Used Car Price Prediction

Jialong Zhang
Zehao Liu
Instructor: Dr. Sukhjit Singh Sehra
July 19, 2022

Abstract

A lot of people buy used cars these days because the price of new cars is often too expensive. However, there are many risks associated with buying a used car. Using our model to predict a reasonable price before buying a used car can help buyers clarify their goals.

1. Introduction

A huge amount of people desired to own a car to make their life convenience and the price of car is big concern of those afraid to be cheated and a lot of people waste a plenty of money on not bargain deals on used car. To address the issue, project aims to design a two-stage machine learning engine to predict the price of car by inputting the core property of the car. The price is predominantly influenced by main 14 property in table 1.

Section 2 explains how a serial of property were processed to construct the dataset. Section 3 and Section 4 deal with how different classifiers and regressors were trained and analyzed on the dataset respectively. Finally, Section 5 details the random forest model to predict car's prices

2. Project Description

The dataset comes from https://www.kaggle.com/datasets/avikasliwal/used-cars-price-prediction. It contains test-data.csv and train-data.csv, our project is just using train-data. data contains 14 attributes from table 1, we will do data cleaning on these raw datasets to deal with the missing data and redundant attributes.

Table 1: Data Attributes

index	
Name	The brand and model of the car.
Location	The location in which the car is being sold or is available for purchase.
Year	The year or edition of the model.
Kilometers_Driven	The total kilometres driven in the car by the previous owner(s) in KM.
Fuel_Type	The type of fuel used by the car. (Petrol / Diesel / Electric / CNG / LPG)

Transmission	The type of transmission used by the car. (Automatic / Manual)
Owner_Type	Whether the ownership is Firsthand, Second hand or other.
Mileage	The standard mileage offered by the car company in kmpl or km/kg
Engine	The displacement volume of the engine in cc.
Power	The maximum power of the engine in bhp.
Seats	The number of seats in the car.
New_Price	The price of a new car of the same model.
Price	The price of the used car in INR Lakhs.

To processing the data, we first imported the necessary libraries (e.g. pandas, numpy, matplotlib and seaborn) and loaded the dataset.

```
# Importing necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

We used the ".head()" and ".tail()" functions from the pandas library to return the first and last 5 observations of the dataset allowing a closer look at the data.

```
   Unnamed: 0                              Name    Location  Year  Kilometers_Driven Fuel_Type  ...      Mileage   Engine      Power Seats  New_Price  Price
0           0            Maruti Wagon R LXI CNG      Mumbai  2010              72000       CNG  ...   26.6 km/kg   998 CC  58.16 bhp   5.0        NaN   1.75
1           1  Hyundai Creta 1.6 CRDi SX Option        Pune  2015              41000    Diesel  ...   19.67 kmpl  1582 CC  126.2 bhp   5.0        NaN  12.50
2           2                      Honda Jazz V     Chennai  2011              46000    Petrol  ...    18.2 kmpl  1199 CC   88.7 bhp   5.0  8.61 Lakh   4.50
3           3                 Maruti Ertiga VDI     Chennai  2012              87000    Diesel  ...   20.77 kmpl  1248 CC  88.76 bhp   7.0        NaN   6.00
4           4   Audi A4 New 2.0 TDI Multitronic  Coimbatore  2013              40670    Diesel  ...    15.2 kmpl  1968 CC  140.8 bhp   5.0        NaN  17.74
```

