Regression Project with different Modelling techniques

Report
B.M.SURAJ

Introduction:

This project is for a second hand car selling company which is selling Toyota Corolla extensively. The company wants to predict the price of same cars left in the stock based on the past data of prices that have been sold to the customers. This prediction will help the company in predicting the price of this car model even before customer starts negotiating and sell it for a reasonable selling price.

Data description

The company wants us to use the data based on the following variables and neglect the remaining ones.

- Age: Provides the age of the car from manufacturing date.
- KM: Total kilometres covered by vehicle.
- HP: Horse power of the car.
- cc: Engine displacement measured in cubic capacity.
- Doors: No. of doors the particular model has as it comes in different variants.
- Gears: n speed Gearbox (n referring to no. of gears).
- Quarterly_tax: Amount paid as tax in dollars.
- Weight: Weight of the car.
- Fuel_type: Type of fuel used in the car.
- Price: Total price of the car (this is our target variable.

Methodology

- Importing basic libraries and loading dataset.
- Exploratory Data Analysis
- Determining Normality
- Train Test split
- Linear Regression Model
- Linear Regression Model with Polynomial Features
- Regularization Techniques with polynomial features
- Result

Dataset: After importing data set we had tweak a few changes like first removing unnecessary columns, then renamed the column with age for convenience, checked if there is any null values present in the dataset and assigned dummy variables to the column fuel type which consists of three fuel types – Diesel, petrol, CNG.

Our final data set looked like this:

	price	age	km	hp	cc	doors	gears	quarterly_tax	weight	CNG	Diesel	Petrol
0	13500	23	46986	90	2000	3	5	210	1165	0	1	0
1	13750	23	72937	90	2000	3	5	210	1165	0	1	0
2	13950	24	41711	90	2000	3	5	210	1165	0	1	0
3	14950	26	48000	90	2000	3	5	210	1165	0	1	0
4	13750	30	38500	90	2000	3	5	210	1170	0	1	0

Exploratory Data Analysis:

Then we started with our EDA which showed the shape of dataset as 1436 rows and 12 columns.

Then we used different methods in analysing the data which are shown below:

```
df.describe()
```

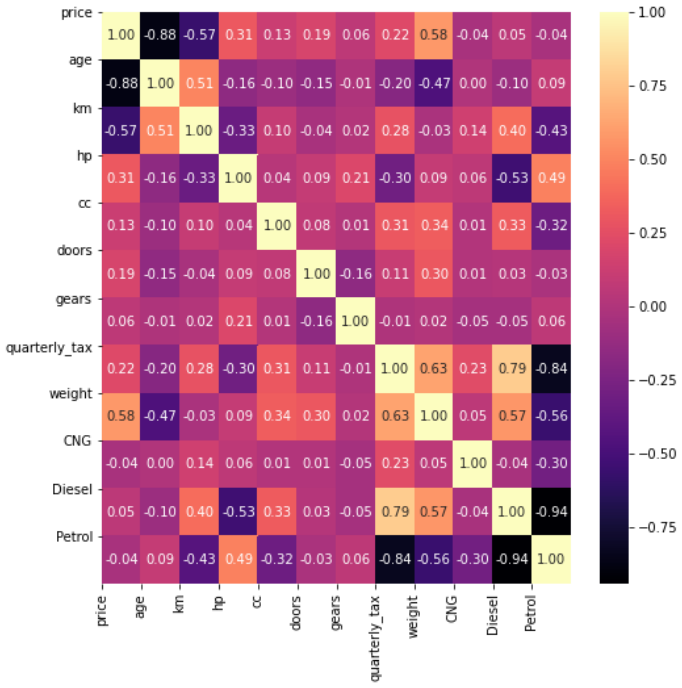
	price	age	km	hp	cc	doors	gears	quarterly_tax	weight	CNG	Diesel
count	1436.000000	1436.000000	1436.000000	1436.000000	1436.00000	1436.000000	1436.000000	1436.000000	1436.00000	1436.000000	1436.000000
mean	10730.824513	55.947075	68533.259749	101.502089	1576.85585	4.033426	5.026462	87.122563	1072.45961	0.011838	0.107939
std	3626.964585	18.599988	37506.448872	14.981080	424.38677	0.952677	0.188510	41.128611	52.64112	0.108196	0.310411
min	4350.000000	1.000000	1.000000	69.000000	1300.00000	2.000000	3.000000	19.000000	1000.00000	0.000000	0.000000
25%	8450.000000	44.000000	43000.000000	90.000000	1400.00000	3.000000	5.000000	69.000000	1040.00000	0.000000	0.000000
50%	9900.000000	61.000000	63389.500000	110.000000	1600.00000	4.000000	5.000000	85.000000	1070.00000	0.000000	0.000000
75%	11950.000000	70.000000	87020.750000	110.000000	1600.00000	5.000000	5.000000	85.000000	1085.00000	0.000000	0.000000
max	32500.000000	80.000000	243000.000000	192.000000	16000.00000	5.000000	6.000000	283.000000	1615.00000	1.000000	1.000000

