Machine Learning Application in Predicting Price of Used Cars

Ramesh Mainali[1]

[1]Catholic University of America

mainali@cua.edu

Abstract. *Predicting the value of used cars has several applications for both customers and retailers. The main goal of this project is to use publicly available data sets to train machine learning models which can ultimately be applied to predict the price of used cars. I gathered recent data sets of used cars comprising of price and several factors that were listed for sale in the United States. Next, I performed a detailed data cleaning removing features that are irrelevant to estimate the car price. Additionally, I analyzed the data to find the trend in the data sets. Finally, I developed and compared six supervised machine learning models that can be used to predict the price of used cars. Among the six regression models, I found extreme gradient boosting (XGBoost) regressor provides the best prediction with an R-squared score of 0.91.*

1. Introduction

The used car market has grown tremendously over the past decade. The global used car market size has been valued at $ 1332 billion in 2019, and it is expected to reach $ 2140 billion by 2027 [1]. Due to the current pandemic situation, the demand for the used cars is expected to rise further. Income generation of people has been curbed, which leads customers to become very careful while investing a large sum. This has lead people to reconsider their new car purchase plan. As such, reliable used car comes as a great alternative to such customers provided that they find a good deal. However, as opposed to a new car whose price is mostly fixed by the manufacturer, identifying a good value for a used car depends on several factors. Some of the major factors influencing the price may include car make and model, age of cars, condition of used car as well as location. Hence, developing a model to predict a reliable price will benefit customers who are looking for a good deal at the used car price.

The goal of this project is to build a supervised machine learning model to predict the price of used cars based on various listed features. This document is organized as follows. In §2 I will discuss the datasets used in this project. Then I will explore the datasets to investigate various trends in §3. In §4, I will present various techniques used to prepare data to use as machine learning model inputs. In the next section (§5), I will discuss different machine learning models considered to pick the best model. Finally, in §6 I will present results and discuss it's various implications

2. Datasets

In this project, I use a data set of used cars that are publicly available on the Kaggle website [2]. The data is generated directly by scrapping the craigslist sites in several US regions in 50 different states including Washington DC. The data set is scrapped every

few months, as such they are mostly very recently listed data on craigslist. From the craigslist sites, the author scrapped the main key features from the car listing such as price, odometer, year of the model, make, model, etc. The data is available in .csv format which includes 458213 rows and 26 columns.

The selection of the dataset is motivated primarily by two main factors. Firstly, the data set represents raw data since it is directly scrapped from recent craigslist listings from all over the US. They are mostly uncleaned data having both categorical and numerical features as well as several outliers and missing values. This kind of dataset presents a good opportunity for data cleaning. Secondly, the data set is fairly new since they are recently scrapped from the internet which is very important to predict current market price value.

For data cleaning and analysis, we use the following python tools:

- **Data Wrangling**: pandas, numpy
- **Machine Learning:** Scikit-learn
- **Data Visualization:** matplotlib, seaborn

In the initial stage of data cleaning, we first removed 14 features (columns) that are likely irrelevant to predict car value. This include the following features:

- Unnamed:0 –> This basically is equivalent to data frame index, and doesn't provide any price predicting value.
- ID –> Unique identification number which doesn't play any role in car price.
- url –> Url of craigslist add.
- region –> This represent specific region (area) where the craigslist add was listed. Although this may likely affect car price, I decide to only include a broader location identification based on the state.
- region url –> Nothing to do with car price.
- cylinders –> This may slightly affect price of car, but this information is likely already incorporated in car model.
- VIN –> Unique identification number of car. Nothing to do with car price.
- size –> This information is encoded in car body type.
- paint color –> This feature is likely more important for a new car. Since our data set already has large feature, I decide to remove this for further analysis.
- image_url –> Nothing to do with car price.
- description –> Description of car may encode some useful information. But this has to deal with natural language processing which is beyond scope of current project.
- lat –> latitude of location where the car is listed. This gives location information for which we will only use state information.
- long –> similar to latitude. _date –> The data are recently scrapped from the internet. So we can extract information on car age directly from model year.

