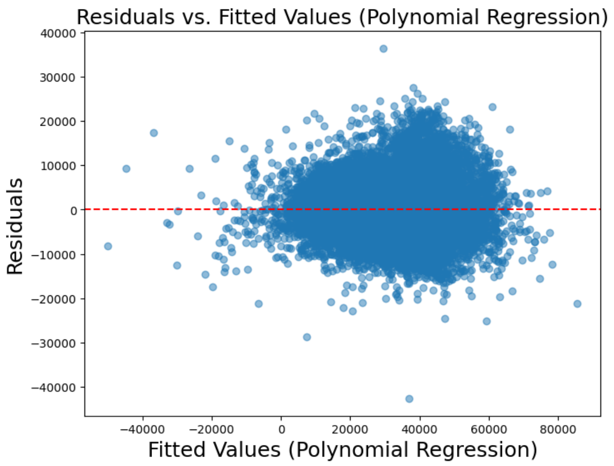
These are *the important features* that we will use to fit our models:
Year, mileage, one_owner, driver_reviews_num, highway MPG, manufacturer, transmission, drivetrain, fuel_type, exterior_color, and **engine displacement (L).**

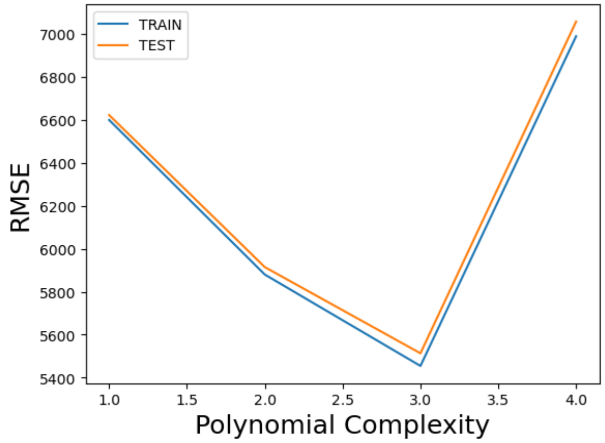
4. **Data Splitting:**
   Our last step before creating our models was to create suitable training and testing sets for our models. For this purpose, we used scikit-learn’s train_test_split to divide our dataset into 70% training data and 30% test data, giving us 55,470 training samples and 23,774 test samples.

5. **Model Evaluation and Predictions:**

   a. *Multiple Linear Regression*: We used the top 10 significant features for modeling but found Simple Linear Regression unsuitable due to a violation of linearity assumptions in the residual plot. Consequently, we tried for Polynomial Regression. Analysis revealed underfitting in Simple Linear Regression and overfitting at degree 4 in polynomial regression. Optimal performance is achieved with cubic polynomial regression (degree 3), supported by the residual plot showing non-constant variance.







b. *Decision Tree*: A decision tree regressor with an absolute error criterion and a maximum depth of 4 is subjected to cost complexity pruning. Cross-validation with 5 folds and grid search over the cost-complexity alpha is performed to determine the optimal pruning parameter, aiming to find the model with the best trade-off between complexity and performance measured by negative mean absolute error.

c. *Random Forest*: Perform cross-validation with 5 fold and grid search with multiple parameters (such as the number of estimators, max depth, cost complexity pruning alpha, etc.) to find the best estimator that can be used to predict with better accuracy and reduce the risk of overfitting.

