BAX 452 Final Project

Prediction of Pre-owned Car Price

Jiaqi Lu, Xiaochen Guo, Letian (William) Ma

Executive Summary **2**

Industry background **2**

Dataset **3**

Machine Learning Models **5**

Data pre-processing 5

Models 6

Linear regression 6

K-Nearest Neighbor 6

XGBoost 7

Random Forest 8

Model Comparison and Recommendation 9

Conclusion **11**

Executive Summary

One of our team members is thinking about buying a car recently. Buying a used car is an absolutely cost-effective choice, but how to get a car at a reasonable price is what we cared about. However, different from new cars with fixed prices, the price of used cars has so many determining elements, such as years of use, brands, etc. So it’s hard to decide on a uniform price for a used car. With this concern, our team decided to dive into the used car market. It took some effort to find a suitable dataset for our project from Kaggle, which contains relevant data provided by Craigslist. After cleaning the dataset, our team applied several cutting-edge models we learned from Machine Learning lessons to figure out the most critical factors affecting prices and select the best model to predict prices. Our project can offer our team members useful reference of the used car’s price and help more individuals who want to buy a used car get a bargain. What’s more, for used car sellers, they can offer a better service with the knowledge of essential features for used cars.

Industry background

The globally used car market size is expected to grow at a compound annual growth rate (CAGR) of 5.5% from 2020 to 2027 with a $1,332.2 billion market value in 2019, indicating that the used car market has great potential and more and more companies joined into this industry. With the increasing demand for used cars, the used car price prediction system to determine a car’s value based on multiple variables is needed. Currently, there are several online car price estimator systems like Kelley Blue Book, CarMax, CarFax, etc., but we don’t know how they determine the car price, and in this project, we seek to predict the car price based on the dataset. By predicting the car price accurately, we could help make the auto market more transparent and help the online pricing website use another model to double-validate their evaluations.

Dataset

The data used in this project was from Kaggle. The owner scraped the relevant information from Craigslist, one of the world’s largest collections of used vehicles for sale. The dataset provides car sales, including columns like price, condition, manufacturer, latitude/longitude, and 18 other categories and updates every few months.

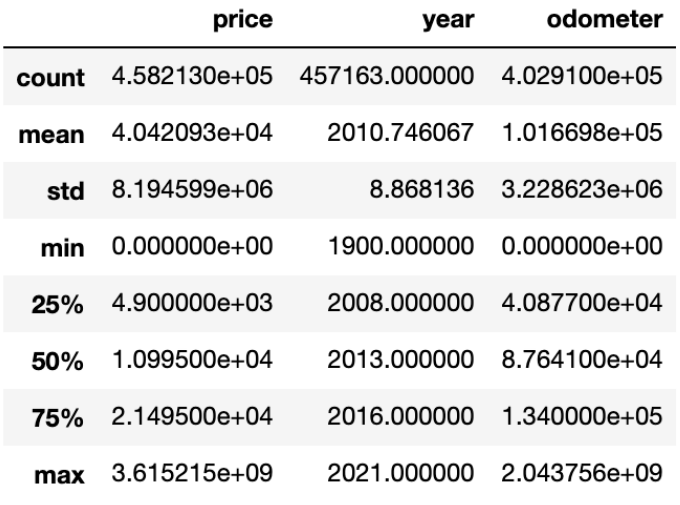
Data Wrangling and EDA

This section will discuss how we cleaned the dataset and how we handled the missing data and extreme values.

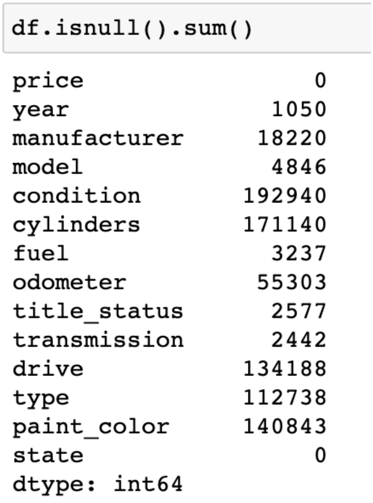
Data cleaning:

Before we did the data cleaning, we checked all the columns and their information. We manually eliminated some irrelevant columns, including ‘*id’, ‘url’, ‘image_url’, ‘size’, ‘region_url’, ‘VIN’, ‘lat/long’, ‘region’, etc.* Because our goal is to predict the price with shared information, the data which is unique for each car is not helpful for prediction.

