A Project Report On

“Car MVP: Most Viable Predictions”

With special reference to

“Linear Regression Machine Learning”

Project Report submitted to T.I.E. & I.C. Solutions

In fulfillment of the requirements for the internship of

MACHINE LEARNING INTERNSHIP

Submitted By

Mr. ASHUTOSH

1AT17CS011



Under the Guidance of

Mr. ABHISHEK C

ATRIA INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BENGALURU

Acknowledgement

I ASHUTOSH from Atria Institute of Technology bearing USN 1AT17CS011 take full responsibility of this project on Linear Regression of Car MVP. This work has been done by me from September 2020 to October 2020. The Project has been completed within the stipulated time frame and submitted to my internship instructor.

I would also like to present my heartfelt thanks to Mr. Abhishek C my instructor for the internship, He helped me a lot during this period on my project and cleared my doubts. A significant part of share goes to him for the success of this project.

Also, I would thank Take It Easy Engineers and I C Solutions for conducting such a beneficial internship and such a nice training I feel more confident about myself and my work now.

Huge Thanks and appreciation to you all.

NAME: ASHUTOSH

USN: 1AT17CS011

DATE: 19TH October 20

PLACE: MUZAFFARPUR

Abstract

Cars are an important and vital means of transport. Approximately every other person uses cars for their journey. This has led to a lot of new companies to produce much cars and people are utilizing its benefits but This significant amount of increase in car production is the reason which gave birth to a new problem the unfair pricing and more marginalization of car prices. So to curb that I made a Machine learning model which helps in predicting the fair price of a car using all the features of the cars and gives the customer the amount what they need to shed if they are going to buy a new car for themselves next time. Also, it could be used by the companies to set a right margin for their cars. This model works on Linear regression Algorithm and provides a value which can used. It takes cares of the features and inputs and the target variable ‘price’ is predicted accordingly.

About the company

I.C. Solutions –

ICS is a digital service provider that aims to provide software, designing and marketing solutions to individuals and businesses. At ICS, we believe that service and quality is the key to success.We provide all kinds of technological and designing solutions from Billing Software to Web Designs or any custom demand that you may have. Experience the service like none other!

Some of our services include:

Development - We develop responsive, functional and super fast websites. We keep User Experience in mind while creating websites. A website should load quickly and should be accessible even on a small view-port and slow internet connection.

Mobile Application - We offer a wide range of professional Android, iOS & Hybrid app development services for our global clients, from a startup to a large enterprise.

Design - We offer professional Graphic design, Brochure design & Logo design. We are experts in crafting visual content to convey the right message to the customers.

Consultancy - We are here to provide you with expert advice on your design and development requirement.

Videos - We create a polished professional video that impresses your audience.

image.

INDEX

Introduction………………………………………………………….06

Problem Statement and Objective…………………………………...07

Requirement Specification……………………………………….….08

Exploratory Data Analysis……………………………………….….[09](#EDA)

Preparing Machine Learning Model……………………...…………[17](#model)

ML model Chart………………………………………………….….[21](#chart)

Hurdles………………………………………………………………[22](#hurdle)

Conclusion……………………………………………………….….[25](#Conclusion)

Bibliography………………………………………………………...[26](#bib)

(Note: for ease in movement-

All the page numbers are hyperlinked in the document by clicking on the desired page number it takes to the targeted heading.

All the headings are hyperlinked on clicking of the heading it will take to the index.)

Introduction

Amidst in COVID-19 situation it was tough to get an internship and being my domain as Machine Learning it was tougher, I wanted to build something which can be useful for a community and thus looked at a common problem which affects the people day to day life or matter as a whole. Cars being an important part helped me to do something in this direction. This internship is completed under guidance of Mr. Abhishek Sir from TIE and with IC solutions. The internship comprises of an initial training and a project for which I chose this problem statement.

This work is carried by me from an artificially generated dataset and with the exact important properties a car use to have. The accuracy has been taken care of and the details was very minutely checked. As of now this is in the initial phase and can be taken for production after its beta staging.

(this project is also hosted on [Github](https://github.com/Aashutosh748), you can find it here <https://github.com/Aashutosh748/carmvp> )

Problem Statement –

Cars are an important and vital means of transport. Approximately every other person uses cars for their journey. This has led to a lot of new companies to produce much cars and people are utilizing its benefits but This significant amount of increase in car production is the reason which gave birth to a new problem the unfair pricing and more marginalization of car prices.