Correlation:

```
df.corr()
```

	price	age	km	hp	cc	doors	gears	quarterly_tax	weight	CNG	Diesel	Petrol
price	1.000000	-0.876590	-0.569960	0.314990	0.126389	0.185326	0.063104	0.219197	0.581198	-0.039536	0.054084	-0.038516
age	-0.876590	1.000000	0.505672	-0.156622	-0.098084	-0.148359	-0.005364	-0.198431	-0.470253	0.002389	-0.097740	0.092611
km	-0.569960	0.505672	1.000000	-0.333538	0.102683	-0.036197	0.015023	0.278165	-0.028598	0.144016	0.403060	-0.433160
hp	0.314990	-0.156622	-0.333538	1.000000	0.035856	0.092424	0.209477	-0.298432	0.089614	0.062109	-0.533453	0.489110
cc	0.126389	-0.098084	0.102683	0.035856	1.000000	0.079903	0.014629	0.306996	0.335637	0.005941	0.327723	-0.315170
doors	0.185326	-0.148359	-0.036197	0.092424	0.079903	1.000000	-0.160141	0.109363	0.302618	0.009680	0.025495	-0.027589
gears	0.063104	-0.005364	0.015023	0.209477	0.014629	-0.160141	1.000000	-0.005452	0.020613	-0.049537	-0.048847	0.063182
quarterly_tax	0.219197	-0.198431	0.278165	-0.298432	0.306996	0.109363	-0.005452	1.000000	0.626134	0.233791	0.792726	-0.835452
weight	0.581198	-0.470253	-0.028598	0.089614	0.335637	0.302618	0.020613	0.626134	1.000000	0.052756	0.568087	-0.560470
CNG	-0.039536	0.002389	0.144016	0.062109	0.005941	0.009680	-0.049537	0.233791	0.052756	1.000000	-0.038074	-0.296717
Diesel	0.054084	-0.097740	0.403060	-0.533453	0.327723	0.025495	-0.048847	0.792726	0.568087	-0.038074	1.000000	-0.942976
Petrol	-0.038516	0.092611	-0.433160	0.489110	-0.315170	-0.027589	0.063182	-0.835452	-0.560470	-0.296717	-0.942976	1.000000

Correlation heat map:



Skewness:

```
#Finding skewness
#-0.5 to 0.5:fairly symmetrical
#(-1 to 0.5) & (0.5 to 1):moderately skewed
#less than -1 or greater than 1:data are highly skewed
#Here all are fairly symmetrical
df.skew()
```

