Used Car Price Predictor Report

Overview:

Data Science Problem:

In this project, we will be predicting the price of a car based on a dataset of 42088 rows of car information. With rapid changes occuring in the auto industry, it is important for both dealerships and potential buyers to get fair estimates when determining the value of used cars. Every year, more electric vehicles are purchased, new auto manufacturers are making a name for themselves, and government regulation is affecting the cost to own different types of vehicles. With these factors considered, there has been a lot of volatility in the auto industry over the last 5 years. This means that it is ever more important to have up-to-date data and models that would allow the buyer and seller to properly price a car based on a variety of factors that are readily available such as age, mileage, make, location, and more.

Scope and Limitations:

There was a significant amount of time allotted for the creation and fine-tuning of this project so time was not a notable constraint, however, there still was not enough time to go in depth to create a highly complex solution to this problem. In addition, the resources at our disposal would have made it very challenging to consider a complex solution due to the relatively poor computational capacity of our computers and the size of the dataset. Despite this, we were still able to create adequate models that achieved what we hoped for given the resources at our disposal.

Project Objectives and Expectations:

The goal of this project is to create a model that can predict the price of a car reasonably well so it can be useful in a real setting. It is important to be realistic with our expectations since the used car market is inconsistent and depends on so many factors that are both known and unknown. After taking this into consideration, we have decided that our goal is to have a model that can predict the price at a reasonably high rate so that the prediction is within 20% of the actual predicted values.

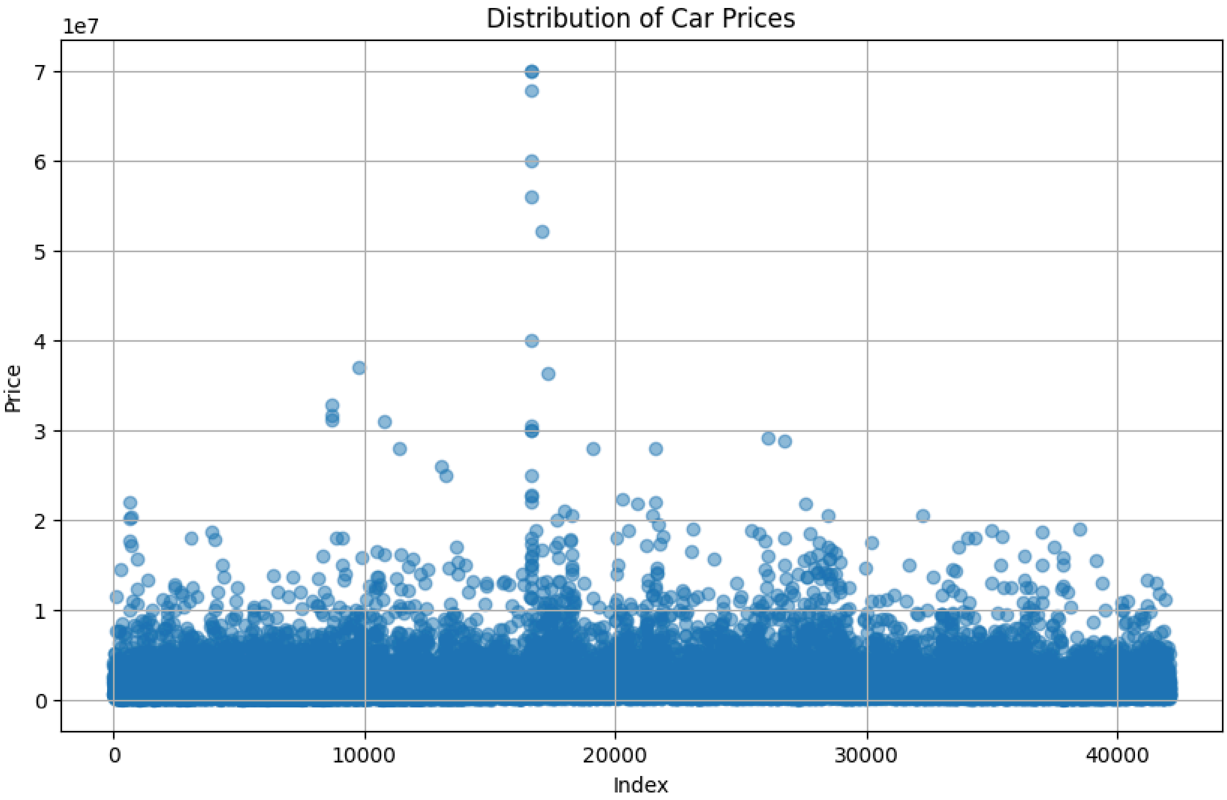
Project Structure:

The project consists of the following sections: data collection, data exploration, data preparation, model training, and lastly the results and visualizations of them. The reasoning behind this structure is that our goal was to get an initial understanding of the dataset through data exploration, then moved on to clean the data and select features, and lastly strived to create models and evaluate them using metrics and visualizations.

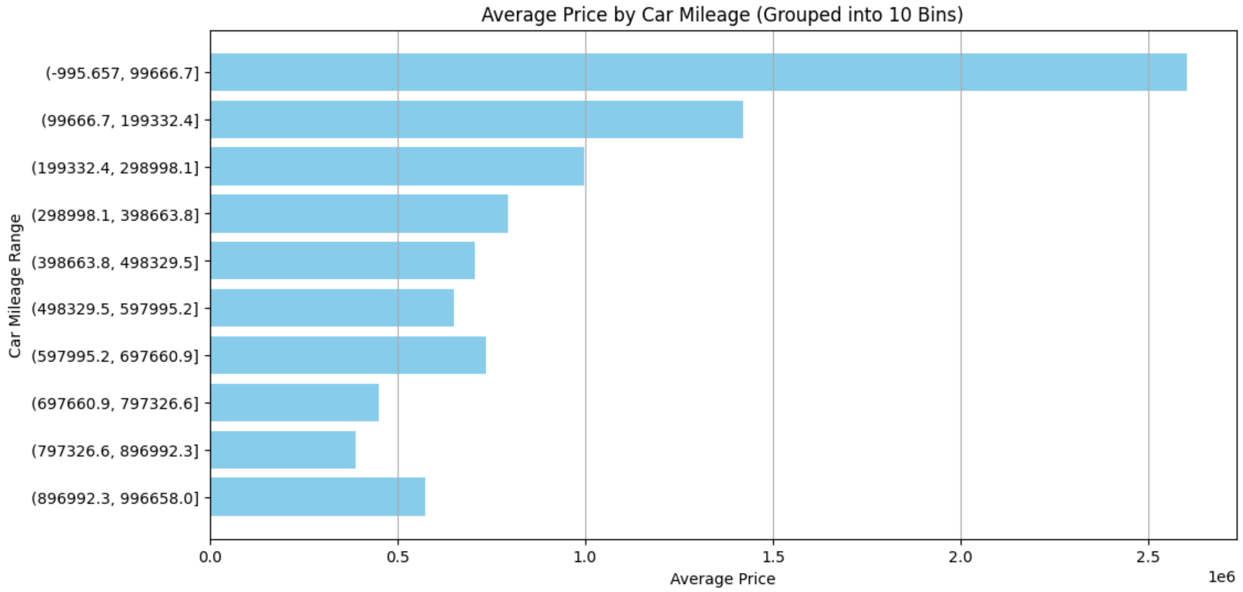
Project Summary

Data Exploration:

We loaded in the csv file using pandas and looked at the head of the dataframe to see if there was anything noticeable upon first glance. The data seemed to be organized very well with adequate values in each column given the context of the car dataset that we have. The summary statistics give us a brief overview of the count, mean, standard deviation, etc. of the numerical data in this dataframe. We then look at the null counts and are pleasantly surprised to see that there are no null values in this dataset. This is where the surface level exploration ends and we begin to perform more advanced techniques to get a better understanding of the data that we are working with. We counted the number of unique values in each categorical attribute and found that some features had a long list of unique attributes while others had only a few. This information is important because it is much easier to work with the shorter list because they can be more easily converted into numerical data later on during model building by using techniques such as one-hot-encoding. We moved on to create a variety of visualizations to see if we could draw any conclusions from the information that we have. We made a distribution of car prices which showed how the majority of the cars fall into the cheap to moderate price range but there are some potential outliers that are much more expensive. This logically makes sense in the context of the car market because most cars are made for the average person but there are some extraordinarily expensive cars that are made for the rich.