In the next step, we checked null values and extreme data in the dataset.



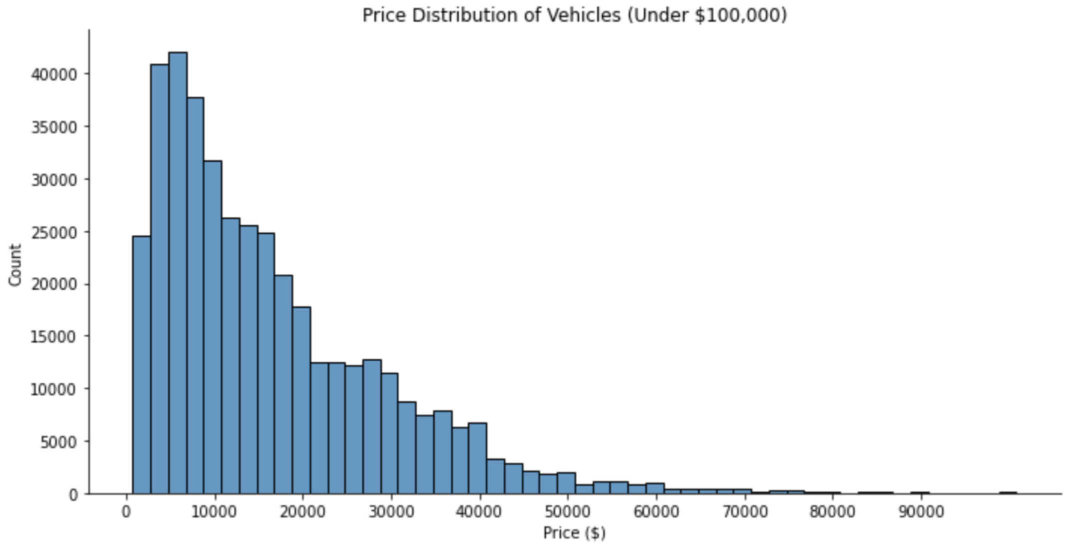
	price	year	odometer
count	4.582130e+05	457163.000000	4.029100e+05
mean	4.042093e+04	2010.746067	1.016698e+05
std	8.194599e+06	8.868136	3.228623e+06
min	0.000000e+00	1900.000000	0.000000e+00
25%	4.900000e+03	2008.000000	4.087700e+04
50%	1.099500e+04	2013.000000	8.764100e+04
75%	2.149500e+04	2016.000000	1.340000e+05
max	3.615215e+09	2021.000000	2.043756e+09



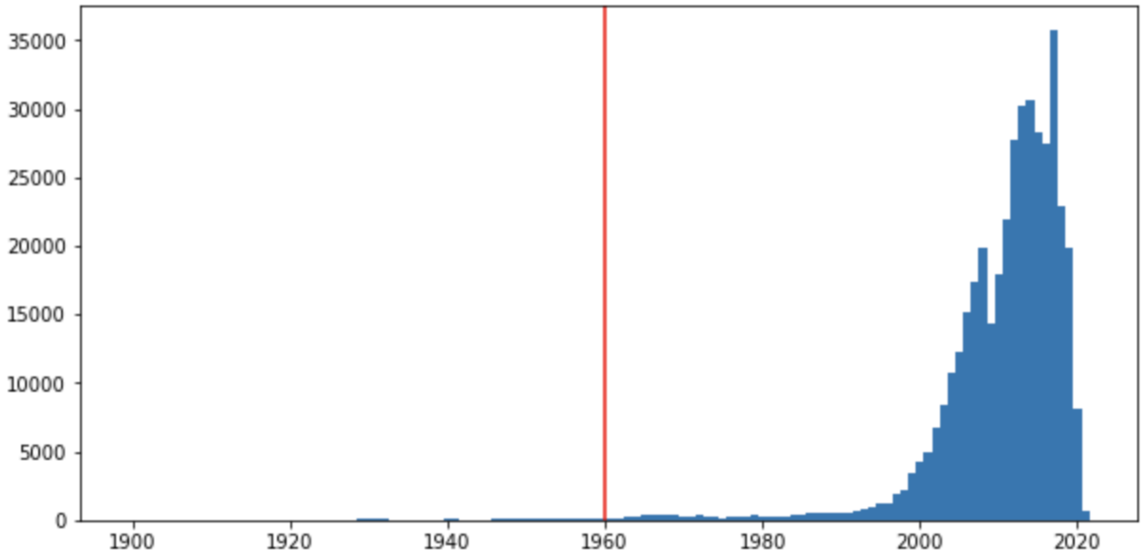
```
df.isnull().sum()
```

```
price                0
year              1050
manufacturer     18220
model             4846
condition       192940
cylinders       171140
fuel              3237
odometer         55303
title_status      2577
transmission      2442
drive           134188
type            112738
paint_color     140843
state                0
dtype: int64
```

As we observe, we first eliminate the extreme price as they restrain the prediction value. We dropped the rows with prices over $100,000 and less than $750 as the price doesn't represent the typical market values.