2.1. Outliers Removal:

After dropping irrelevant features from the datasets, the next step is to identify any outliers in the dataset. Here, I used descriptive statistics to check the numerical values of different features (price, year, and odometer). First, I found that the model year of the car ranges from as old as 1900 to as new as 2021. The oldest model cars are either wrongly entered into the datasets or are extremely rare vintage cars. I looked into the median price

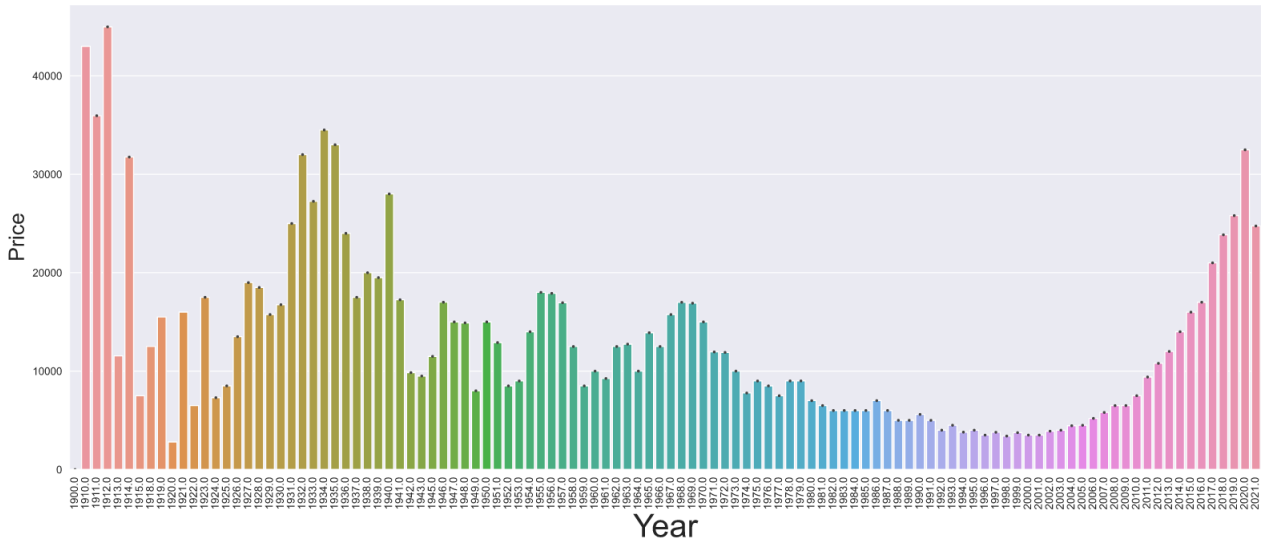

Figure 1. Distribution of median car price according to year. The chart shows that the median price of the oldest car (>30 years) old are more expensive than 15-25 years old cars.

distribution of cars according to model years which showed surprising results. The median price of the oldest cars (>25 years old) is actually higher than the 15-25 years old car (Figure 1). This supports my second hypothesis that the oldest car is likely vintage cars. Those types of cars are likely to be sold more in auctions rather than craigslist, so I removed any cars that are older than 25 years from the datasets.

Next, I found that the listed price of cars ranges from 0 to 3.6 billion. The price of zero likely means the seller is not willing to reveal the price in their listing, whereas the highest prices are likely entered wrongly in the craigslist listing. In order to remove those outliers, I used a method based on the interquartile range. In this method, we identify outliers as any value that are above 1.5 times of interquartile range (IQR) than the third quartile (Q_3) or any value below 1.5 times of interquartile range (IQR) than the first quartile (Q_1). This method removed the highest car values from the datasets but there were still several cars with a price of 0. So, I further use a low price threshold of 1000 dollars in order to remove any outliers at the car price.