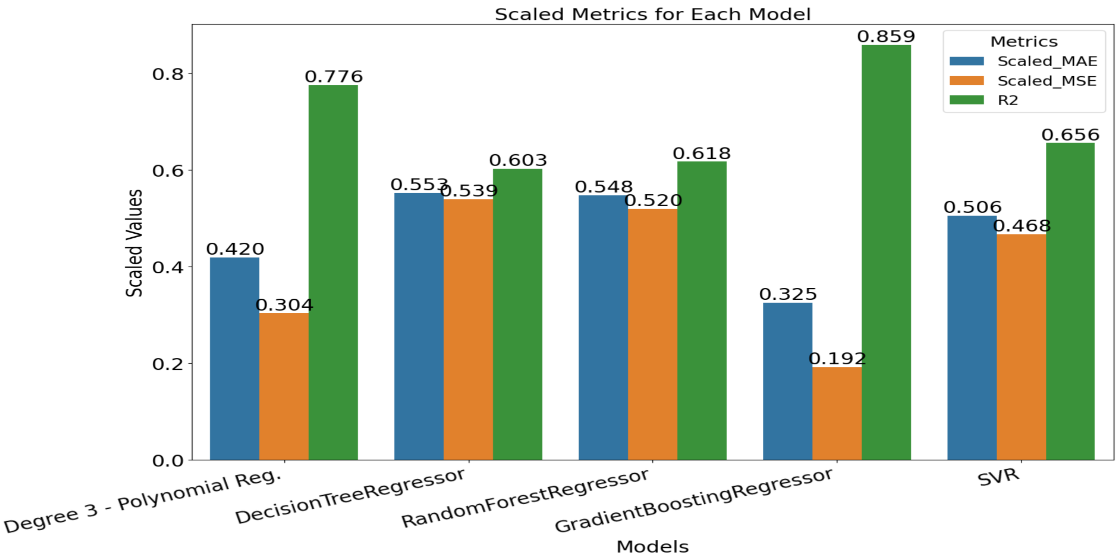
d. *Gradient Boosting*: The implementation is similar to Random Forest. This method offers the advantage of meticulous and thorough data assessment, allowing us to set a learning rate to control the learning process and enhance the model's performance. It stands out as the most optimal approach for mitigating overfitting.

e. *SVM*: We standardized the data before fitting the SVM model. As with all of our other models, we conducted 5-fold cross-validation and utilized grid search to identify the estimator that provides the optimal hyperplane for each specific kernel with multiple parameters. This approach is appropriate to deal with high-dimensional feature space.

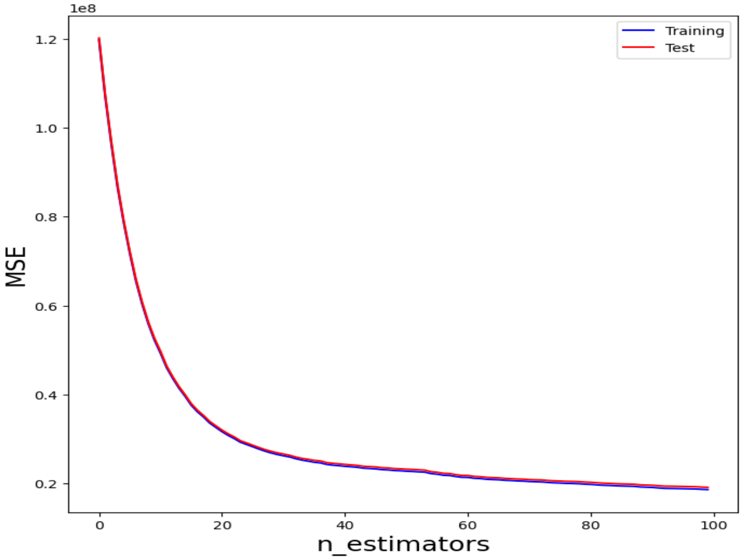
V. Experimental results and discussions

Our study focuses on predicting car prices through machine learning models. In metric evaluation, the lower **MAE** or **MSE** is considered better in terms of accuracy and higher values of **R^2** are desirable. We conclude from this that **Gradient Boosting** is the best car price predictor. **Multiple**

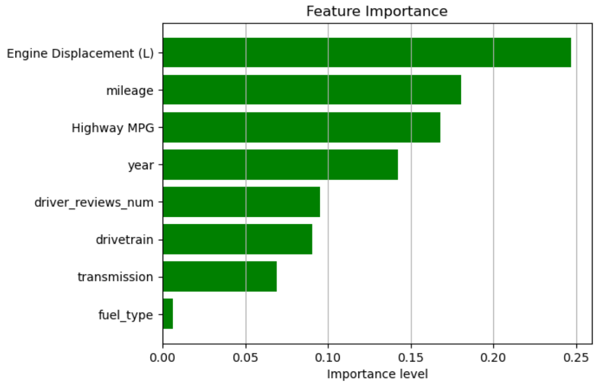
Polynomial Regression also provides reasonable metrics. Therefore, we conclude that Gradient Boosting Regressor and Multiple Polynomial Regression are most suitable for predicting the prices of car sales provided the features utilized in our model.