After handling the price, we dropped the car generated before the 1960s as they should no longer be in the market. We deleted the rows in which the odometer is in the top1% mileages.



We deleted the unknown fuel, transmission manufacturer, and title_status.

Next, we handled the missing data. For the condition column, we labeled the car generated after 2019 as ‘new’ and before 2017 as ‘like new’. We filled the rest as ‘unknown’ since the condition doesn’t correlate with the odometers. Lastly, we filled others using the ‘ffill’ method. This method is valid by propagating the last valid observation forward to the next valid based on time-series. Thus, the last known value is available. So far, we cleared all the missing values in odometer, condition, price, and other features. We received a cleaned dataset with 14 columns and 342,702 observations, and stored it in a CSV file.

Machine Learning Models

In this section, we used different algorithms to predict the price of preowned cars. As the dataset is supervised data with dependent variables [‘price’] and other independent variables (car’s features), we set the goal of this project as the accuracy of predicting the price for future observations. Based on the dataset columns, we decided to apply the methods below:

- Simple Linear Model (Baseline)
- K-Nearest Neighbor
- XGBoost
- Random Forest

Data pre-processing

Label Encoding. Before applying different methods, we check the data type of each column, and only 2 of them are numerical variables as the rest 12 are categorical. We used Label Encoding to transform the categorical data into numerical form to build the machine learning models.

Data Scaling. Also, to avoid the range of values influencing the prediction, we standardized each column using Min-Max scaler, which transforms features in the given range. Without scaling, the estimated coefficients would be inappropriate because large value features will take larger weights and smaller values will get low values due to their scales.

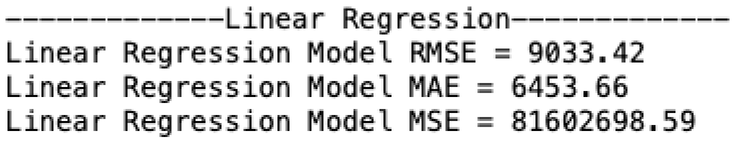
Reduce ‘model’ type. We observed that the model includes too many models from different manufacturers and ‘F-350’ is the same as ‘f350 lariat’. To reduce the model variety, we only fetched the left word in the string and reduced the car model type from 2839 to 292.

Data Split. 20% of the data was split to test, and 80% of the data was split to train the model.

Models

Linear regression

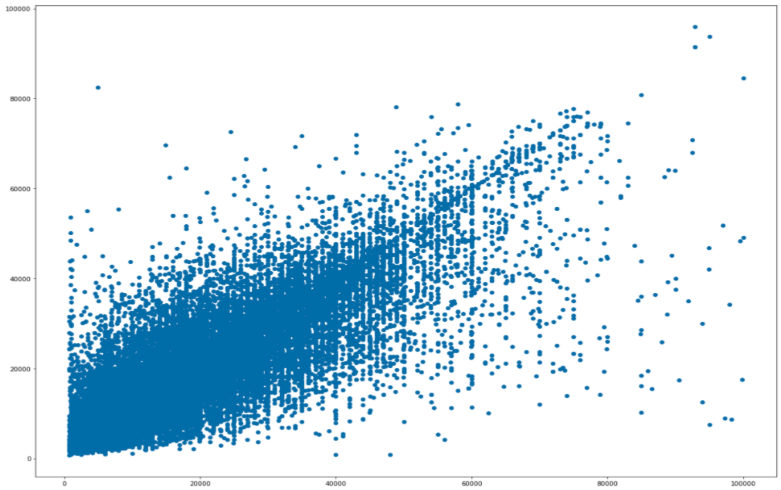
We use linear regression as a baseline, and in linear regression, RMSE is 9033.42.



```
-------------Linear Regression-------------
Linear Regression Model RMSE = 9033.42
Linear Regression Model MAE = 6453.66
Linear Regression Model MSE = 81602698.59
```