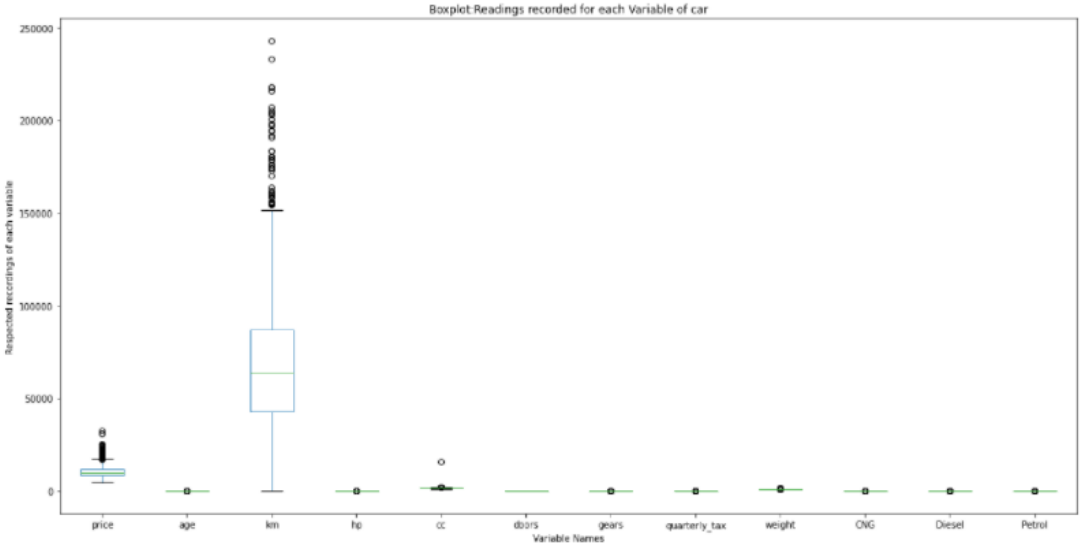
```
price             1.703885
age              -0.826702
km                1.015912
hp                0.955836
cc               27.431793
doors            -0.076395
gears             2.283960
quarterly_tax     1.993834
weight            3.108639
CNG               9.036211
Diesel            2.529601
Petrol           -2.344439
dtype: float64
```

Kurtosis:

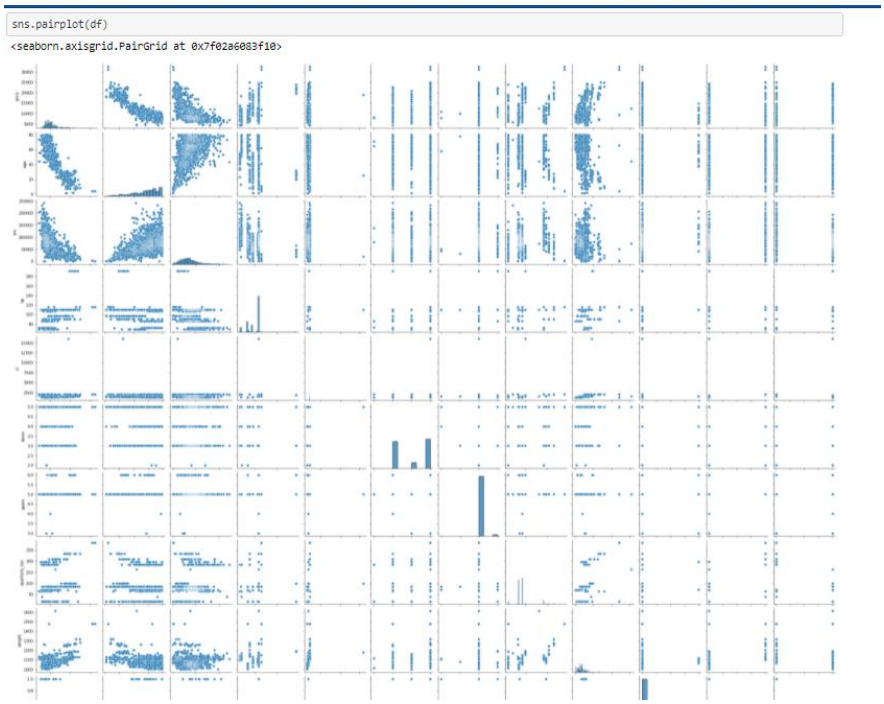
```
#Finding Kurtosis
#3:mesokurtic (std tail and zero excess distribution)
#<3:platykurtic (thin tail negative excess distribution)
#>3:leptokurtic (fat tailed and positive excess distribution)
df.kurt()
```

```
price              3.737781
age               -0.076632
km                 1.685057
hp                 8.836434
cc               930.711227
doors             -1.874765
gears             37.703476
quarterly_tax      4.298345
weight            19.362901
CNG               79.764204
Diesel             4.405014
Petrol             3.501268
dtype: float64
```

