A

Project Report

“**Predicting car price using regression techniques**”

Submitted to

Department Of Information Technology Of

BUNTS SANGHA’S

S.M SHETTY COLLEGE OF SCIENCE, COMMERCE AND MANAGEMENT STUDIES

Hiranandani Garden, Powai, Mumbai-400 076

SUBMITTED BY

MISHRA DEEPAK SURYABHAN

Seat No. 4103729

On 2019-20

Under the Guidance of

Prof. Dr. Tushar Sambare

Submitted in the partial fulfilment of the requirement for qualifying

MSC (I.T.) Semester–IV Examination

BUNTS SANGHA’S

S.M SHETTY COLLEGE OF SCIENCE, COMMERCE AND MANAGEMENT STUDIES

Hiranandani Garden, Powai, Mumbai-400 076

Project Certificate

This is certified that the project Titled

“**Predicting car price using regression techniques**”

undertaken at the

Bunts Sangha’s S.M. Shetty College of Commerce & Management Studies

Mr. MISHRA DEEPAK SURYABHAN

of M.sc (IT), Part 2

Seat No. 4103729

In partial fulfilment of M.Sc. (IT) degree (Semester IV) Examination.

It is the certified that he has completed all the required phases of the project.

Internal Examiner External Examiner

Coordinator Principal

Acknowledgements

We are using this opportunity to express my gratitude to everyone who supported us throughout this project. We are thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work.

I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

My Sincere thanks to Prof. Dr. Tushar Sambare (Head of IT Department), for providing me the necessary administrative assistance in the completion of the work.

All the thanks are, however, only a fraction of what is due to Almighty for granting me the opportunity and the divine grace to successfully accomplish this assignment.

We certify that the work done by us for conceptualizing and completing this project is original and authentic.

Date: September 8 , 2020 **Deepak Mishra**

Place: Mumbai

INDEX

Abstract

Determining whether the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle's price on the market.

The focus of this project is developing machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases. The predictions are based on Automobile Dataset from an online source, and it is in CSV format.

We implement and evaluate various machine learning methods on a dataset consisting of the sale prices of different makes and models.

Different techniques like linear regression , multiple linear regression and Polynomial Regression have been used to make the predictions. The predictions are then evaluated and compared in order to find those which provide the best performances.

Our results show that all the three methods provided comparable performance. In the future, we intend to use more sophisticated algorithms to make the predictions.

CHAPTER 1: INTRODUCTION

Car price prediction is somehow interesting and popular problem. As per information that was gotten from the Agency for Statistics of BiH, 921.456 vehicles were registered in 2014 from which 84% of them are cars for personal usage. This number is increased by 2.7% since 2013 and it is likely that this trend will continue, and the number of cars will increase in future. This adds additional significance to the problem of the car price prediction.

Accurate car price prediction involves expert knowledge, because price usually depends on many distinctive features and factors. Typically, most significant ones are brand and model, age, horsepower and mileage. The fuel type used in the car as well as fuel consumption per mile highly affect price of a car due to a frequent changes in the price of a fuel. Different features like exterior color, door number, type of transmission, dimensions, safety, air condition, interior, whether it has navigation or not will also influence the car price. In this project, we applied different methods and techniques in order to achieve higher precision of the used car price prediction.

1.1 Motivation

Deciding whether a used car is worth the posted price when you see listings online can be difficult.

Several factors, including mileage, make, model, year, etc. can influence the actual worth of a car. From the perspective of a seller, it is also a dilemma to price a used car appropriately. Based on existing data, the aim is to use machine learning algorithms to develop models for predicting used car prices.

1.2 Dataset

For this project, we are using the Automobile Dataset from an online data source, and it is in CSV (comma separated value) format.

data source: https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data

data type: csv

The features available in this dataset are :- symboling, normalized-losses, make, fuel-type, aspiration, num-of-doors, body-style, drive-wheels, engine-location, wheel-base, length, width, height, curb-weight, engine-type, num-of-cylinders, engine-size, fuel-system, bore, stroke, compression-ratio, horsepower, peak-rpm, city-mpg, highway-mpg and price.

1.3 Tools & Techniques

Tools:

Python for model development using libraries like pandas, numpy, matplotlib, seaborn, scikitlearn, imblearn etc.

Techniques:

Histograms, bar plots and density plots for data visualization and feature selection.

Scatter plots and heat maps to detect correlation and tune feature selection.

In-built algorithms to develop predictive models and cross-validation for evaluation.

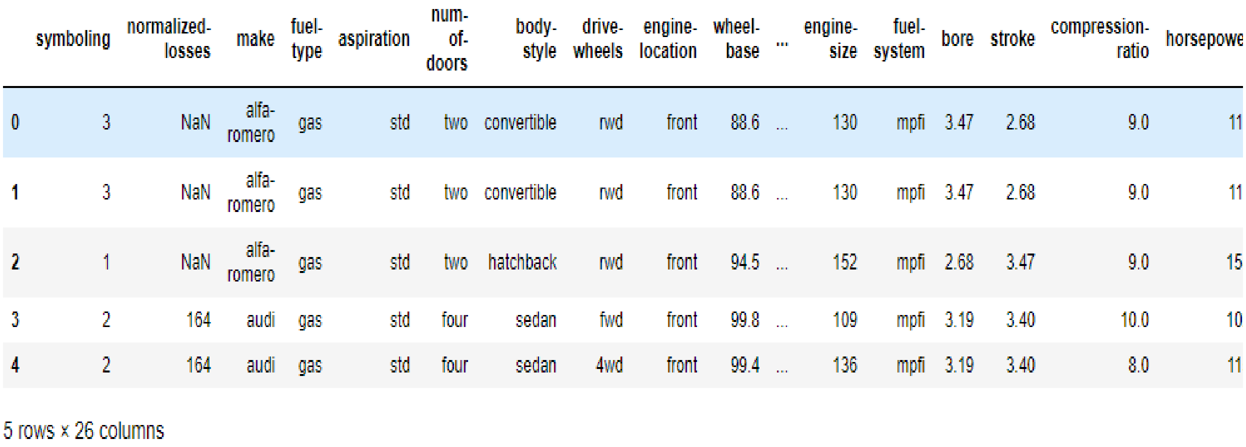
Chapter 2: Data Pre-Processing

In order to get a better understanding of the data, we plotted a histogram of the data. We noticed that the dataset had many outliers and missing data primarily due to large price sensitivity of used cars. Typically, models that are the latest year and have low mileage sell for a premium, however, there were many data points that did not conform to this. This is because accident history and condition can have a significant effect on the car's price. Since we did not have access to vehicle history and condition, we pruned our dataset to three standard deviations around the mean in order to remove outliers.

We converted the Make, Model and State into one-hot vectors.

