Car Price Prediction with Linear Regression

Darren Liu

Introduction

Motivation: Nowadays, because of the lack of chips, used cars are getting more and more expensive. People are usually confused if they should get a car now or if they should wait. Hence, more and more people need a car price predicting tool to predict the price of used cars to avoid spending too much money.

Objectives and goals: The goal of this project was to build a regression model to predict used car prices.

Methodology

The data I used:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1193 entries, 0 to 1194
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   index         1193 non-null   int64
 1   name          1193 non-null   object
 2   price         1193 non-null   object
 3   mileage       1193 non-null   object
 4   ex_color      1193 non-null   object
 5   in_color      1193 non-null   object
 6   drivetrain    1193 non-null   object
 7   mpg           1193 non-null   object
 8   fuel_type     1193 non-null   object
 9   transmission  1193 non-null   object
 10  engine        1193 non-null   object
dtypes: int64(1), object(10)
memory usage: 111.8+ KB
```

Used car data scraped from cars.com

Methodology

Tools I used:

Pandas for exploratory data analysis

Matplotlib and Seaborn for plotting

Beautifulsoup and Requests for web scraping

Scikit Learn and Statsmodels for building regression models

Pickle for saving regression models in a pickle file

Methodology

How I used the data:

Firstly, I decided to drop the color column because many of the color names are confusing and almost impossible to turn into numerical data. Besize, the color of the car will not have a major influence on the car price in most cases.

Secondly, I got the car years and makes from their names. The year column is turned into the age column based on the current year. Then I made dummy variables for car makes.

Furthermore, fuel type is separated into gas, diesel and electric, final mpg is calculated by adding the lowest mpg and highest mpg together, and transmission is divided into auto, cvt, and manual.

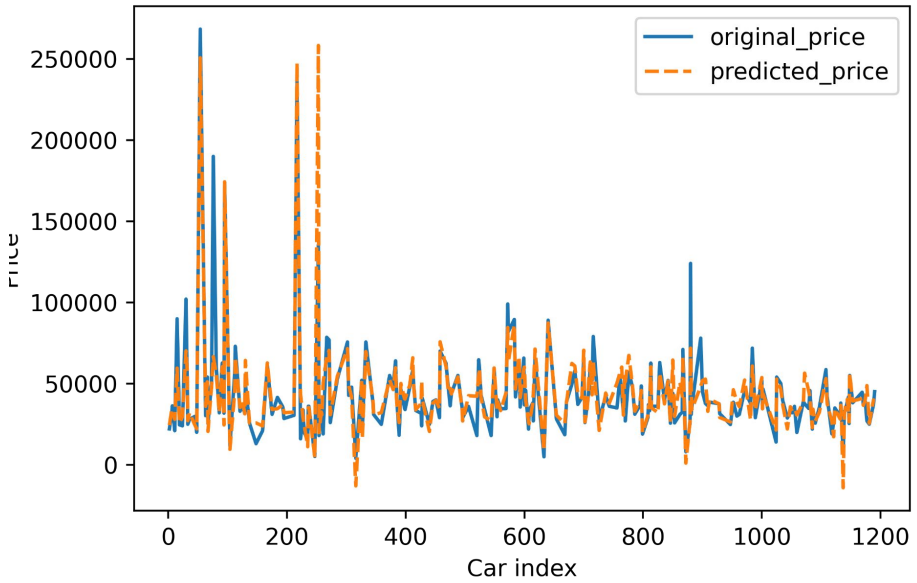
At last, engine is divided into engine_L as liters and engine_V as volts because I don't understand what other things are. Drivetrains are combined into 2 types: 2-wheel-drive and 4-wheel-drive.

Results

Initial model performance:

R^2 value on training dataset: 0.958

R^2 value on validation dataset: 0.765

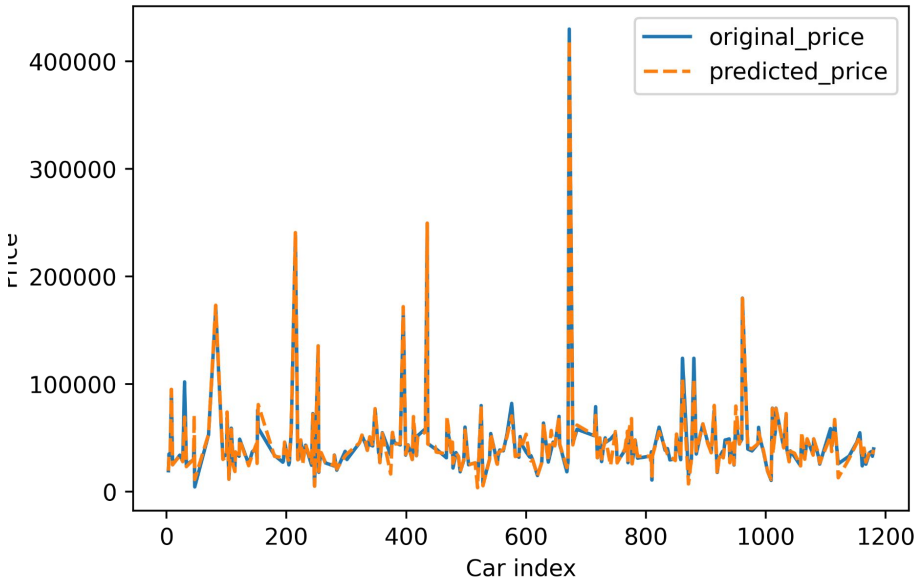


Results

After creating polynomial features and solving overfitting through lasso regression:

R^2 value on training dataset: 0.985

R^2 value on validation dataset: 0.980



Conclusions

The model generated for now is quite accurate to me, but it can still be improved by adding much more training data and including more features and car makes and models.

Future Work

There are lots of car features not included in the model, such as engine type, color, wheel size, intelligent features and so on. In the future, improving the model by including more data from cars.com or even other websites and more features is one of the most important things to do that will help a lot with the accuracy.

Thanks for your attention!