K-Nearest Neighbor

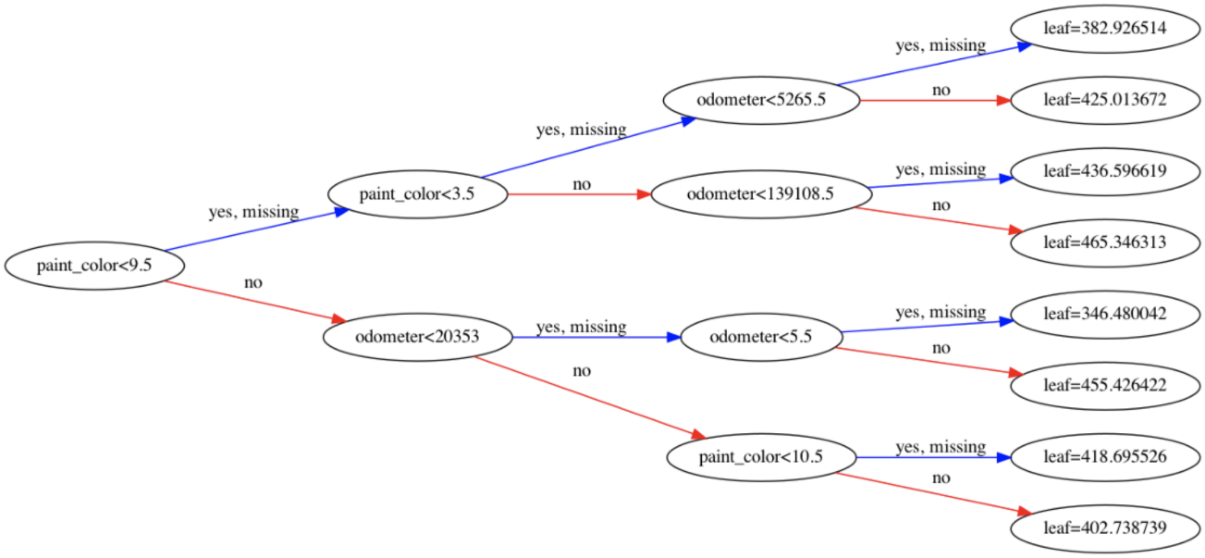
When the dataset is small, using KNN can be an appropriate choice, especially with multiple categorical variables. After calculating the distance using Hamming distance method, we create multiple k factors and compute Root Square Mean Error for each k factor. By applying cross-validated grid-search over a parameter grid, we got the optimized best number of Neighbor k = 4. However, KNN’s prediction is hard to interpret, though it can get high accuracy. The scatter plot below shows the prediction using the KNN model.



XGBoost

XGBoost is a powerful machine learning method that implements parallel tree boosting fast and accurately. Before implementing the XGBoost, we constructed the dataset into a dense matrix and set up three parameters: General parameter, Booster parameter and Learning task parameters. As the dependent variable is numeric, We fitted the model with a tree booster, set the objective as 'reg: squarederror', learning rate (eta) = 0.3 as shrinkage size, max_depth as 5 to make the tree model simple.

By applying the 5-fold cross-validation, we estimated the accuracy by evaluating the Root Mean Square Error in each fold.

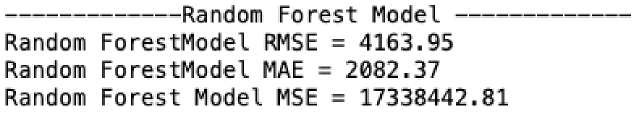


However, the XGBoost has its limitations ------ as the decision tree shows below, the XGBoost model predicts the final result in each branch and each leaf represents the sum of predictions.

Random Forest

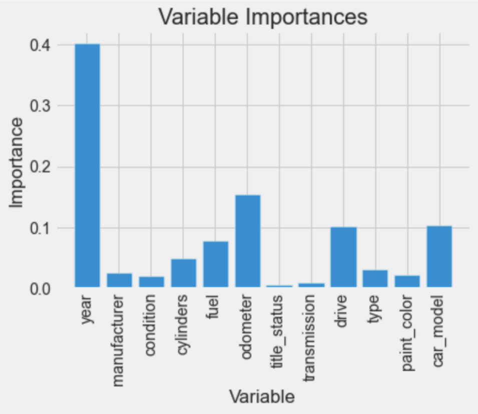
First, we used Random Forest. Random forest is a set of decision trees, which is used for classification and regression. Based on the bootstrap method, random forest shows the real structure of the dataset and averages out the useless signals.

