Car price prediction

Priyadarsi Das

**ACKNOWLEDGMENT**

References:

* Data Trained Course Materials / Videos / Projects
* Udemy course: Data Science Bootcamp by Jose Portilla
* Youtube Video Links:

<https://www.youtube.com/watch?v=yoLpcelanpI&t=1908s&ab_channel=KrishNaik>

<https://www.youtube.com/watch?v=6WDFfaYtN6s&list=PLZoTAELRMXVPwYGE2PXD3x0bfKnR0cJjN&ab_channel=KrishNaik>

Professionals that helped me in completion of the project:

* Sajid Choudhary

INTRODUCTION

Business Problem Framing

With the covid 19 impact in the market, we have seen lot of changes in the car market. Now some cars are in demand hence making them costly and some are not in demand hence cheaper. One of our clients works with small traders, who sell used cars. With the change in market due to covid 19 impact, our client is facing problems with their previous car price valuation machine learning models. So, they are looking for new machine learning models from new data. We have to make car price valuation model.

Data Collection Phase

You have to scrape at least 5000 used cars data. You can scrape more data as well, it’s up to you. more the data better the model

In this section You need to scrape the data of used cars from websites (Olx, cardekho, Cars24 etc.) You need web scraping for this. You have to fetch data for different locations. The number of columns for data doesn’t have limit, it’s up to you and your creativity. Generally, these columns are Brand, model, variant, manufacturing year, driven kilometers, fuel, number of owners, location and at last target variable Price of the car. This data is to give you a hint about important variables in used car model. You can make changes to it, you can add or you can remove some columns, it completely depends on the website from which you are fetching the data.

Try to include all types of cars in your data for example- SUV, Sedans, Coupe, minivan, Hatchback.

Model Building Phase

After collecting the data, you need to build a machine learning model. Before model building do all data pre-processing steps. Try different models with different hyper parameters and select the best model.

Follow the complete life cycle of data science. Include all the steps like.

1. Data Cleaning

2. Exploratory Data Analysis

3. Data Pre-processing

4. Model Building

5. Model Evaluation

6. Selecting the best model

Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem

The following models were used on the dataset:

* GradientBoosting
* RandomForest
* ExtraTree
* XGB
* Bagging
* Huber
* ElasticNet
* BayesianRidge
* KNeighbors
* Ridge
* Lasso
* LinearRegression
* SGD
* AdaBoost
* DecisionTree
* RANSAC

Data Sources and their formats

Variables

Make

Model

Year

Owner

Transmission

Fuel

Kms

Location

Price (Target Variable)

Steps

Web scrapping Data from olx.com

Perform Exploratory Data Analysis

Handle Null Values

Check for outliers

Check co-relation of variables

Check for skewness

Find best random state

Train Test Split

Model Building, Feature Selection, Feature Extraction

Hyper-parameter Tuning

Saving the model

EDA Observations

Most of the houses have the following properties:

1. , is replaced from the columns
2. Rows having String values in integer columns are dropped
3. Object columns with integer values are converted to integer
4. Object columns with string values are label encoded
5. On removing outliers by z score method, 5% data loss is incurred, hence outliers are removed from the dataset
6. Skewness is checked with Power transformation. However, the base data has no skewness, hence transformer is not applied
7. The best random state is found to be 71. Same is used for train test split.
8. There are no null values in the dataset
9. Most of the cars were bought in between 2000 – 2010
10. Most of the cars were used by 1 owner
11. Transmission has 2 categories: Automatic and Manual
12. Fuel has 5 categories
13. Variable linearly related to Price column: Year, Number of Owners, Kms used
14. Automatic vehicles are high priced
15. Petrol vehicles have highest price followed by LPG and then Electric
16. CNG & Hybrid vehicles are lowest priced vehicles

Hardware, Software and Libraries Used for the project

Hardware used – Intel i5 Laptop, 12 GB RAM, 2 GB Graphics, 240 GB SSD

Software used – Anaconda Jupyter Notebook

Packages used – numpy, pandas, matplotlib, seaborn, sklearn, scipy

Model/s Development and Evaluation

Model Building without Pipelines - Results

Linear Regression

R2 Score: 0.40

It is seen that applying lasso does not improve the score

Catboost Regressor

R2 Score 0.52

Decision Tree Regressor

R2 Score: 0.48

----------------------------------------------------

Hyperparameter tuning on Catboost Regressor gives us

R2 Score: 0.77

Model Building with Pipelines – Results

Standard Scaler + Different Models gives the following results:

('XGB', 0.8424667717647588),

('RandomForest', 0.7949179019446566),

('Bagging', 0.7762161601759182),

('GradientBoosting', 0.734116661368993),

('ExtraTree', 0.7233775207437104),

('DecisionTree', 0.5909982204250677),

('KNeighbors', 0.5089155317478375),

('AdaBoost', 0.38373907347431807),

('BayesianRidge', 0.3502038442424747),

('Ridge', 0.3501652582578137),

('Lasso', 0.350162802733961),

('LinearRegression', 0.3501626546848372),

('SGD', 0.34979365115350963),

('ElasticNet', 0.3222970175035715),

('Huber', 0.2647695899098993),

('RANSAC', 0.08260287556220287)

Applying Feature Selection

SelectKBest and f\_regression is used to select 30 best parameters

('XGB', 0.849262991659535),

('RandomForest', 0.7997260800872877),

('Bagging', 0.7835757928149858),

('GradientBoosting', 0.7300818306326006),

('ExtraTree', 0.7185206656951154),

('DecisionTree', 0.6627224674221378),

('KNeighbors', 0.5456486678739007),

('BayesianRidge', 0.35023447737418684),

('Ridge', 0.35020159541006624),

('Lasso', 0.3501992085305186),

('LinearRegression', 0.350199069913208),

('SGD', 0.3497131339337049),

('ElasticNet', 0.3223121412271344),

('AdaBoost', 0.2872911020457127),

('Huber', 0.2648569717502475),

('RANSAC', 0.07812810827011604)

Applying Feature Extraction

Standard Scaler and PCA are used to perform Feature Extraction

('ExtraTree', 0.5879755665783052),

('RandomForest', 0.5552465764365048),

('XGB', 0.527519633941352),

('KNeighbors', 0.5016577937919695),

('Bagging', 0.49949617075948327),

('GradientBoosting', 0.48182888661437456),

('BayesianRidge', 0.35020043398830386),

('Ridge', 0.3501654713184062),

('Lasso', 0.35016272284814),

('LinearRegression', 0.3501626546848372),

('SGD', 0.34855478257073863),

('ElasticNet', 0.3139694795721067),

('Huber', 0.264555989435908),

('DecisionTree', 0.19767011293968223),

('AdaBoost', 0.19442699253775547),

('RANSAC', 0.08051001481974362)

Hyper parameter Tuning of Final Models

Models and R2 Score with GridSearchCV

'RandomForest', 0.77

'GradientBoosting', 0.83

‘Random Forest’ : 0.79

‘XGB’ : 0.85

Conclusion

Base models with pipelines will all features gives better results than applying Feature Selection & Feature Extraction.

XGB is selected as the final model at R2 Score of 85.87 with Hyper-parameter tuning.