In the last step, I removed any cars whose odometer reading showed more than 500000 miles. This removed several outliers some of whose odometer was listed as high as 2 billion miles, which is practically not possible.

2.2. Missing Values:

After outliers removal, there were no any missing values in the numerical features like price, year and odometer. However, there were many missing values in each categorical features except the state. Out of those features, condition, drive and body type had most missing values (37%, 25% and 19%, respectively). Since some of these information like condition, drive and body type might be encoded in model year and car model, I simply replaced those missing values with "unknown" category. For the remaining features, the missing values were fairly low (at most 2.5%) which were dropped from the datasets for further analysis.

Table 1. Data frame table showing relevant features in predicting car price after dropping irrelevant features.

	price	year	manufacturer	model	condition	fuel	odometer	title_status	transmission	drive	type	state
394893	4500	2008.000000	hyundai	accent	like new	gas	93000.000000	clean	automatic	fwd	hatchback	ut
382269	22900	2015.000000	mercedes-benz	benz ml350	like new	gas	69700.000000	clean	automatic	fwd	SUV	tx
218221	9995	2014.000000	nissan	pathfinder	unknown	other	168419.000000	clean	automatic	unknown	unknown	ms
364231	7959	2009.000000	ford	mustang	unknown	gas	103901.000000	clean	automatic	rwd	coupe	tn
213380	7899	2013.000000	toyota	camry se	excellent	gas	120236.000000	clean	automatic	unknown	sedan	mn
352073	9399	2011.000000	acura	mdx	like new	gas	139934.000000	clean	automatic	4wd	SUV	sc
303926	5295	2009.000000	ford	edge limited	excellent	gas	189000.000000	clean	automatic	fwd	SUV	oh
211113	22490	2017.000000	nissan	rogue	unknown	gas	33518.000000	clean	automatic	unknown	unknown	mn
285756	6800	2013.000000	mazda	5 touring	good	gas	103760.000000	clean	automatic	unknown	mini-van	ny
426286	11995	2011.000000	cadillac	cts coupe awd	excellent	gas	99186.000000	clean	automatic	4wd	coupe	wi

3. Exploratory Data Analysis

After outliers removal, there were no missing values in the numerical features like price, year, and odometer. However, there were many missing values in each categorical features except the state. Out of those features, condition, drive, and body type had the most missing values (37%, 25%, and 19%, respectively). Since some of this information like condition, drive, and body type might be encoded in the model year and car model, I simply replaced those missing values with the "unknown" category. For the remaining features, the missing values were fairly low (at most 2.5%) which were dropped from the datasets for further analysis.

I first explored the most type of cars available for sale on craigslist. For this, I first grouped the data by the different manufacturers which revealed that ford, Chevrolet, Toyota, Honda, and Nissan are the top five most common cars available for sale on the craigslist site.

Similarly, I found that f-150, Silverado-1500, 1500, Camry, and Altima are top-five car models listed on craigslist site. This is shown in Figure 2 where I plot the word cloud for the most common car manufacturer and car model in the datasets.

In the next step, I determined the distribution of car prices. As shown in figure 3, I estimated how car price is related to car body type. It appears that the most expensive