We also created visuals for average price vs transmission type, average price vs fuel type, and average price vs drive type. These visuals allow us to see which of these categorical values tends to be more expensive in comparison to the others. Each gave some insight and had potential to be used as features in the models later on. We also looked at average height in comparison to horsepower and average price in comparison to mileage. Mileage is an obvious important indicator of car price because it does well in showing us what kind of shape the car is in. To visualize this we had to create bins for 10 different ranges of miles that encapsulated the entire dataset.



It is also important to note that the lower car mileages occur substantially more frequently in comparison to the higher car mileages.

Data Preparation:

We began by removing the unnamed column, country column, and city column since these likely will not be impactful on price. We then applied one hot encoding for the car_fuel and car_transmission columns to transform these categorical variables into numerical ones that can be used in the models. To handle car make and model, we implemented target encoding. Other encoding methods like one hot encoding or binary encoding did not seem feasible because of the large number of unique brands and models. Target encoding involves replacing the categorical value with the mean target value for each respective category. This allowed us to preserve the relationship between car brand/model and price without having to extrapolate the data.

To prevent data leakage, we split the data into training and test sets before applying target encoding. After these steps, we now have all numerical attributes so we are able to create a correlation matrix with all the features. There is some multicollinearity present but for the most part the features seem independent of each other so it is safe to move on with them for model training.

Model Creation:

Given our goal to build models which can predict the price of used cars, we executed one hot encoding for the categorical feature which exhibited manageable quantities of distinct value classes. For car brand and car model we employed target encoding, a method which utilizes the mean target variable magnitude pertaining to each specific class and applying this as the measure for each instance depending on which class they have. Albeit potentially losing some natural distribution due to reliance on mean, this technique effectively allows for a categorical feature to be converted to numerical in a single dimension. The data was then split into training and test sets for both features and the target, car price, before standardizing the feature sets to mitigate the potential impact of notable magnitude discrepancies regarding the manner in which features are measured.

When considering which models would be suitable given the task at hand, it was essential to ponder the essence of specific algorithms in an effort to employ those which align with our dataset and goals. From such a consideration, we came away with the intent on utilizing the following models: Linear Regression, Decision Tree Regression, Random Forest Regression, and Gradient Boosting Regression.

We start with linear regression as a conventional means of initiating modeling phases for prediction contexts as this model is quite straightforward and offers rapid computational speed. Given the potential for notable extenuating factors influencing price and in turn eliciting non-linearity, we did not expect good performance from linear regression yet we maintained its usage for the reasons above.