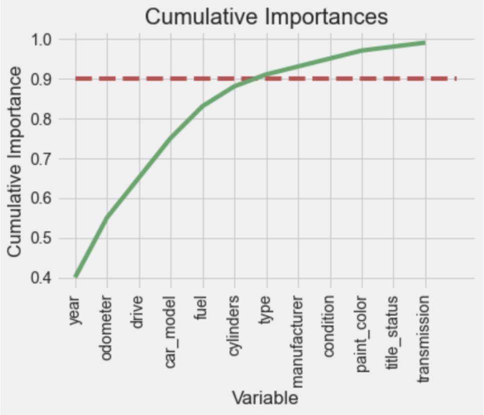
We grew 100 trees in this forest using all 12 variables. We use MAE and RMSE to measure the performance of prediction. As a result, MAE is 2082.37 and RMSE is 4163.95.

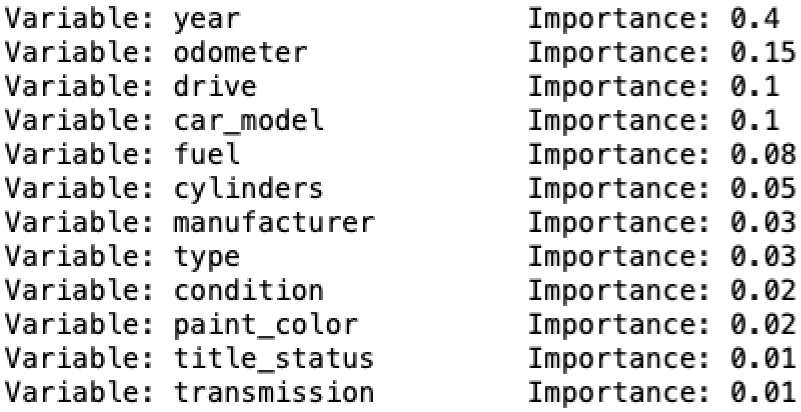


```
-------------Random Forest Model -------------
Random ForestModel RMSE = 4163.95
Random ForestModel MAE = 2082.37
Random Forest Model MSE = 17338442.81
```

We look at the relative importance of variables to see if we need to select variables to make a better model. The following pictures show the relative importance of each variable and cumulative importances. The importance of ‘year’ is up to 0.4, which means it is the most important variable in the random forest model, this is in line with reality. Year of use is generally the first question asked by customers. The next variable is ‘odometer’, which also largely affects the choice of individuals. The Sum of importances of year, odometer, drive, car_model, fuel, cylinders, and type reached 90% so that we used these seven variables to build the next model.

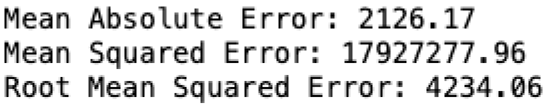






```
Variable: year                   Importance: 0.4
Variable: odometer               Importance: 0.15
Variable: drive                  Importance: 0.1
Variable: car_model              Importance: 0.1
Variable: fuel                   Importance: 0.08
Variable: cylinders              Importance: 0.05
Variable: manufacturer           Importance: 0.03
Variable: type                   Importance: 0.03
Variable: condition              Importance: 0.02
Variable: paint_color            Importance: 0.02
Variable: title_status           Importance: 0.01
Variable: transmission           Importance: 0.01
```

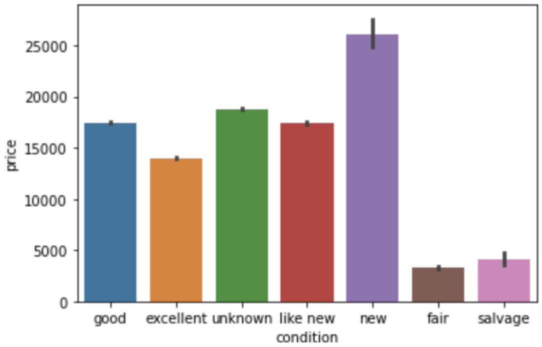
We still grew 100 trees but this time we used seven selected variables. MAE is 2126.17 and RMSE is 4234.06, which indicates that the model using selected variables doesn't improve prediction accuracy. Therefore, we chose the model with all 12 variables.