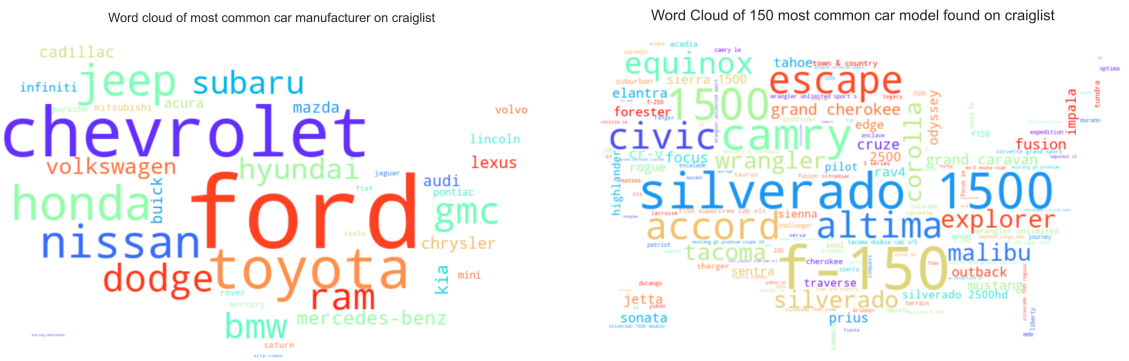

Figure 2. Word Cloud showing the most common manufacturer of used car(left) and the most common car model (right).

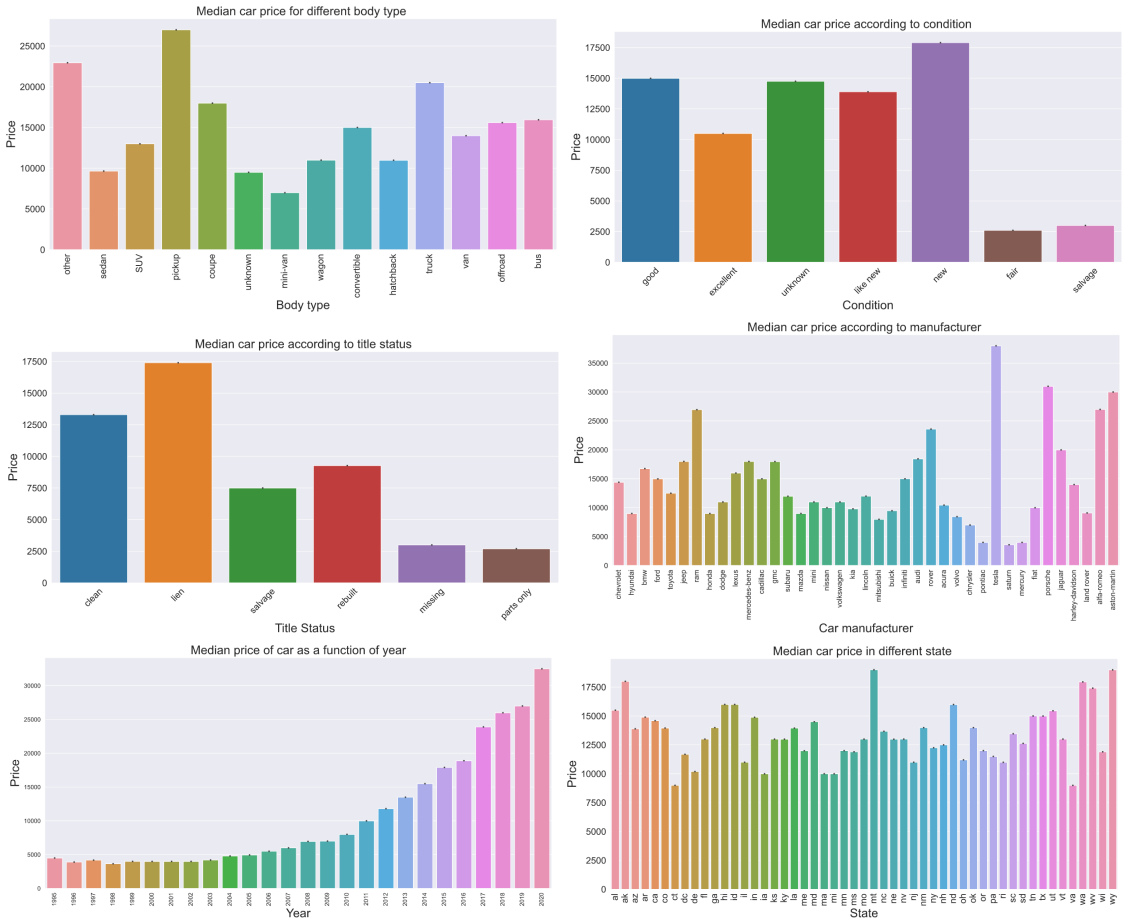

Figure 3. Word Cloud showing the most common manufacturer of used car(left) and the most common car model (right).