2.1 Identify and handle missing values

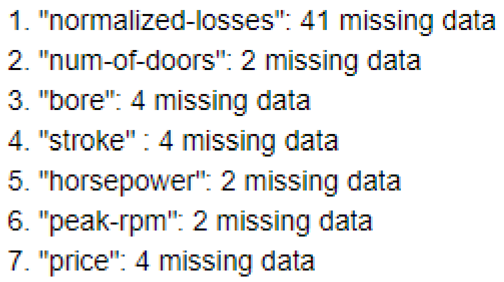
In the car dataset, missing data comes with the question mark "?". We replace "?" with NaN (Not a Number), which is Python's default missing value marker, for reasons of computational speed and convenience.



	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	engine-size	fuel-system	bore	stroke	compression-ratio	horsepowe
0	3	NaN	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68	9.0	11
1	3	NaN	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68	9.0	11
2	1	NaN	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	...	152	mpfi	2.68	3.47	9.0	15
3	2	164	audi	gas	std	four	sedan	fwd	front	99.8	...	109	mpfi	3.19	3.40	10.0	10
4	2	164	audi	gas	std	four	sedan	4wd	front	99.4	...	136	mpfi	3.19	3.40	8.0	11

5 rows × 26 columns

Based on the summary above, each column has 205 rows of data, seven columns containing missing data -



1. "normalized-losses": 41 missing data
2. "num-of-doors": 2 missing data
3. "bore": 4 missing data
4. "stroke" : 4 missing data
5. "horsepower": 2 missing data
6. "peak-rpm": 2 missing data
7. "price": 4 missing data

2.2 Data Standardization

Standardization is the process of transforming data into a common format which allows the researcher to make the meaningful comparison.

Data is usually collected from different agencies with different formats. (Data Standardization is also a term for a particular type of data normalization, where we subtract the mean and divide by the standard deviation)

Example

Transform mpg to L/100km:

In our dataset, the fuel consumption columns "city-mpg" and "highway-mpg" are represented by mpg (miles per gallon) unit. Assume we are developing an application in a country that accept the fuel consumption with L/100km standard

We will need to apply data transformation to transform mpg into L/100km?

The formula for unit conversion is

L/100km = 235 / mpg

2.3 Data Normalization

Normalization is the process of transforming values of several variables into a similar range. Typical normalizations include scaling the variable so the variable average is 0, scaling the variable so the variance is 1, or scaling variable so the variable values range from 0 to 1

Example

To demonstrate normalization, let's say we want to scale the columns "length", "width" and "height"

Target: would like to Normalize those variables so their value ranges from 0 to 1.

Approach: replace original value by (original value) / (maximum value)

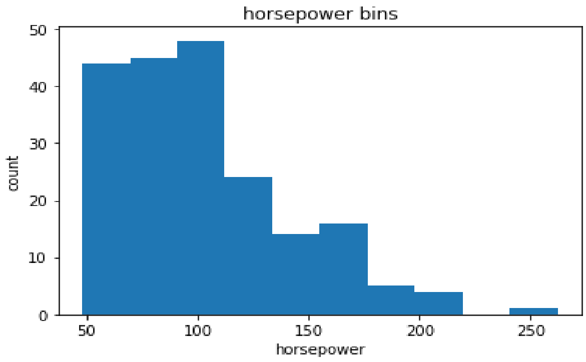
2.4 Binning

Binning is a process of transforming continuous numerical variables into discrete categorical 'bins', for grouped analysis.

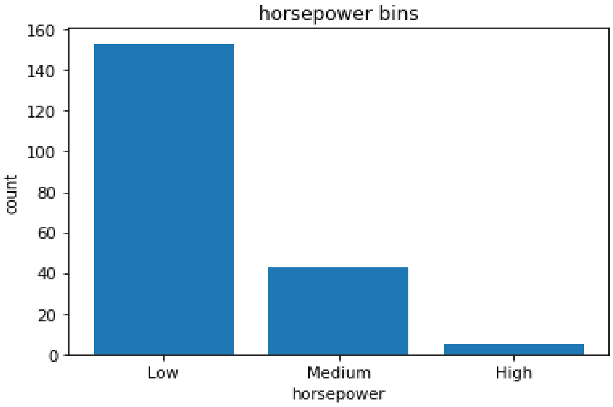
Example:

In our dataset, "horsepower" is a real valued variable ranging from 48 to 288, it has 57 unique values.

We will use the Pandas method 'cut' to segment the 'horsepower' column into 3 bins-



Before



After

2.5 Feature Engineering

After treating for missing values and outliers, it was discovered that some of the columns did not have enough variation to be meaningful.

So we further removed such columns to create a trimmed down dataset of only 10 features.

Table : Selected Features

Feature	Status	Reasons
symboling	Reject	Not useful
normalized-losses	Reject	Not useful
make	Reject	Not useful
fuel-type	Reject	Not useful
aspiration	Reject	Not useful
num-of-doors	Reject	Not useful
body-style	Reject	Not useful
drive-wheels	Accept	
engine-location	Reject	Not useful
wheel-base	Accept	
length	Accept	
width	Accept	
height	Reject	Not useful
curb-weight	Accept	
engine-type	Reject	Not useful
num-of-cylinders	Reject	Not useful

engine-size	Accept	
fuel-system	Reject	Not useful
bore	Accept	
stroke	Reject	Not useful
compression-ratio	Reject	Not useful
horsepower	Accept	
peak-rpm	Reject	Not useful
city-mpg	Accept	
highway-mpg	Accept	
price	Accept	

CHAPTER 3: EXPLORATORY DATA ANALYSIS

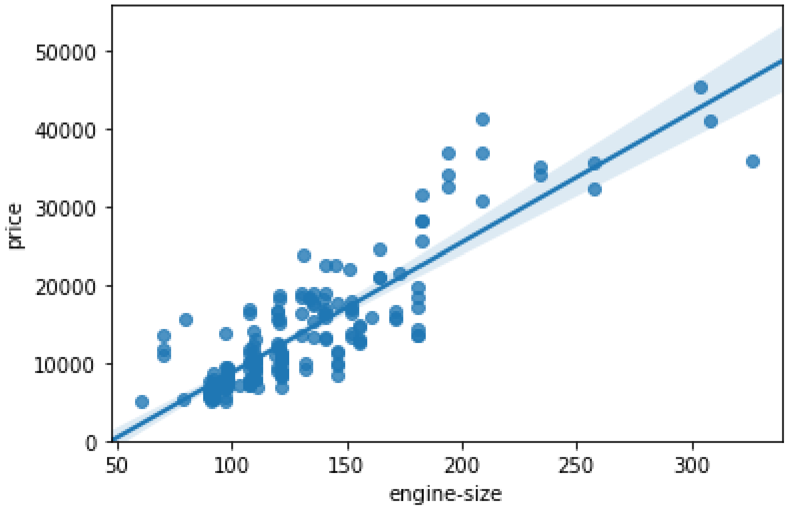
Before building the model on the cleaned data, some exploratory analysis needs to be done which would help us in the future to fine-tune and evaluate out model.