```
      Unnamed: 0                      Name   Location  Year  Kilometers_Driven Fuel_Type  ...     Mileage   Engine      Power Seats  New_Price  Price
6014        6014          Maruti Swift VDI      Delhi  2014              27365    Diesel  ...   28.4 kmpl  1248 CC     74 bhp   5.0  7.88 Lakh   4.75
6015        6015  Hyundai Xcent 1.1 CRDi S     Jaipur  2015             100000    Diesel  ...   24.4 kmpl  1120 CC     71 bhp   5.0        NaN   4.00
6016        6016     Mahindra Xylo D4 BSIV     Jaipur  2012              55000    Diesel  ...   14.0 kmpl  2498 CC    112 bhp   8.0        NaN   2.90
6017        6017        Maruti Wagon R VXI    Kolkata  2013              46000    Petrol  ...   18.9 kmpl   998 CC   67.1 bhp   5.0        NaN   2.65
6018        6018     Chevrolet Beat Diesel  Hyderabad  2011              47000    Diesel  ...  25.44 kmpl   936 CC   57.6 bhp   5.0        NaN   2.50
```

We use “.info()" to know the columns and their corresponding data types.

	Year	Kilometers_Driven	Seats	Price
count	6019.000000	6.019000e+03	5977.000000	6019.000000
mean	2013.358199	5.873838e+04	5.278735	9.479468
std	3.269742	9.126884e+04	0.808840	11.187917
min	1998.000000	1.710000e+02	0.000000	0.440000
25%	2011.000000	3.400000e+04	5.000000	3.500000
50%	2014.000000	5.300000e+04	5.000000	5.640000
75%	2016.000000	7.300000e+04	5.000000	9.950000
max	2019.000000	6.500000e+06	10.000000	160.000000

Use describe() function returns the count, average, standard deviation, minimum, maximum and quantity of the data.

```
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Unnamed: 0         6019 non-null   int64
 1   Name               6019 non-null   object
 2   Location           6019 non-null   object
 3   Year               6019 non-null   int64
 4   Kilometers_Driven  6019 non-null   int64
 5   Fuel_Type          6019 non-null   object
 6   Transmission       6019 non-null   object
 7   Owner_Type         6019 non-null   object
 8   Mileage            6017 non-null   object
 9   Engine             5983 non-null   object
 10  Power              5983 non-null   object
 11  Seats              5977 non-null   float64
 12  New_Price          824 non-null    object
 13  Price              6019 non-null   float64
dtypes: float64(2), int64(3), object(9)
```

Now, we have a good basic understanding of the data. Then we perform data cleanup, Use ".isnull().sum()" to find the missing data, then use ".dropna()" to delete the records with NULL values, and finally use "reset_index" to reset the index.

```
Name                   0
Location               0
Year                   0
Kilometers_Driven      0
Fuel_Type              0
Transmission           0
Owner_Type             0
Mileage                2
Engine                36
Power                 36
Seats                 42
New_Price           5195
Price                  0
```

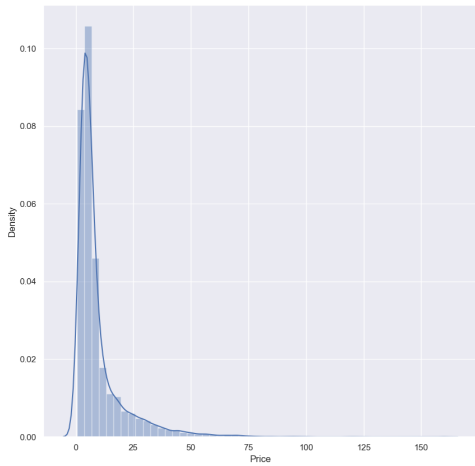
```
#delete the records with NULL values
print(train_data.isnull().sum())
train_data.drop(["New_Price"],axis=1,inplace=True)
train_data = train_data.dropna(how='any')
train_data = train_data.reset_index(drop=True)
```

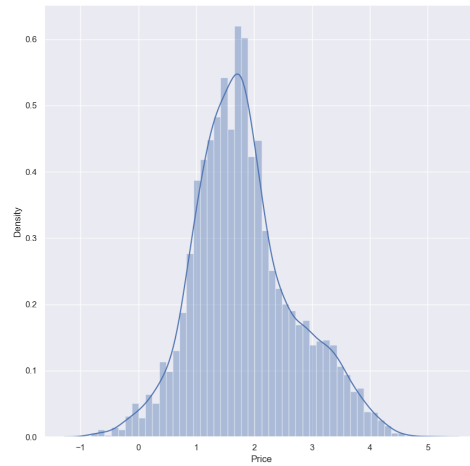
We found some data with units, we need to remove the units from the data and change the data type.