car type is pickup cars, while the sedan and minivans are the least expensive in the group. In terms of car condition, the cars in new and good condition are found to be the most expensive cars, while cars in fair and salvage condition are least expensive cars. Next, I found that the most expensive cars are those having "lien" title status. This is because these cars are relatively new cars (less than 5 years) since customers are yet to pay off their car loan from the initial purchase. Not surprisingly, cars with missing titles and those up for sale only for body parts are evaluated to be the least expensive vehicles.

In the next category, I explored the car manufacturers and found that the median car price from luxury car makers is listed at the highest price. For example, the top three most expensive vehicles come from Tesla, Aston-martin, and Porsche which are all well knows as luxury carmakers. I then noticed that the price of the oldest car in our datasets (after initial cleaning) is listed at the lowest price which is in agreement with our expectation. Finally, I investigated the median car price in different states. The top three states with the highest car price were found to be Montana, Wyoming, and Arkansas while the lowest car prices were found in Virginia, Connecticut, and Massachusetts.

For the numerical features, I evaluated correlation among the features where I found that the price is positively correlated with the model year and negatively correlated with the odometer reading. This suggests that the new car models and those with the lowest odometer reading are the most expensive cars found on the craigslist site.

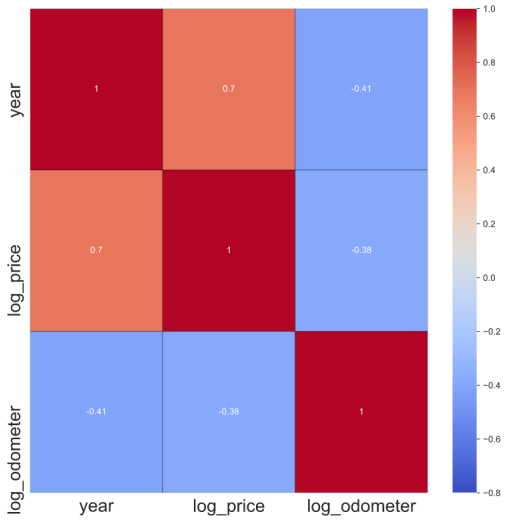

Figure 4. Plot showing correlation map of data set.

4. Data pre-processing:

Now, the data is cleaned, we need to further prepare the data to feed into machine learning algorithms. This procedure involves the following steps:

4.1. Data scaling:

For various machine learning algorithms (particularly those based on distance evaluation), scaling of input data is a crucial first step. This will bring the values of independent features to a common range so that no variable is dominated. By following this step, the machine learning models will learn to not assign larger weightage to higher value features and vice versa. There is two main types of data scaling: Normalization and Standardization. For this work, I used the standardization procedure which changes the input data into output with a mean of 0 and variance (standard deviation) equal to 1.

4.2. One Hot Encoding:

Since our data sets have several categorical features, we need to convert them into numerical features since many machine learning algorithms cannot handle categorical features. Here, I used the One Hot Encoding method to change categorical input features to binary numbers. Each 1 or 0 will then represent the presence or the absence of a particular categorical feature. The advantage of this method is it directly converts categorical datasets into numerical values. However, if the features have several categories, this will result in several new column features.

4.3. Dimensionality Reduction:

Since One Hot Encoding introduces several additional columns in the datasets, I decided to use a dimensionality reduction algorithm to reduce the number of features in the data. The goal of the dimensionality reduction is to transform a larger set of variables (features) into a smaller one while still maintaining important information from the larger set. The method I applied is called Principal Component Analysis(PCA) which computes principal vectors (components) that are linearly uncorrelated and contain most of the variance in the data. Applying this procedure reduced the features from 1057 to 259.