At this point, however, the dataset should be separated into training and testing sets. The testing data is meant to evaluate how well generalized the model is – that is, whether it works just as well in practical application as it does in theory. Any insights generated here will be used to build the model and if they are generated on testing data as well, it'd cause the model to be overfitted.

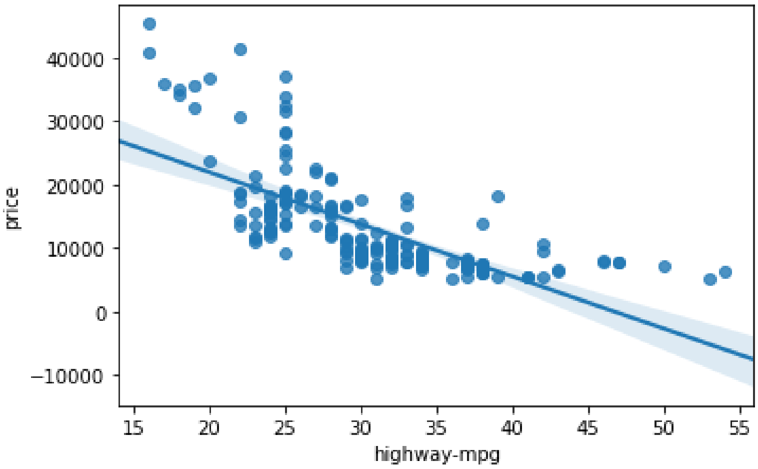
3.1 Analyzing Individual Feature Patterns

3.1.1 Continuous numerical variables:

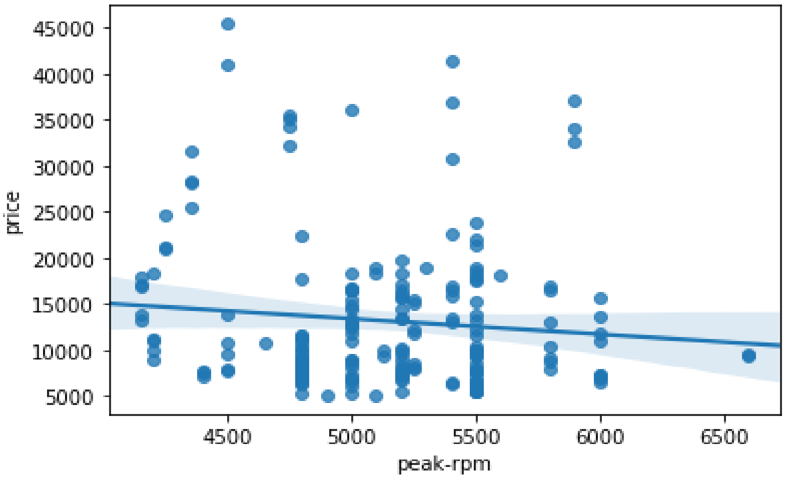
In order to start understanding the (linear) relationship between an individual variable and the price. We can do this by using "regplot", which plots the scatterplot plus the fitted regression line for the data.



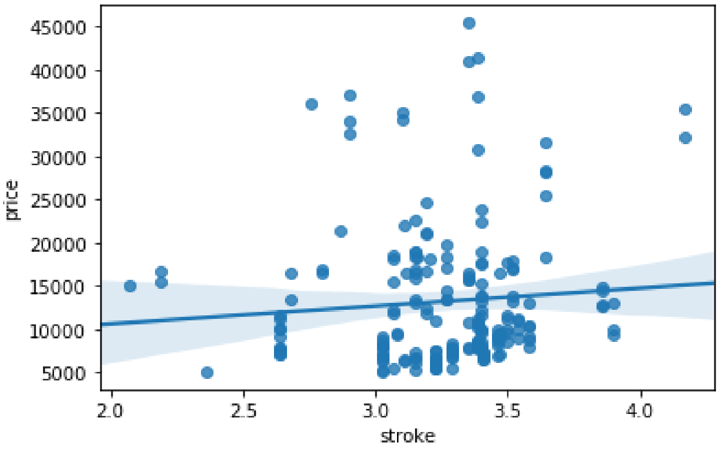
engine-size vs price



highway-mpg vs price



Peak-rpm vs price

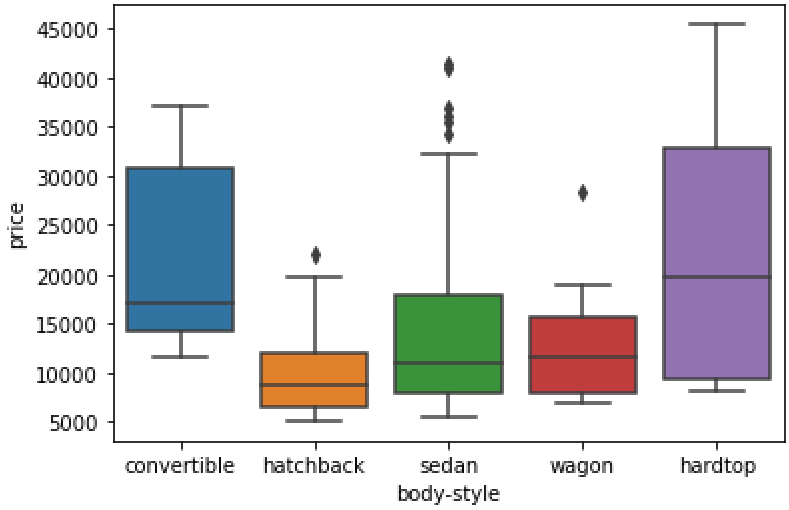


Stroke vs price

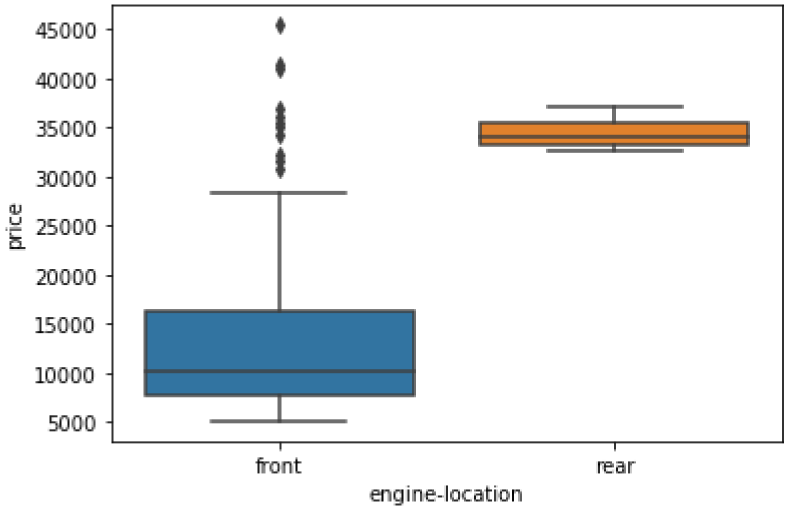
3.1.2 Categorical variables

Categorical variables that describe a 'characteristic' of a data unit, and are selected from a small group of categories. The categorical variables can have the type "object" or "int64". A good way to visualize categorical variables is by using boxplots.