Boxplots:



Pairplots:

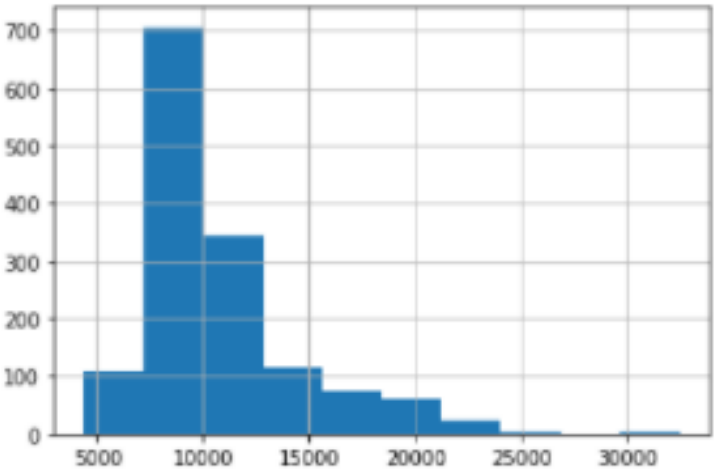


Determining Normality

Making our target variable normally distributed often will lead to better results If our target is not normally distributed, we can apply a transformation to it and then fit our regression to predict the transformed values.

How can we tell if our target is normally distributed? There are two ways:

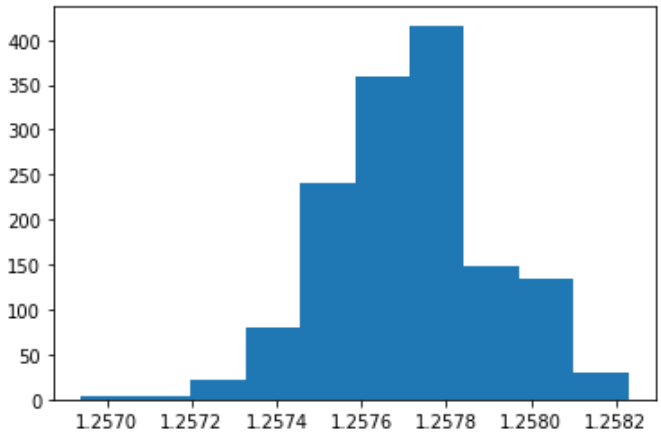
- Visually
- Using a statistical test



Definitely doesn't look normal due to long tail on the right side. And also from above skewness and kurtosis we can say that price is higly skewed and has excess distribution.
Lets test statistically

- This test outputs a "p-value". The higher this p-value is the closer the distribution is to normal.
- Frequentist statisticians would say that you accept that the distribution is normal (morespecifically: fail to reject the null hypothesis that it is normal) if p > 0.05.

Pvalue with normal test gave us 4.503464013882672e-106 which is not normally distributed.
Then we transformed our target variable with the help of log, square root and boxcox.
Boxcox gave us the best result with normalization:



Modelling:
We used train test split with considering 20% of the data as test size.
We used different modelling techniques namely Linear regression with standard scaling and normalized target variable, Linear regression without scaling, Linear regression using polynomial features with normalized target, Regularization techniques with polynomial features (Ridge regression and Lasso regression with alpha = 0.001).

Results:
Results of respective modelling techniques is presented as a table below:

Model	r2 Score
Linear ss	0.76
Linear	-9.28
Linear pf	0.89
Ridge pf	0.89
Lasso pf	-0.04

We can now conclude this project with the help of above result that linear regression with polynomial features and ridge regression with polynomial features has 89%. Thus we can say linear regression with polynomial features gives a fair prediction and further can be used to predict the prices of Toyota corolla car prices in future.