```
# remove some units from the data
train_data['Mileage'] = train_data['Mileage'].str.replace(' kmpl','')
train_data['Mileage'] = train_data['Mileage'].str.replace(' km/kg','')
train_data['Engine'] = train_data['Engine'].str.replace(' CC','')
train_data['Power'] = train_data['Power'].str.replace('null bhp','112')
train_data['Power'] = train_data['Power'].str.replace(' bhp','')

train_data['Mileage'] = train_data['Mileage'].astype(float)
train_data['Engine'] = train_data['Engine'].astype(float)
train_data['Power'] = train_data['Power'].astype(float)
```

After pre-processing the data, we used the visualization library Seaborn to create various statistical plots for visual exploration analysis. We first performed the distribution of prices.



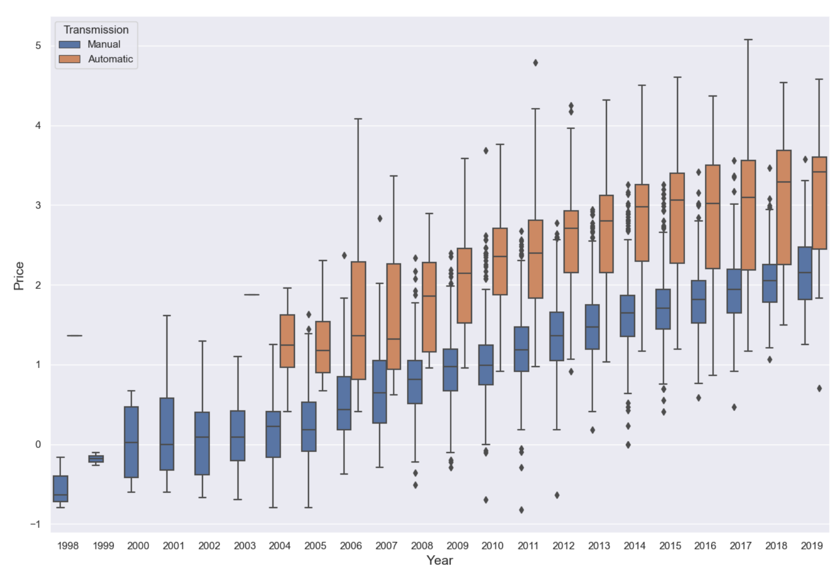


We observe that the distribution of prices shows a high positive skewness to the right, implying a large number of outliers in the data set. (left image). We use Log(Price) to transform the value of Price to visualize the distribution of prices more normally. (As shown in the figure on the right).

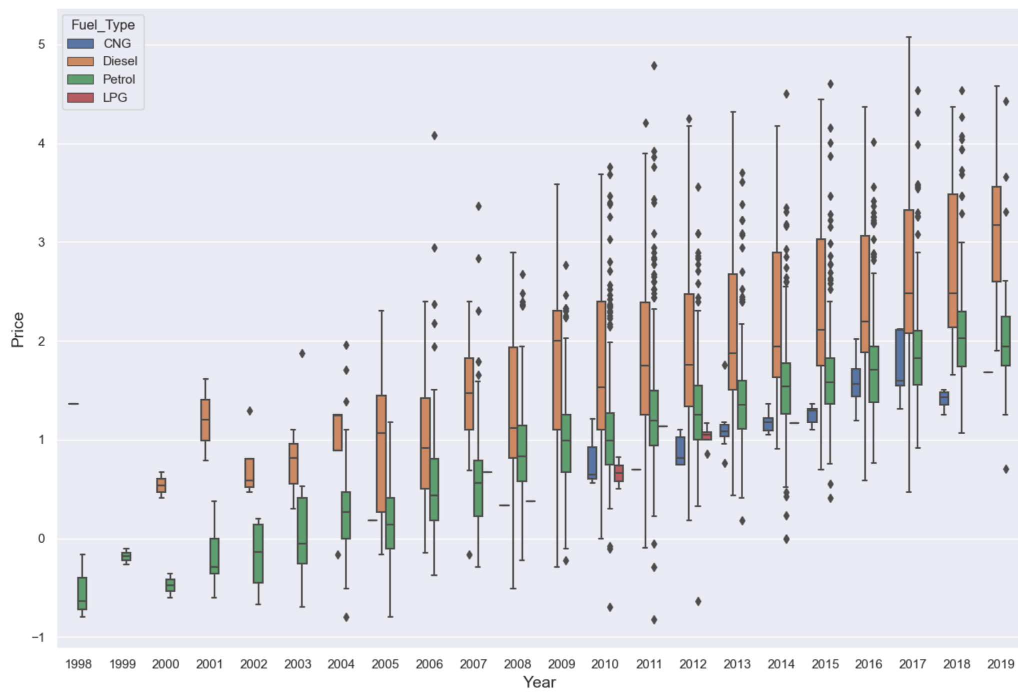
we use the ".corr()" function of pandas to find correlations and use the heat map in seaborn to visualize the correlation matrix.