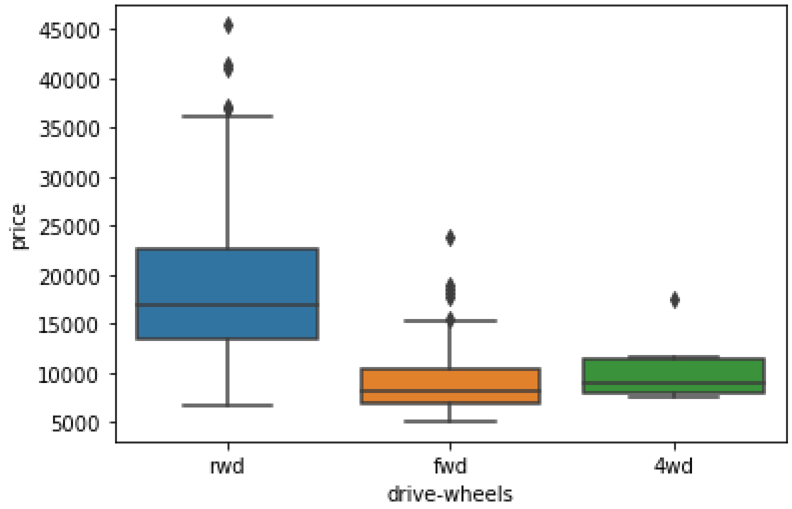
body-style vs price



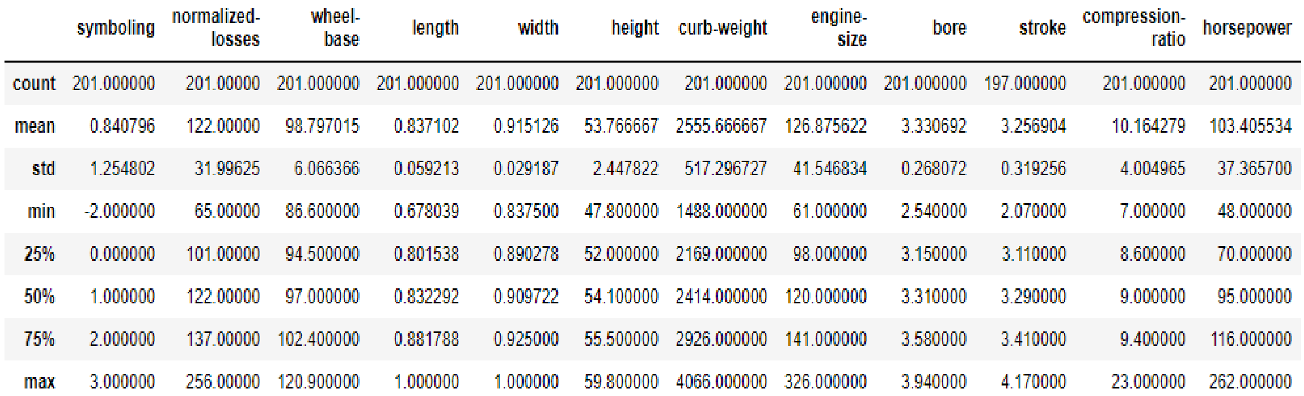
engine-location vs price



drive-wheels vs price



3.2 Descriptive Statistical Analysis



	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke	compression-ratio	horsepower
count	201.000000	201.00000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	197.000000	201.000000	201.000000
mean	0.840796	122.00000	98.797015	0.837102	0.915126	53.766667	2555.666667	126.875622	3.330692	3.256904	10.164279	103.405534
std	1.254802	31.99625	6.066366	0.059213	0.029187	2.447822	517.296727	41.546834	0.268072	0.319256	4.004965	37.365700
min	-2.000000	65.00000	86.600000	0.678039	0.837500	47.800000	1488.000000	61.000000	2.540000	2.070000	7.000000	48.000000
25%	0.000000	101.00000	94.500000	0.801538	0.890278	52.000000	2169.000000	98.000000	3.150000	3.110000	8.600000	70.000000
50%	1.000000	122.00000	97.000000	0.832292	0.909722	54.100000	2414.000000	120.000000	3.310000	3.290000	9.000000	95.000000
75%	2.000000	137.00000	102.400000	0.881788	0.925000	55.500000	2926.000000	141.000000	3.580000	3.410000	9.400000	116.000000
max	3.000000	256.00000	120.900000	1.000000	1.000000	59.800000	4066.000000	326.000000	3.940000	4.170000	23.000000	262.000000

3.3 Correlation and Causation Analysis

Correlation: a measure of the extent of interdependence between variables.

Causation: the relationship between cause and effect between two variables.

It is important to know the difference between these two and that correlation does not imply causation. Determining correlation is much simpler the determining causation as causation may require independent experimentation.

Pearson Correlation

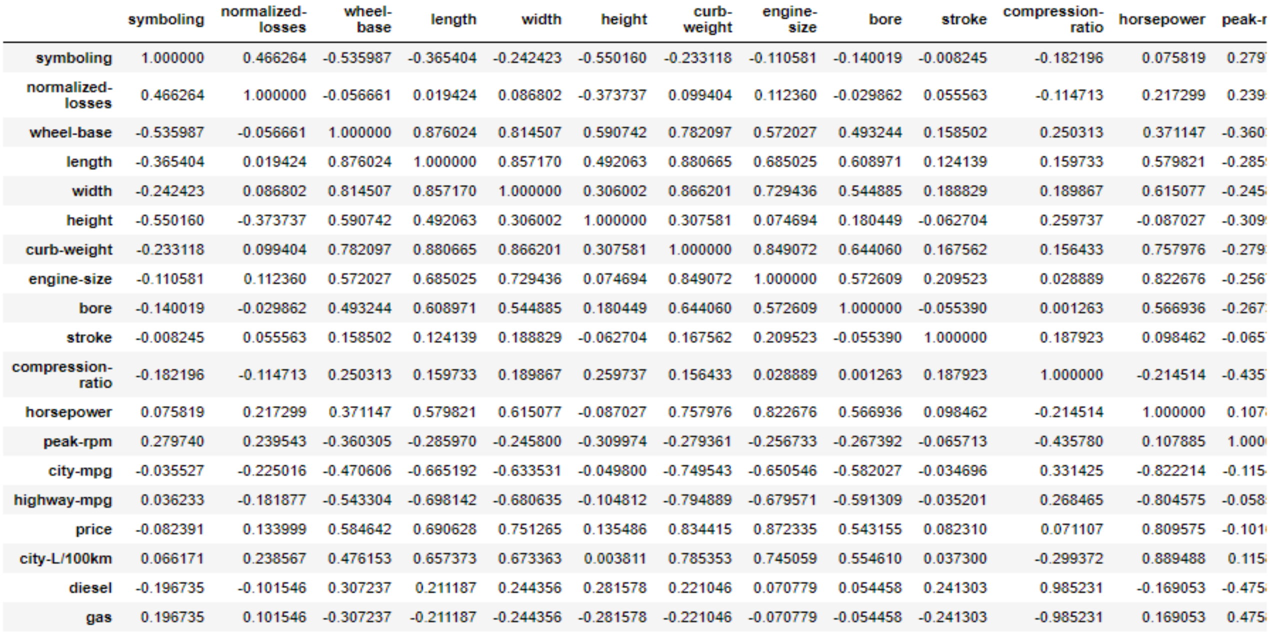
The Pearson Correlation measures the linear dependence between two variables X and Y.