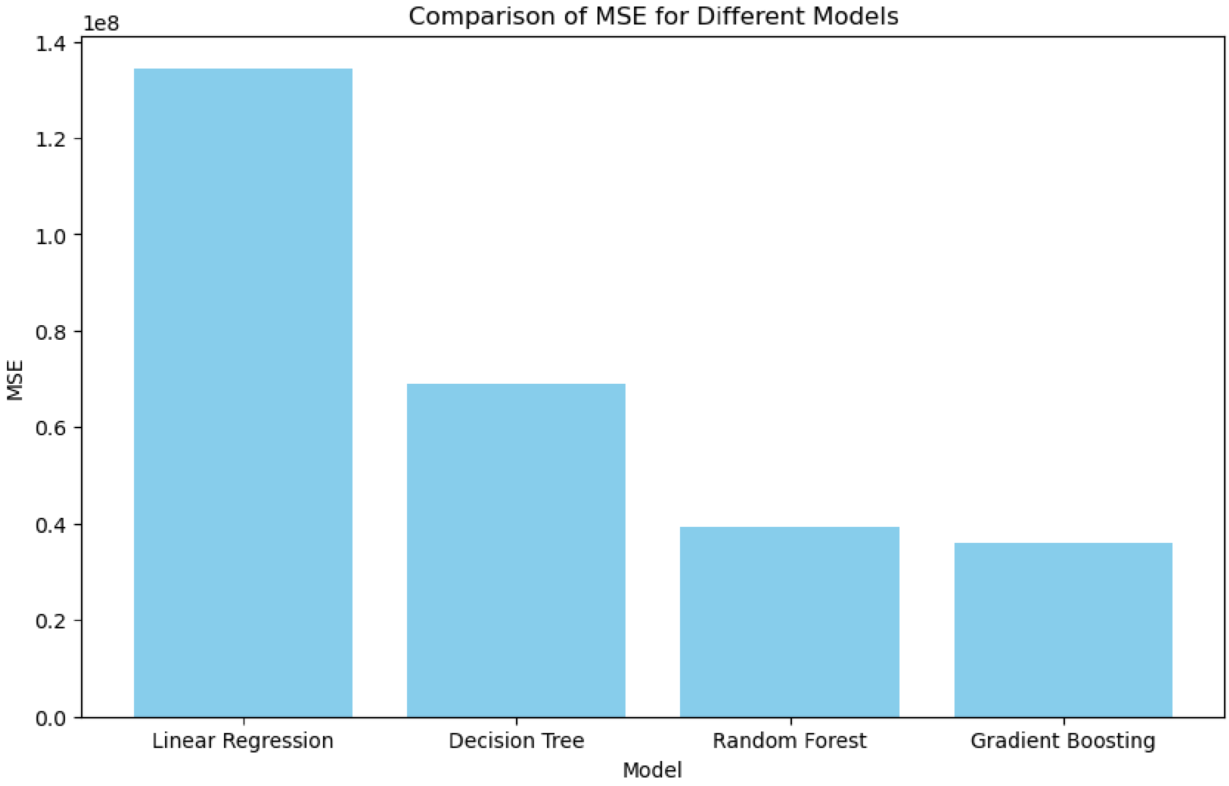
Following this we pivot to an adaptation of conventional classification based decision tree, Decision Tree Regression. This method harnesses the decision tree algorithm however, with the alteration which results in continuous values being produced as the measure of the target which each feature corresponds to given the value which an instance exhibits for said feature. This is effectively done by employing some balanced measure such as mean or median to attribute the manner in which this feature essentially corresponds to a continuous target variable. In our context, we employ sklearn decision tree regressor which utilizes mean for this. This process is conducted analogous to conventional decision trees with regards to the traversing about a tree where splits are conducted by certain features with a culminating predicted continuous value for the target the ultimate result. We employed this method with the intention of viewing how it stacks up in comparison (as the basic unit to) to ensemble versions which are more powerful and exhibit better performance in most contexts.