```
#heatmap
plt.figure(figsize=(10,10))
sns.heatmap(data.corr(),annot=True,cmap='RdYlGn')
plt.show()
```

The heat map shows the relationship between each parameter. It is clear that price has a strong positive correlation with the ‘power’, the ‘engine’ and the ‘year’ of the car. Price has a strong negative correlation with the ‘Mailage' of the car. Price has a little correlation with 'Kilometers_Driven' and 'Seats'. This indicates that people pay more attention to the powertrain of a car when buying a used car.



This box plot tells us that the newer the car, the more expensive it is. Automatic cars are always more expensive than manual cars, so we conclude that automatic cars cost more.



This box plot tells us that diesel cars are more expensive than Petrol cars.

We handled the categorical data with LabelEncoder before proceeding with the modeling.

```
#Handling Categorical parameters
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder().fit(data['Cars'])
data['Cars'] = label_encoder.transform(data['Cars'])

label_encoder = LabelEncoder().fit(data['Location'])
data['Location'] = label_encoder.transform(data['Location'])

label_encoder = LabelEncoder().fit(data['Fuel_Type'])
data['Fuel_Type'] = label_encoder.transform(data['Fuel_Type'])

label_encoder = LabelEncoder().fit(data['Transmission'])
data['Transmission'] = label_encoder.transform(data['Transmission'])

label_encoder = LabelEncoder().fit(data['Owner_Type'])
data['Owner_Type'] = label_encoder.transform(data['Owner_Type'])
```

3.Methodology

The project execute six difference model to fitting the data, including linear regression, Multilayer perceptron, KNeighbors classifier, decision tree classifier, decision tree regressor and random forest regressor in our example. Linear regression decision tree regressor and random forest regressor is successes to predict all the content, since their accuracy is over 0.8. Random forest have the highest accuracy that is around 0.94 so that it becomes our best choice of model.

Following is our graph:

	model	Root Mean Squared Error	Accuracy on Traing set	Accuracy on Testing set
3	KNeighborsClassifier	354.978130	0.210092	0.005578
5	Multilayer perceptron	331.878794	0.013490	0.006085
4	DecisionTreeClassifier	134.025082	0.997502	0.026369
2	LinearRegression	152.635554	0.847900	0.805983
0	DecisionTreeRegressor	116.530787	0.999993	0.886914
1	RandomForestRegressor	83.398528	0.991817	0.942078

Linear regression is one of the supervised Machine learning algorithms in Python that observes continuous features and predicts an outcome. It assigns optimal weights to variables to create a line ax+b to predict the output. [1]

Kneighbors is a supervised learning algorithm that considers different centroids and uses a usually Euclidean function to compare distance. Then, it analyzes the results and classifies each point to the group to optimize it to place with all closest points to it. It classifies new cases using a majority vote of k of its neighbours. The case it assigns to a class is the one most common among its K nearest neighbours. [2]

A decision tree falls under supervised Machine Learning Algorithms in Python and comes of use for both classification and regression. It takes an instance, traverses the tree, and compares important features with a determined conditional statement. [3]

A random forest is an ensemble of decision trees. In order to classify every new object based on its attributes, trees vote for class- each tree provides a classification. The classification with the most votes wins in the forest.[4]

A multilayer perceptron (MLP) is a fully connected class of feedforward artificial neural network (ANN). It can distinguish data that is not linearly separable. [5]

4.conclusion

This is a data analysis project for machine learning. We did clean the raw data by null vale. After data Preprocessing, six model are used to fitting our prediction. According to the analysis, random forest is best model in our prediction and this project is useful to predict the price of used car.

5. Reference

[1] https://en.wikipedia.org/wiki/Linear_regression

[2] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[3] https://en.wikipedia.org/wiki/Decision_tree_learning

[4] https://en.wikipedia.org/wiki/Random_forest

[5] https://en.wikipedia.org/wiki/Multilayer_perceptron