Objective –

To curb that I made a Machine learning model which helps in predicting the fair price of a car using all the features of the cars and gives the customer the amount what they need to shed if they are going to buy a new car for themselves next time. Also, it could be used by the companies to set a right margin for their cars. This model works on Linear regression Algorithm and provides a value which can used. It takes cares of the features and inputs and the target variable ‘price’ is predicted accordingly.

Requirement Specification

Hardware requirements

* CPU - Intel i7 8th Gen processor
* GPU - Nvidia Geforce 2080
* RAM – 8GB and onwards
* Storage – 48GB and onwards

Software requirements

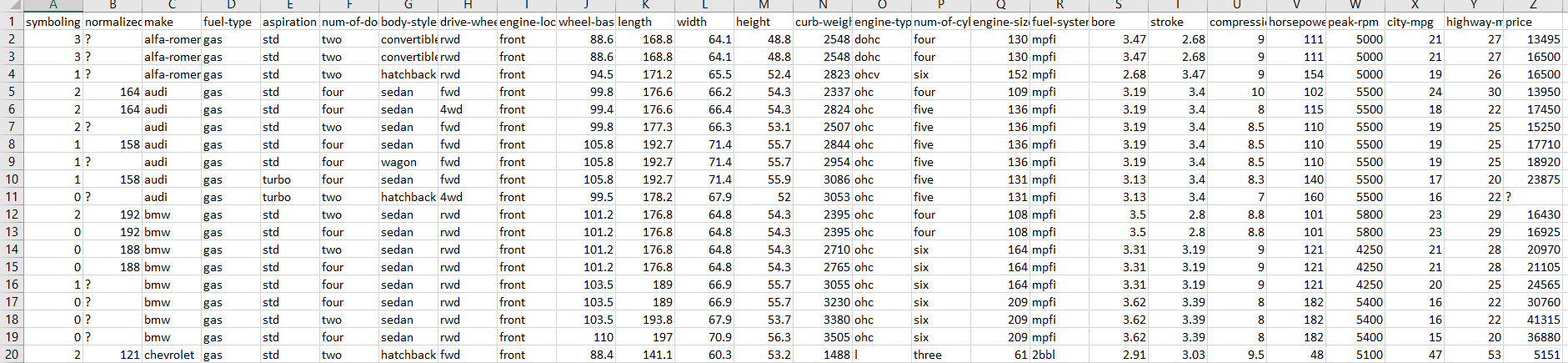
* Windows 8 and onwards
* Tensorflow
* Anaconda IDE
* Jupyter Notebook
* Numpy, Pandas, MatplotLib, Seaborn Libaries
* Code Editor like Visual Studio

[Exploratory Data Analysis](#index)

1. Finding the Dataset

This was the most important aspect of the project as Datasets are very important and much required. The first step was to get the correct dataset which not only has the proper values but also meaningful attributes.

So, the data which I got my hands on was fulfilling the necessity. Mentioned in the project as ‘Automobile price data \_Raw\_.csv’.



Here we can see the useful attributes with our target variable.

This graph gives details about

symboling

normalized-losses

make

fuel-type

aspiration

num-of-doors

body-style

drive-wheels

wheel-base

length

width

height

curb-weight

engine-type

num-of-cylinders

engine-size

fuel-system

bore

stroke

compression-ratio

horsepower

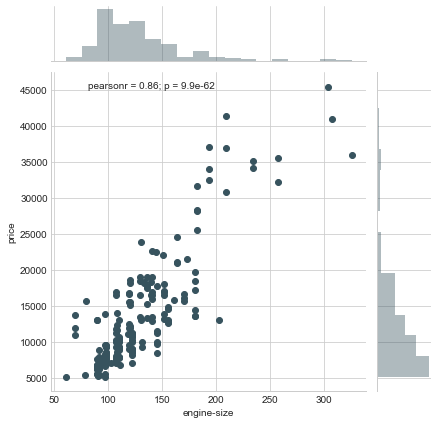
peak-rpm

city-mpg

highway-mpg

price

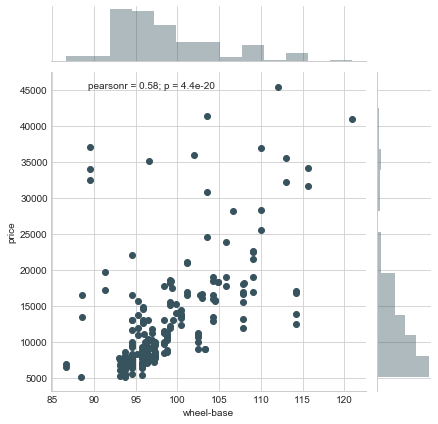
1. JointPlot for Engine Size vs Price



This Graphs gives an idea about the direct proportionality between engine-size and price.