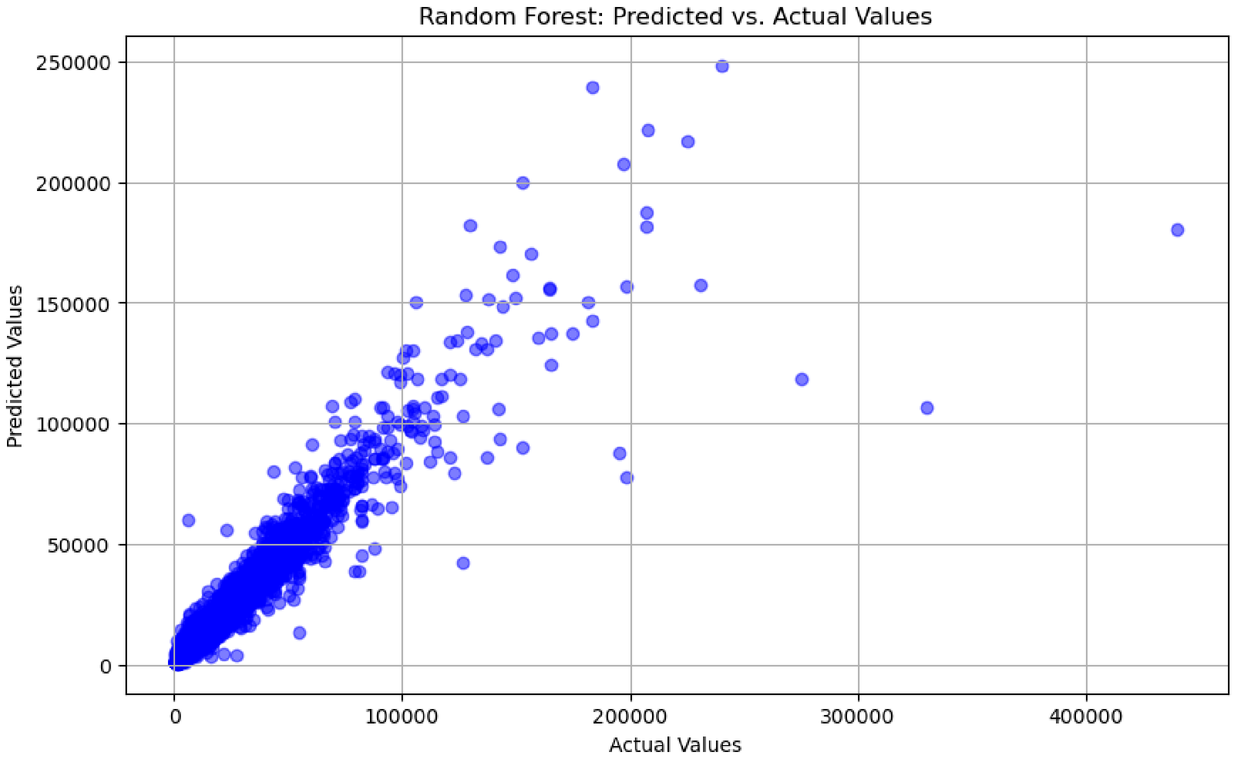
Subsequently, we progressed to Random Forest regression which is a bagging ensemble of decision trees. This technique produces a robust model as it performs random sampling with replacement on the dataset to train a multitude of decision trees in these different subsets of the data. Following this, each decision tree in the now formed forest conducts the decision tree algorithm as discussed above with each producing a predicted continuous target value. Following this, the results are aggregated which could be theoretically conducted by unique metrics however the typical method is via the mean predicted value. As a result of using sklearns randomforestregressor, this aggregation for our model is facilitate via the mean. Focusing on our specific implementation of this algorithm, we initiate the forest with 100 trees as a customary initialized value and a random state of 42 to provide the random seed for the sampling of the subsets for each tree. The specification of a specific random seed allows for this model to be reproduced since the manner in which the randomization occurs is a set value.