4.4. Training & testing sets:

Finally, I divide the input data sets into training data and testing data. This will be very useful in checking the validity of the machine learning model. For this process, I assigned 70% data sets as training data and the remaining 30% as testing data.

5. Machine Learning Models:

We are interested in predicting the car price which is a continuous variable. The solution for a continuous variable requires a regression analysis that predicts an outcome (y) based on single or multiple input variables (x). I considered six different regression models to predict the car price which are briefly described below:

- **Linear Regression:** This is a simple baseline model which follows a linear mathematical model to determine a dependent variable (output) from a set of the independent variable (inputs). For an input set of X and output of Y, linear regression takes the form of:

$$Y = mX + c, \tag{1}$$

  where m is the slope and c is an intercept. This model is easy to understand and the most efficient to execute. However, it is also prone to underfitting and sensitive to outliers in input data.

- **K Neighbors Regression:** This is another simple algorithm which makes a prediction based on a weighted average of K nearest data points, where K is the number of nearby data points that are used to make a prediction for a given input.

- **Support Vector Regression:** Support Vector Regression(SVR) is a type of regression that supports both linear and non-linear regression which is based on Support Vector Machine (SVM). The objective of SVR is to calculate a linear regression function in a multidimensional feature space which is formed by input data via a nonlinear function (kernel).

- **Decision Tree Regression:**Decision Tree Regressor is a supervised machine learning model that predicts the target by learning decision rule. It builds a model by breaking down data into smaller subsets while at the same time developing an associated decision tree in an incremental fashion. The final result is in the form of a tree structure with nodes representing decision and leaves (or end nodes) representing prediction. While the algorithm is great for decoding the non-linear relationship between input and output variables, it is also highly sensitive to a small change in input features.

- **Random Forest Regression:** Random Forest is basically an ensemble of decision trees. Random Forest Regression uses multiple decision trees and a statistical technique bagging to build an ensemble machine learning model. It is an attractive option to decode the non-linear nature of features. However, it can only make predictions based on the average of previously observed variables. In another word, it can not extrapolate to make a prediction for an input variable outside of the

observed range since the range of prediction is bound by the largest and smallest labeled data. This becomes problematic when the inputs of training data differ from the one used for prediction.

- **XGBoost Regression:** XGBoost stands for extreme gradient boosting which is a tree-based algorithm. It is a sophisticated form of gradient boosting decision tree algorithm which is preferred for its performance and execution time. In the Gradient boosting approach, new models are created to predict errors of prior models which are then added together to make the final prediction. It uses a gradient descent algorithm to minimize the loss when adding new models. The algorithm provides a perfect combination of software and hardware optimization techniques yielding results in the shortest time frame using lesser computing resources.

5.1. Model Performance:

In order to determine how close the data are fitted by the regression models, I used the following two metrics:

- **R-squared (R2) :** R-squared (also known as the coefficient of determination) is computed by subtracting the ratio of the sum of squares of residuals (SS_{res}) from the regression model to the total sum of squares of errors (SS_{tot}) from the average model from 1. R-squared (R2) is given by:

$$R2 = 1 - \frac{SS_{res}}{SS_{tot}}. \tag{2}$$

  R-squared value explains the degree to which the input variables explain the variation of the predicted variable. For example, the R-squared value of 0.9 tells us that the input variable explains 90% of the variation in the predicted variables.