The resulting coefficient is a value between -1 and 1 inclusive, where:

1: Total positive linear correlation.

0: No linear correlation, the two variables most likely do not affect each other.

-1: Total negative linear correlation.



	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke	compression-ratio	horsepower	peak-r
symboling	1.000000	0.466264	-0.535987	-0.365404	-0.242423	-0.550160	-0.233118	-0.110581	-0.140019	-0.008245	-0.182196	0.075819	0.279
normalized-losses	0.466264	1.000000	-0.056661	0.019424	0.086802	-0.373737	0.099404	0.112360	-0.029862	0.055563	-0.114713	0.217299	0.239
wheel-base	-0.535987	-0.056661	1.000000	0.876024	0.814507	0.590742	0.782097	0.572027	0.493244	0.158502	0.250313	0.371147	-0.360
length	-0.365404	0.019424	0.876024	1.000000	0.857170	0.492063	0.880665	0.685025	0.608971	0.124139	0.159733	0.579821	-0.285
width	-0.242423	0.086802	0.814507	0.857170	1.000000	0.306002	0.866201	0.729436	0.544885	0.188829	0.189867	0.615077	-0.245
height	-0.550160	-0.373737	0.590742	0.492063	0.306002	1.000000	0.307581	0.074694	0.180449	-0.062704	0.259737	-0.087027	-0.309
curb-weight	-0.233118	0.099404	0.782097	0.880665	0.866201	0.307581	1.000000	0.849072	0.644060	0.167562	0.156433	0.757976	-0.279
engine-size	-0.110581	0.112360	0.572027	0.685025	0.729436	0.074694	0.849072	1.000000	0.572609	0.209523	0.028889	0.822676	-0.256
bore	-0.140019	-0.029862	0.493244	0.608971	0.544885	0.180449	0.644060	0.572609	1.000000	-0.055390	0.001263	0.566936	-0.267
stroke	-0.008245	0.055563	0.158502	0.124139	0.188829	-0.062704	0.167562	0.209523	-0.055390	1.000000	0.187923	0.098462	-0.065
compression-ratio	-0.182196	-0.114713	0.250313	0.159733	0.189867	0.259737	0.156433	0.028889	0.001263	0.187923	1.000000	-0.214514	-0.435
horsepower	0.075819	0.217299	0.371147	0.579821	0.615077	-0.087027	0.757976	0.822676	0.566936	0.098462	-0.214514	1.000000	0.107
peak-rpm	0.279740	0.239543	-0.360305	-0.285970	-0.245800	-0.309974	-0.279361	-0.256733	-0.267392	-0.065713	-0.435780	0.107885	1.000
city-mpg	-0.035527	-0.225016	-0.470606	-0.665192	-0.633531	-0.049800	-0.749543	-0.650546	-0.582027	-0.034696	0.331425	-0.822214	-0.115
highway-mpg	0.036233	-0.181877	-0.543304	-0.698142	-0.680635	-0.104812	-0.794889	-0.679571	-0.591309	-0.035201	0.268465	-0.804575	-0.058
price	-0.082391	0.133999	0.584642	0.690628	0.751265	0.135486	0.834415	0.872335	0.543155	0.082310	0.071107	0.809575	-0.101
city-L/100km	0.066171	0.238567	0.476153	0.657373	0.673363	0.003811	0.785353	0.745059	0.554610	0.037300	-0.299372	0.889488	0.115
diesel	-0.196735	-0.101546	0.307237	0.211187	0.244356	0.281578	0.221046	0.070779	0.054458	0.241303	0.985231	-0.169053	-0.475
gas	0.196735	0.101546	-0.307237	-0.211187	-0.244356	-0.281578	-0.221046	-0.070779	-0.054458	-0.241303	-0.985231	0.169053	0.475

3.4 Important Variables

We now have a better idea of what our data looks like and which variables are important to take into account when predicting the car price. We have narrowed it down to the following variables:

3.4.1 Continuous numerical variables:

Length

Width

Curb-weight

Engine-size

Horsepower

City-mpg

Highway-mpg

Wheel-base

Bore

3.4.2 Categorical variables:

Drive-wheels

Chapter 4: Model Development

4.1 Simple Linear Regression.

Simple Linear Regression is a method to help us understand the relationship between two variables:

The predictor/independent variable (X)

The response/dependent variable (that we want to predict) (Y)

The result of Linear Regression is a linear function that predicts the response (dependent) variable as a function of the predictor (independent) variable.

Y:$Response\ Variable$

X:$Predictor\ Variables$

Linear function:

$Yhat=a+bX$Yhat=a+bX

a refers to the intercept of the regression line0, in other words: the value of Y when X is 0

b refers to the slope of the regression line, in other words: the value with which Y changes when X increases by 1 unit

4.2 Multiple Linear Regression

What if we want to predict car price using more than one variable?