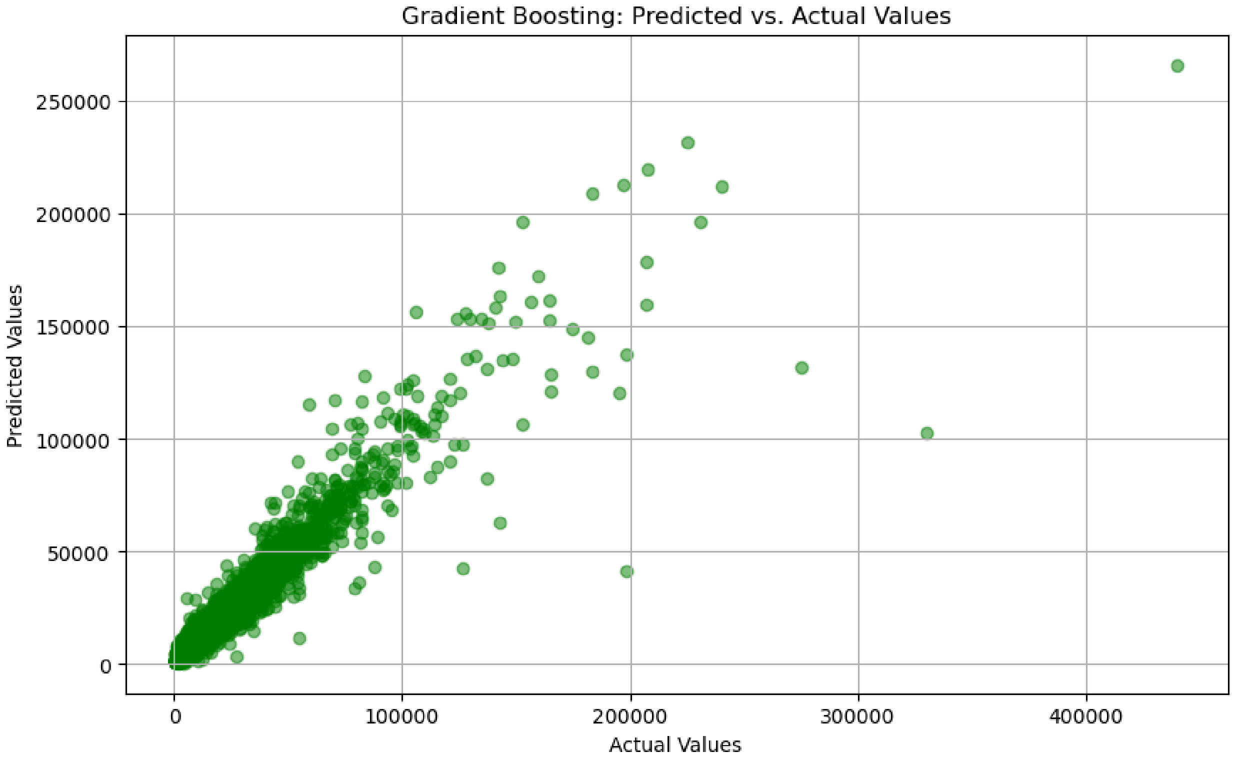
After constructing our Random Forest Regression model, we proceed to execute our final model, Gradient Boosting Regression. This is another enhancement upon decision trees which employs them as base units in an ensemble. Albeit another decision tree ensemble, this is quite different from Random Forest Regression as this method employs boosting as opposed to bagging. Boosting differs from bagging in the sense that it is a sequential training procedure which trains a model, optimizes parameters via gradient descent, and substantially employs the residuals of this first model to train another model with specific attention to said residuals. This model is then optimized via gradient descent as well and the process iterates. This effectively results in a multitude of decision trees, each training on the priors residuals, a method which converts what would eb weak learners to a collective strong learner as they make up for each other's struggles. Various methods can be employed to provide a means of terminating this interactive procedure such as a specified amount of trees, a threshold loss function value, or other context specific criteria. For our usage, we employ a grid search hyperparameter tuning which goes beyond solely looking for termination criteria by broadening the discussion to optimization overall. For this we consider the number of trees, the depth of each tree, and the learning rate to search for values of each which result in the optimally performing model.