- **Root Mean Squared Error (RMSE):** The root mean square error (RMSE) measures the standard deviation of the errors, difference in the predicted and observed value, in the models. It is given by:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(Y_{pred} - Y_{true})^2} \tag{3}$$

6. Result & Summary

I considered six different regression models to predict the car price. The baseline multiple linear regression model did a reasonable job achieving an R-squared value of close to 0.8. On top of that, the linear regression was very easy to run as well as executed in the least amount of time. Another traditional approach, K Neighbors regression even outperforms the linear regression clocking an R-squared value of 0.86. But, it did require a fairly large time to run the model. The next model I considered, Support Vector Regression performed better than the first two models with an R-squared value of 0.9 (see Figure 6). While the performance was great, the training time was quite long making it inefficient for a larger dataset.

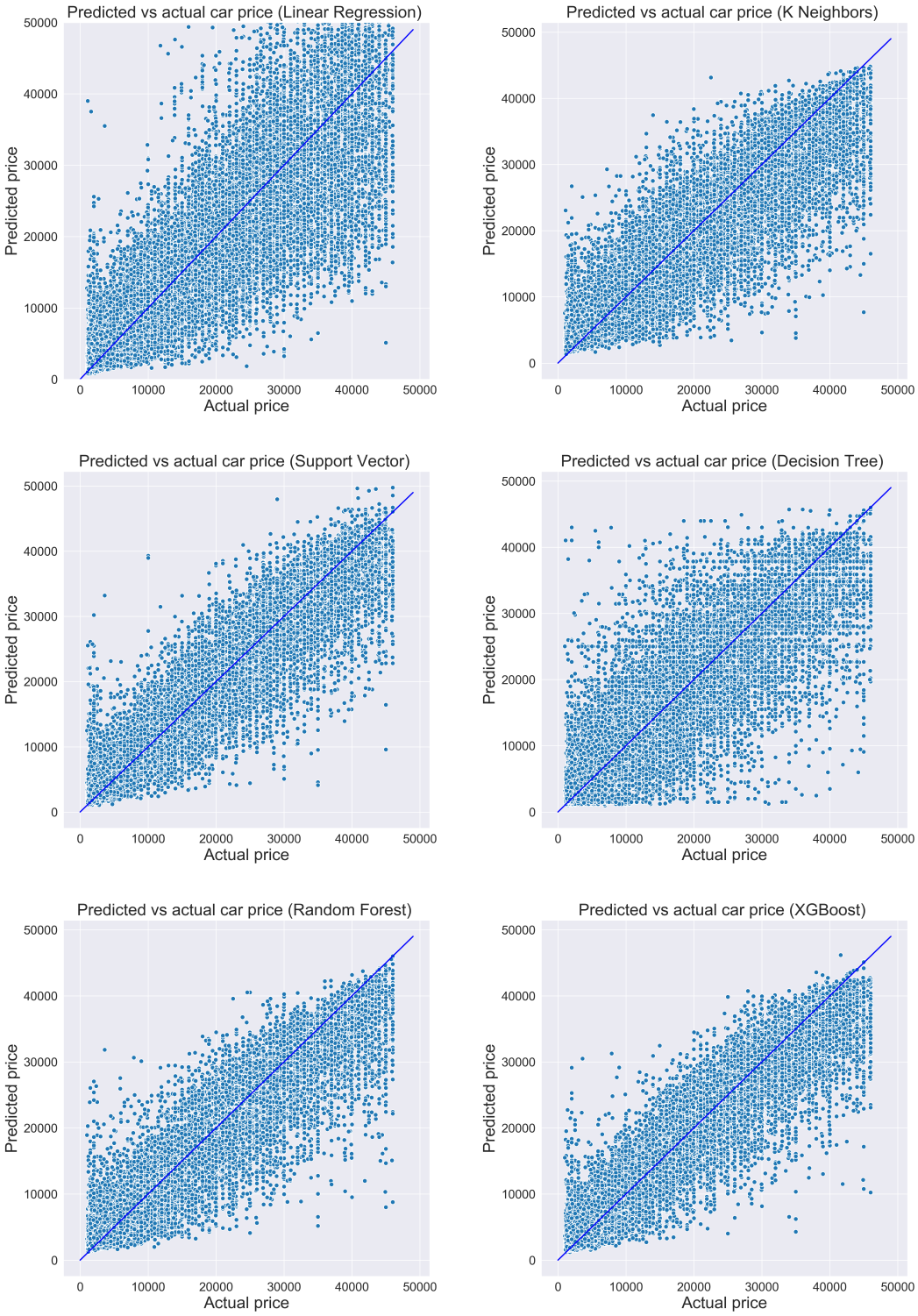

Figure 5. Actual car price vs predicted car price from the six regression model considered in this project. As shown in the figure, the dispersion of predicted value is minimum from Support Vector regression, Random Forest regression and XGBoost regression.

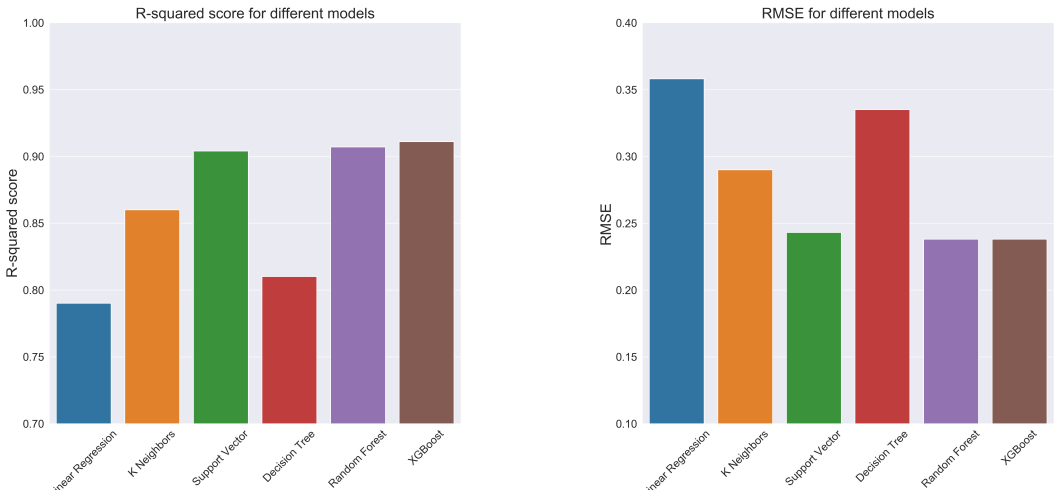

Figure 6. Comparison of R-squared values (left) and RMSE errors (right) of different machine learning models.