If we want to use more variables in our model to predict car price, we can use Multiple Linear Regression. Multiple Linear Regression is very similar to Simple Linear Regression, but this method is used to explain the relationship between one continuous response (dependent) variable and two or more predictor (independent) variables. Most of the real-world regression models involve multiple predictors. We will illustrate the structure by using four predictor variables, but these results can generalize to any integer:

Y:$Response\ Variable$

$X1$:$Predictor\ Variable$ 1

$X2$:$Predictor\ Variable$ 2

$X3$:$Predictor\ Variable$ 3

$X4$:$Predictor\ Variable$ 4

a:$intercept$

$b1$:$coefficients\ of\ Variable$ 1

$b2$:$coefficients\ of\ Variable$ 2

$b3$:$coefficients\ of\ Variable$ 3

$b4$:$coefficients\ of\ Variable$ 4

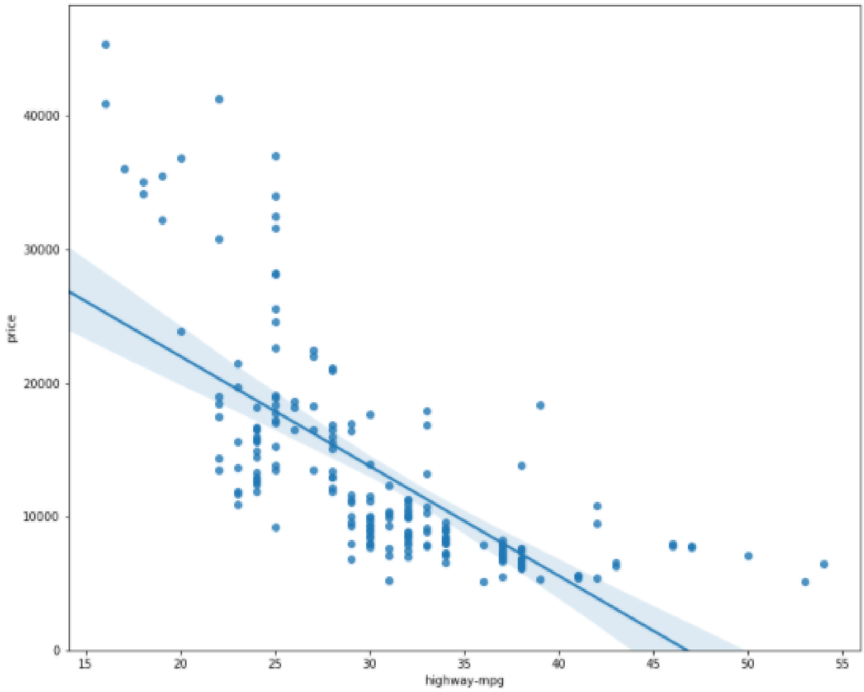
The equation is given by

$\boldsymbol{Yhat=a+b1X1+b2X2+b3X3+b4X4}$

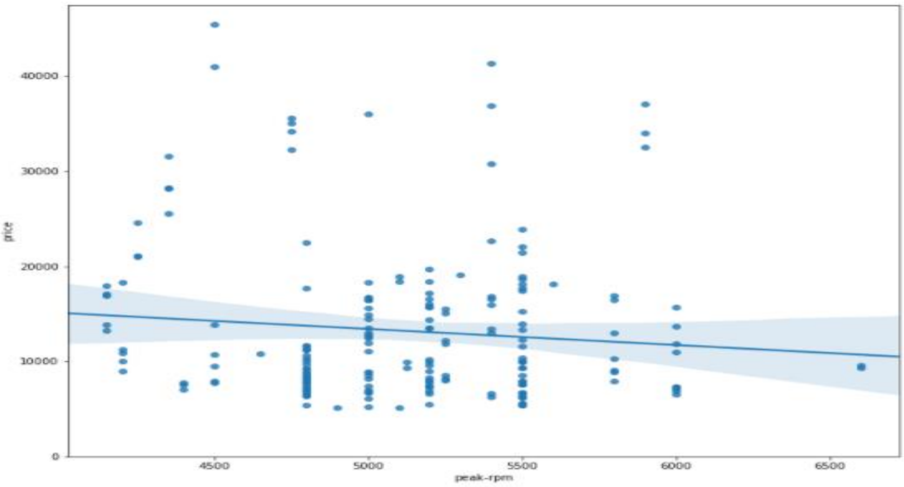
Model Evaluation using Visualization

Simple Linear Regression

Horsepower as potential predictor variable of price

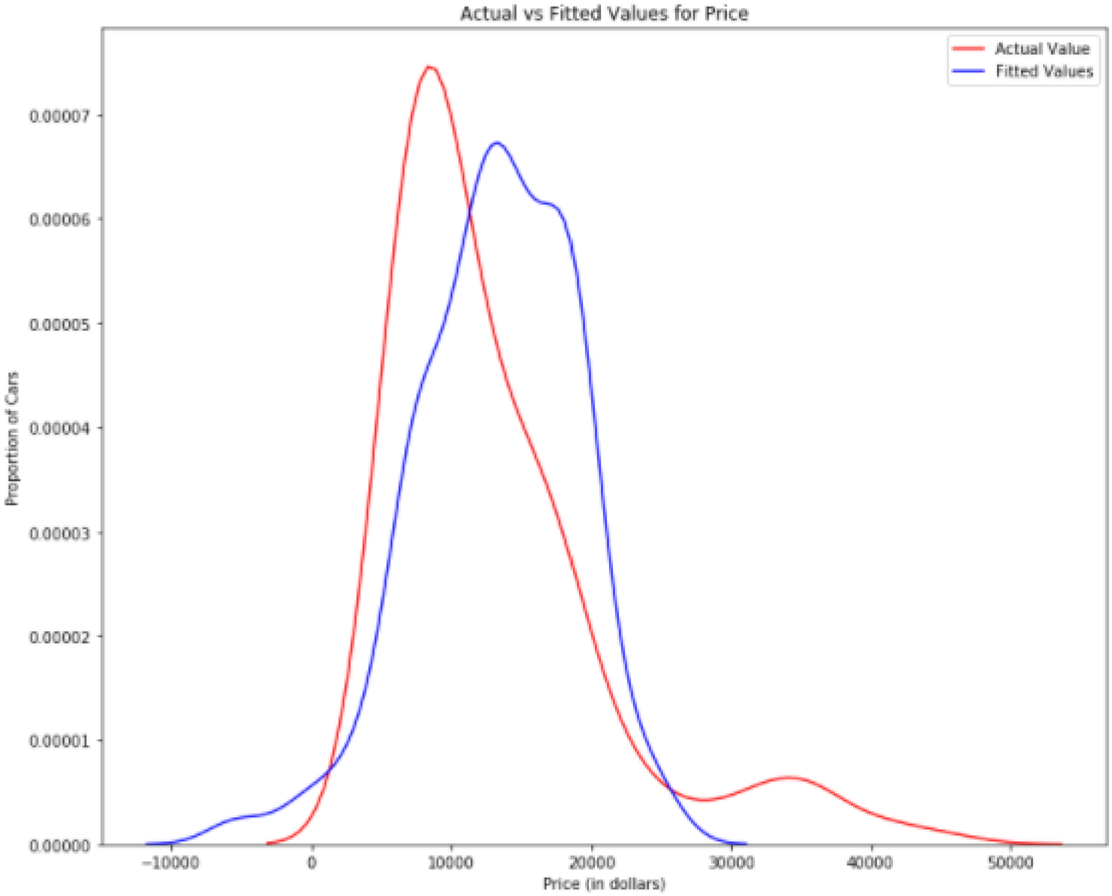


peak-rpm as potential predictor variable of price



Multiple Linear Regression

One way to look at the fit of the model is by looking at the distribution plot: We can look at the distribution of the fitted values that result from the model and compare it to the distribution of the actual values.