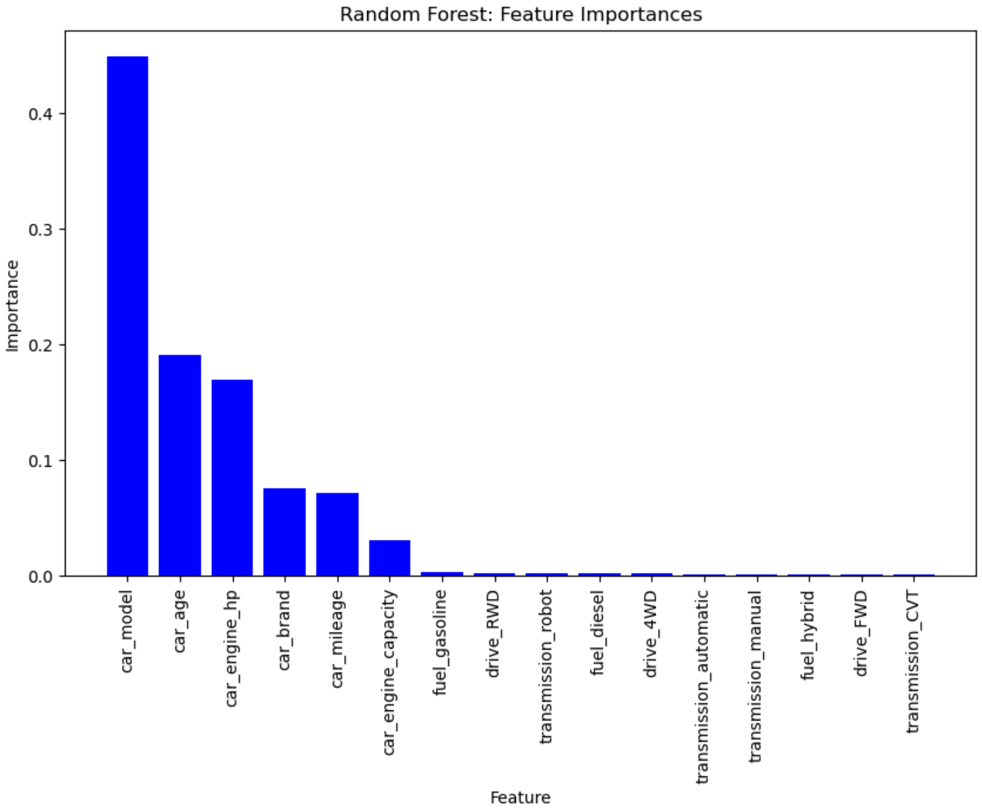
Following the implementations of these models, we consolidated the results via several methods with a discussion and portrayal of said results below.

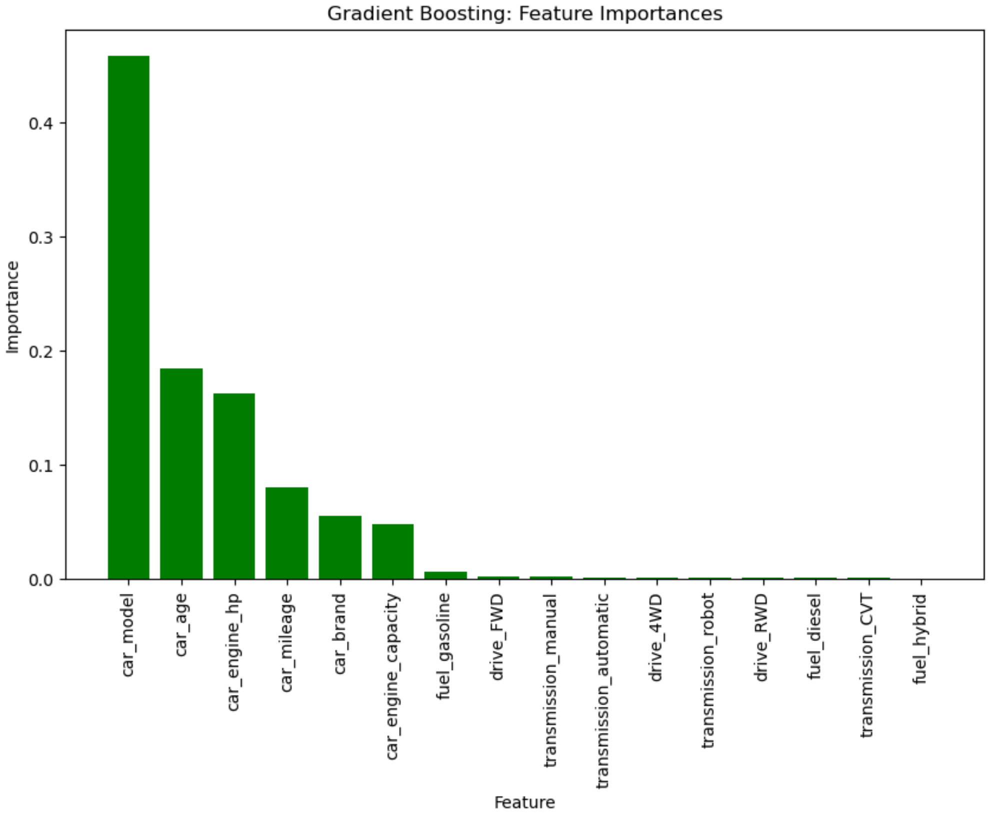
Results:


Comparison of MSE for Different Models
1e8
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
MSE
Linear Regression
Decision Tree
Random Forest
Gradient Boosting
Model


Random Forest: Predicted vs. Actual Values
Predicted Values
Actual Values
250000
200000
150000
100000
50000
0
0
100000
200000
300000
400000


Gradient Boosting: Predicted vs. Actual Values
Predicted Values
Actual Values
250000
200000
150000
100000
50000
0
0
100000
200000
300000
400000


Random Forest: Feature Importances
Importance
0.0
0.1
0.2
0.3
0.4
Feature
car_model
car_age
car_engine_hp
car_brand
car_mileage
car_engine_capacity
fuel_gasoline
drive_RWD
transmission_robot
fuel_diesel
drive_4WD
transmission_automatic
transmission_manual
fuel_hybrid
drive_FWD
transmission_CVT


Gradient Boosting: Feature Importances
Importance
0.0
0.1
0.2
0.3
0.4
Feature
car_model
car_age
car_engine_hp
car_mileage
car_brand
car_engine_capacity
fuel_gasoline
drive_FWD
transmission_manual
transmission_automatic
drive_4WD
transmission_robot
drive_RWD
fuel_diesel
transmission_CVT
fuel_hybrid

Based on the mean-square error and other metrics, the gradient boosting and random forest algorithms are the most accurate, while decision tree and linear regression lag behind. Due to the gradient boosting's ability to transform weak learning nodes into strong learning nodes, it is not surprising that the gradient boosting models outperformed the others. This model and the random forest do a good job balancing complexity with contingencies that prevent overfitting as discussed earlier. When looking at their prediction vs. the actual values, it can be seen that the models generally predicted in the corrected area for each car but there were some cases where they made mistakes. These graphs interestingly show significant similarity to each other but still display some key differences. For example, when looking at the last outlier on the right, the gradient boosted model predicts it much better than the random forest. These two models also had very similar feature importance to one another, the only notable difference being the car mileage and car brand were flipped. This means that the car mileage feature was more important for the gradient boost model while the car brand feature was more important for the random forest model. As an additional note, the values for the MSE are objectively high due to the nature of the target variable, price, being a large continuous variable. As a result of this, we computed the mean absolute percentage error which is an adaptation of MAE to specifically focus on the percent which the error from predicted to actual represents of the magnitude of the actual. This simply provides context to these metrics from a standpoint which avoids the surface level influence of the enormous continuous values of our target variable.

Conclusion

There was substantial disparity in the performance of the models depending on which ones are being looked at. The linear regression performed poorly which is expected since this model is very simple and was likely underfitting the data. The gradient boosted decision tree and the random forest performed the best because they are relatively complex models that also have contingencies that prevent overfitting (unlike a normal decision tree). These models had substantial similarities upon further analysis and this is likely due to their usage of decision trees as the basis for their algorithms. These models could be useful for used car dealerships and for individuals looking to buy a used car since it would allow them to get a general idea of what the car is worth. That being said, it is worth noting how the used car market is one of high complexity with a substantial amount of extenuating factors which can influence the price. The nature of this multifaceted market renders the ability to produce models with objectively impressive performances difficult which is why it is essential to consider this context when evaluating the performance of models to predict used car price. Overall, the ensemble tree methods performed the best as noted above with gradient boosted regression slighting edging out Random Forest Regression in MSE. For future research, some areas of interest for us would involve producing comparable models however using data sourced from another country to compare against the models built on this Russian data. This would allow us to potentially note the circumstantiality regarding the models viability between areas outside of the nation which the data pertains to.