Then I considered three tree-based models, i.e., Decision Tree Regression, Random Forest Regression, and XGBoost regression. After fine-tuning, the Decision tree scored an R-squared of only about 0.81 (lower than KNeighbors and SVR). Next, I fined the tuned Random Forest Model to reach an R-squared score of close to 0.907. Then, finally, the XGBoost model (after fine-tuning) was able to score the best R-squared value of 0.911. Also, the variance of the predicted outcome is the least for XGBoost models (see Figure 5).

Overall, the XGBoost model performed the best among all the models considered. There were still large scatters in the predicted car values. One possible reason is that not all craigslist listings are legitimate. Often time, scammer post car ads with an unrealistic price. This would introduce variance in the datasets. These days, some used car dealers is using craigslist to post ads to promote their business. Most of the time, they only post downpayment prices for the listed. In principle, these could be removed by performing natural language processing on the car listing descriptions.

Finally, I think the model could be further improved by hyper tuning parameters in the whole training set. This took much longer to process, so I had to rely on a small subset to identity the best parameters. Provided computing resources, these models could be optimized to perform better.

References

1. https://www.grandviewresearch.com/industry-analysis/used-car-market?utm_source=prnewswire&utm_medium=referral&utm_campaign=fmcg_14-sept-20utm_term=used-car-market&utm_content=rd1
2. https://www.kaggle.com/austinreese/craigslist-carstrucks-data