Polynomial Regression and Pipelines

Polynomial regression is a particular case of the general linear regression model or multiple linear regression models.

We get non-linear relationships by squaring or setting higher-order terms of the predictor variables.

There are different orders of polynomial regression:

Quadratic - 2nd order

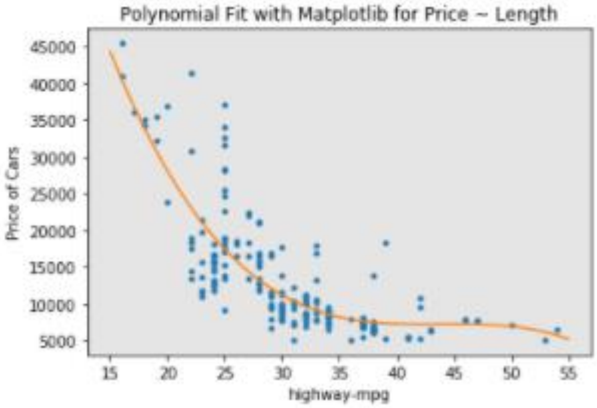
$Yhat=a+b1X2+b2X2$Yhat=a+b1X2+b2X2

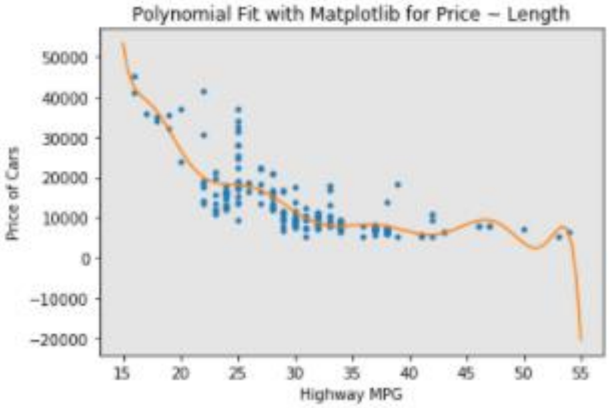
Cubic - 3rd order

$Yhat=a+b1X2+b2X2+b3X3$Yhat=a+b1X2+b2X2+b3X3

Higher order:

$Y=a+b1X2+b2X2+b3X3$....





Measures for In-Sample Evaluation

Two very important measures that are often used in Statistics to determine the accuracy of a model are:

R^2 / R-squared

Mean Squared Error (MSE)

R-squared

R squared, also known as the coefficient of determination, is a measure to indicate how close the data is to the fitted regression line.

The value of the R-squared is the percentage of variation of the response variable (y) that is explained by a linear model.

Mean Squared Error (MSE)

The Mean Squared Error measures the average of the squares of errors, that is, the difference between actual value (y) and the estimated value (ŷ).

Model	R-square	MSE
Simple Linear Regression	49.659%	31635042.944639888
Multiple Linear Regression	80.896 %	11980366.87072649
Polynomial Fit	67.419 %	20474146.426361218

Prediction and Decision Making

Model	R-square	MSE
Simple Linear Regression	49.659%	31635042.944639888
Multiple Linear Regression	80.896 %	11980366.87072649
Polynomial Fit	67.419 %	20474146.426361218

Decision Making: Determining a Good Model Fit

Now that we have visualized the different models, and generated the R-squared and MSE values for the fits, how do we determine a good model fit?

What is a good R-squared value?

When comparing models, the model with the higher R-squared value is a better fit for the data.

What is a good MSE?

When comparing models, the model with the smallest MSE value is a better fit for the data.

Let's take a look at the values for the different models.

Simple Linear Regression: Using Highway-mpg as a Predictor Variable of Price.

R-squared: 0.49659118843391759

MSE: 3.16 x10^7

Multiple Linear Regression: Using Horsepower, Curb-weight, Engine-size, and Highway-mpg as Predictor Variables of Price.

R-squared: 0.80896354913783497

MSE: 1.2 x10^7

Polynomial Fit: Using Highway-mpg as a Predictor Variable of Price.

R-squared: 0.6741946663906514

MSE: 2.05 x 10^7

Simple Linear Regression model (SLR) vs Multiple Linear Regression model (MLR)

Usually, the more variables you have, the better your model is at predicting, but this is not always true. Sometimes you may not have enough data, you may run into numerical problems, or many of the variables may not be useful and or even act as noise. As a result, you should always check the MSE and R^2.

So to be able to compare the results of the MLR vs SLR models, we look at a combination of both the R-squared and MSE to make the best conclusion about the fit of the model.

MSEThe MSE of SLR is 3.16x10^7 while MLR has an MSE of 1.2 x10^7. The MSE of MLR is much smaller.

R-squared: In this case, we can also see that there is a big difference between the R-squared of the SLR and the R-squared of the MLR. The R-squared for the SLR (~0.497) is very small compared to the R-squared for the MLR (~0.809).

This R-squared in combination with the MSE show that MLR seems like the better model fit in this case, compared to SLR.

Simple Linear Model (SLR) vs Polynomial Fit

MSE: We can see that Polynomial Fit brought down the MSE, since this MSE is smaller than the one from the SLR.

R-squared: The R-squared for the Polyfit is larger than the R-squared for the SLR, so the Polynomial Fit also brought up the R-squared quite a bit.

Since the Polynomial Fit resulted in a lower MSE and a higher R-squared, we can conclude that this was a better fit model than the simple linear regression for predicting Price with Highway-mpg as a predictor variable.

Multiple Linear Regression (MLR) vs Polynomial Fit

MSE: The MSE for the MLR is smaller than the MSE for the Polynomial Fit.

R-squared: The R-squared for the MLR is also much larger than for the Polynomial Fit.

Conclusion:

Comparing these three models, we conclude that the MLR model is the best model to be able to predict price from our dataset. This result makes sense, since we have 27 variables in total, and we know that more than one of those variables are potential predictors of the final car price.

The main limitation of this study is the

low number of records that have been used. As future work, we intend to collect more

data and to use more advanced techniques like artificial neural networks, fuzzy logic

and genetic algorithms to predict car prices.

Chapter 5: Model Evaluation

Training and Testing

Now we randomly split our data into training and testing data using the function train_test_split.

number of test samples : 81

number of training samples: 120

Accuracy results of traning and testing data

Data	R^2
test data	0.707688374146705
Traning data	0.6449517437659684

Cross-validation Score

Sometimes we do not have sufficient testing data; as a result, you may want to perform Cross-validation.

We input the object, the feature in this case ' horsepower', the target data (y_data). The parameter 'cv' determines the number of folds; in this case 4.

Accuracy results

Type of evaluation	
Cross validation with 4 folds	77.46
Cross validation with 2 folds	0.51667

Overfitting, Underfitting and Model Selection

Overfitting

Overfitting occurs when the model fits the noise, not the underlying process. Therefore when testing your model using the test-set, your model does not perform as well as it is modelling noise, not the underlying process that generated the relationship.