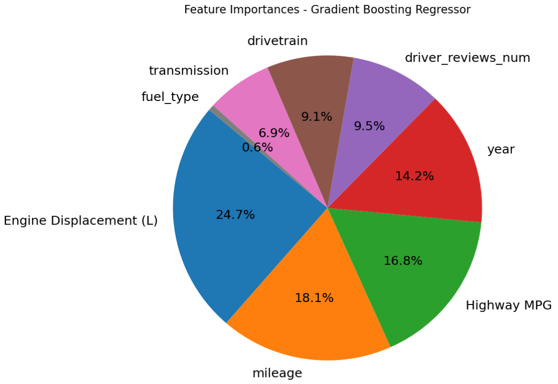


We observe that both the training and test error curves overlap and decrease as the number of estimators increases. This suggests that the Gradient Boosting model generalizes effectively to unseen data, particularly when dealing with a large and diverse dataset. The Gradient Boosting Regressor attains the highest R2 value of 0.859 and achieves the lowest MSE and MAE values of 0.192 and 0.325, respectively. This could be considered the best method for car price prediction.



By employing the Gradient Boosting Regressor, significant factors impacting car prices encompass Engine Displacement, Mileage, Highway MPG, Year, Driver Reviews Number, Drivetrain, and Transmission. Among these, Engine Displacement, Mileage, Highway MPG, and Year collectively contribute to about 74% of the car price, with Engine alone contributing about 25%.





VI. Conclusion and future research directions:

From the analysis of the results of our models, we identified that Gradient Boosting Regression served as the best model for predicting the sales price of used cars, with an R^2 score of 0.859. As this was the most effective model in predicting prices we looked at the features it determined to be most important and identified certain features that seemed to be the most important in determining the price of cars. We determined that engine displacement, mileage, miles per gallon during highway driving, and the year the car was produced were particularly important in determining the price of cars. Engine displacement was especially important in determining the price of cars in our model, with a feature importance of nearly 25%. Additionally, through both exploratory data analysis and the results of our models we determined a significant increase in the price of cars year-over-year during the 2010s, increasing in the aftermath of the COVID-19 pandemic. Having achieved a suitably high R^2 score with both our Multiple Linear Regression and Gradient Boosting Regression models we conclude that machine learning regression models are suitable for predicting the price of used cars, with gradient Boosting Regression being the most suitable.

VII. Future Work:

For future research into this topic, we would like to attempt to generalize our models to various datasets of car sales, and potentially analyze whether certain models are more suited to training sets made of samples from specific car manufacturers or models. By breaking datasets up by car manufacturer or model we hope to be able to better explore how the differences in features within certain lines of cars affect the price of those cars, and whether models trained to specific types of cars perform better in the aggregate than the generalized models described in this report. Additionally, we would like to explore in further detail the accuracy of gradient-boosting regression models in datasets prepared specifically for that type of model, and the optimal depth of the trees within the model to be able to best generalize our results.

VI. References:

Datasets:
https://www.kaggle.com/datasets/andreinovikov/used-cars-dataset

Literature Review:

[1] S. Pudaruth, "Predicting the Price of Used Cars using Machine Learning Techniques," International Journal of Information & Computation Technology, vol. 4, no. 7, pp. 753–764, 2014.
[2] C. Chen, L. Hao, C. Xu; Comparative analysis of used car price evaluation models. *AIP Conf. Proc.* 8 May 2017.
[3] N. Monburinon, P. Chertchom, T. Kaewkiriya, S. Rungpheung, S. Buya, P. Boonpou; "Prediction of prices for used car by using regression models." 2018 5th International Conference on Business and Industrial Research (ICBIR), pp. 115-119. IEEE, 2018.
[4] K. Noor and S. Jan, "Vehicle Price Prediction System using Machine Learning Techniques," International Journal of Computer Applications, vol. 167, no. 9, pp. 27–31, 2017.