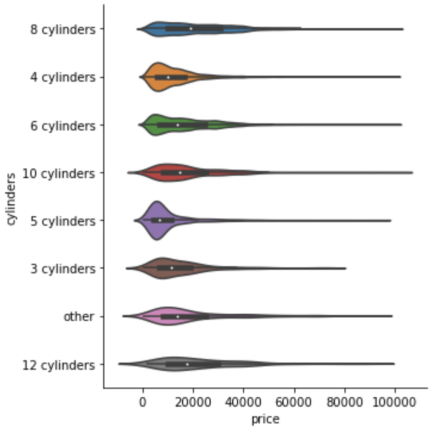


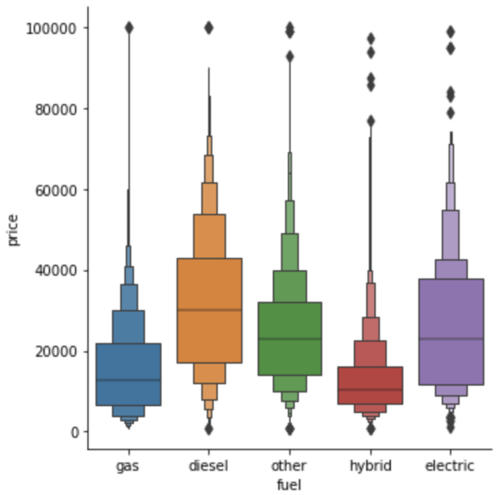
```
Mean Absolute Error: 2126.17
Mean Squared Error: 17927277.96
Root Mean Squared Error: 4234.06
```

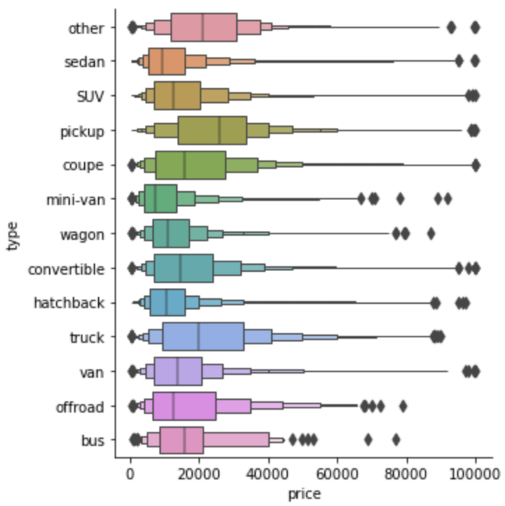
Model Comparison and Recommendation

Among different models, the random forest model performs better compared to other models. We found that Year, Odometer, Car_model, Drive are the most important features to drive the price. Cars using diesel or electric fuel are also having relatively higher prices, which makes sense because most cars that use diesel are trucks.

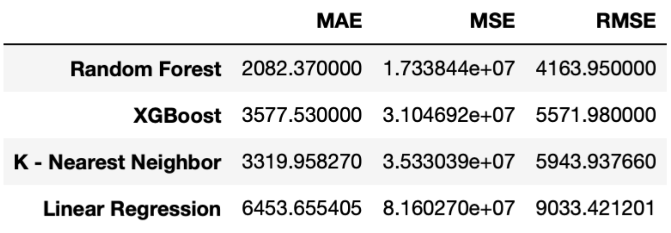








For the online price evaluation system, it is important to set the price major considering the age, fuel, model of the car. As Condition is a subjective description that is based on multiple other variables, it should not be put on the same weights as other features. We recommend the system designer use Random Forest Model to make the prediction of the car price.



	MAE	MSE	RMSE
Random Forest	2082.370000	1.733844e+07	4163.950000
XGBoost	3577.530000	3.104692e+07	5571.980000
K - Nearest Neighbor	3319.958270	3.533039e+07	5943.937660
Linear Regression	6453.655405	8.160270e+07	9033.421201

Conclusion

The main goal of this project is to understand the relationship between the preowned car price with its features, to make the prediction of the used car price with multiple machine learning models. The dataset was uncovered and features were explored deeply. After implementing data cleaning, exploratory data analysis, data pre-processing, we applied different models to predict the price of cars as Linear Regression, KNN, XGBoost, and Random Forest. Lastly, we evaluated different models given RMSE, MAE, and MSE values and concluded that random forest is the best model.

As a suggestion for further studies, we recommend using a bigger dataset and include more features like the gas/mile (per gallon), number of doors, whether the headlight is LED, the interior quality, etc. Moreover, the data cleaning and feature engineering process can evolve a more detailed analysis of the interaction among features.