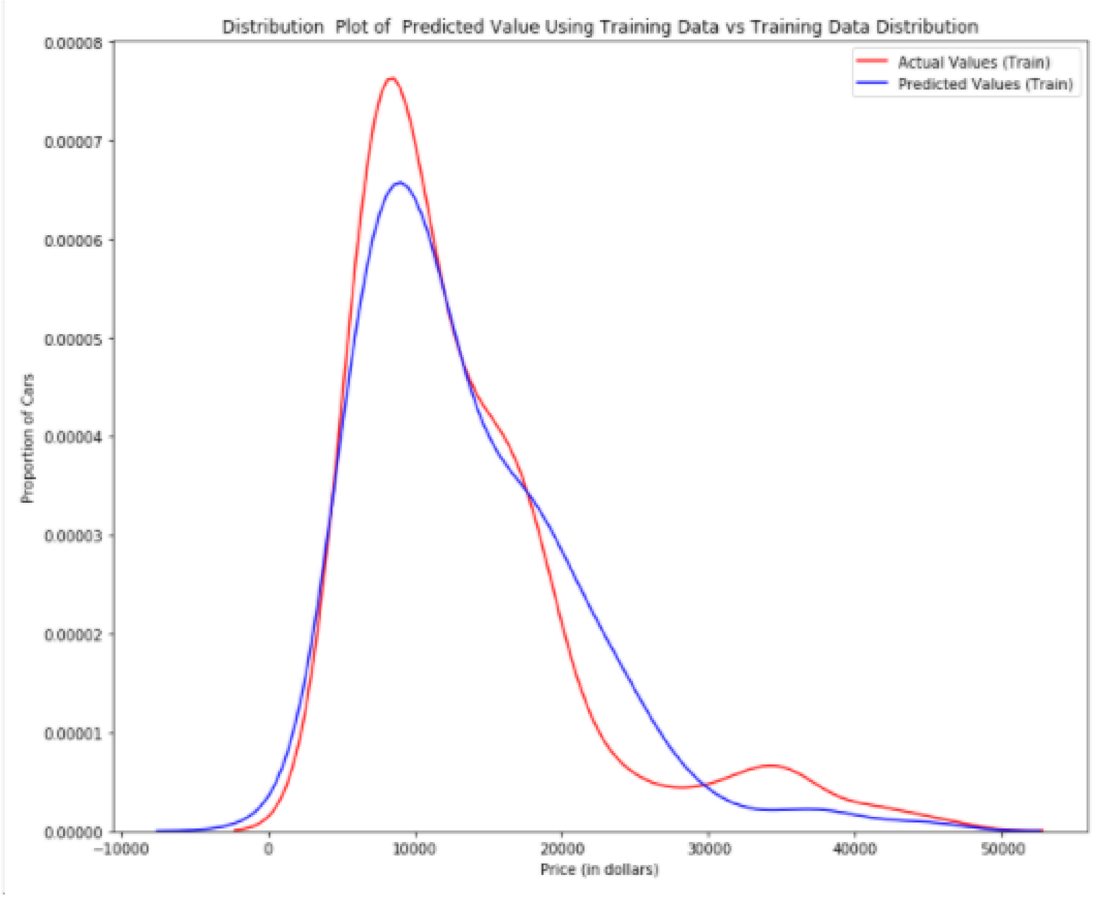
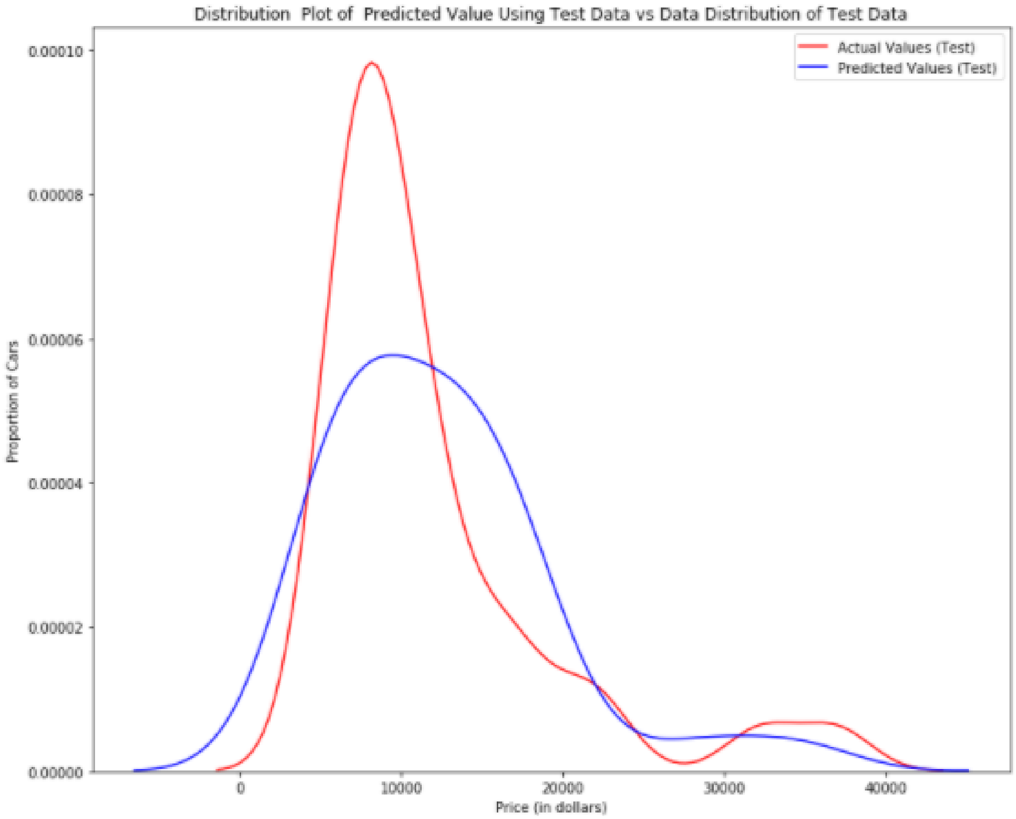


Figure 1: Plot of predicted values using the training data compared to the training data.



Figur 2: Plot of predicted value using the test data compared to the test data.

Comparing Figure 1 and Figure 2; it is evident the distribution of the test data in Figure 1 is much better at fitting the data. This difference in Figure 2 is apparent where the ranges are from 5000 to 15 000. This is where the distribution shape is exceptionally different.

Conclusion

Car price prediction can be a challenging task due to the high number of attributes that should be considered for the accurate prediction. The major step in the prediction process is collection and pre-processing of the data. In this project we were built to normalize, standardize and clean data to avoid unnecessary noise for machine learning algorithms.