If the engine size increases the price also increases and if engine size decreases the price also decreases thus, this direct relationship can be used to understand the data patterns and can be used for further conclusions.

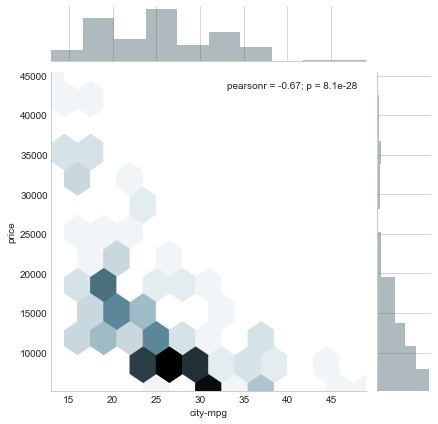
1. JointPlot for Wheel Base vs Price



This Graphs gives an idea about the direct proportionality between wheel base and price.

If the wheel base increases the price also increases and if wheel base decreases the price also decreases thus, this direct relationship can be used to understand the data patterns and can be used for further conclusions.

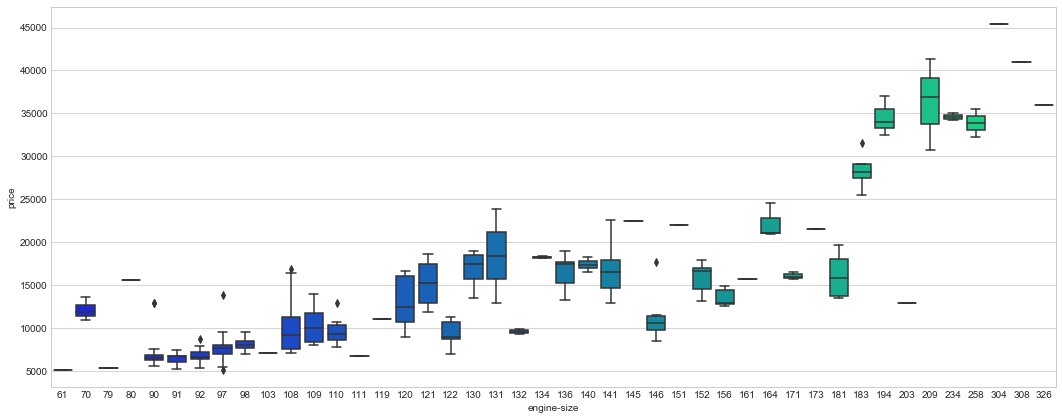
1. HexPlot for City-mpg vs Price



We can see here the Graph hasn’t any particular ordered relationship but if analyzed properly it can be found out the most numbers of price are for city in 30s unit. Thus it indirectly tells that most of the cars are bought for the cities are within this range and this information can be used for the next time while forming conclusions.

The Hex plot is darker where the density is more. As it can be seen in (10000,25-30).

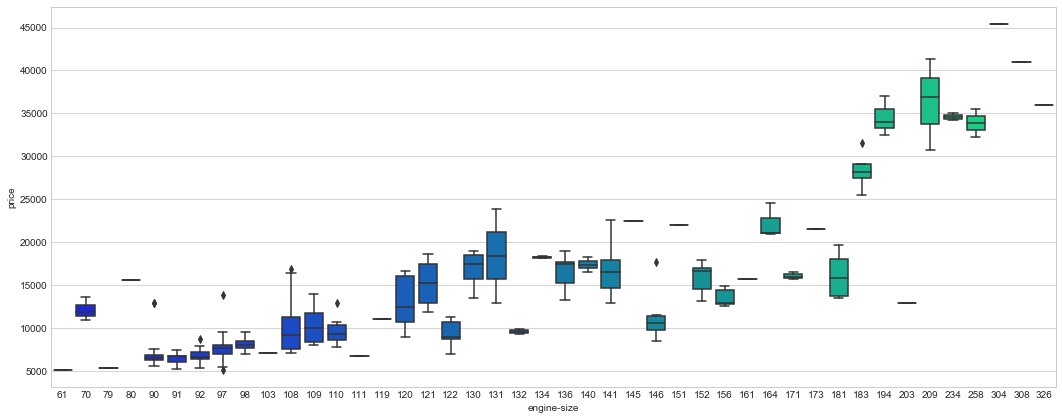
1. BoxPlot for Engine Size and Price



A box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable.

This Engine Size vs Price is not making any sense, It is hard to derive conclusions from it due to so anomalies and outliers. So we will leave it for now.

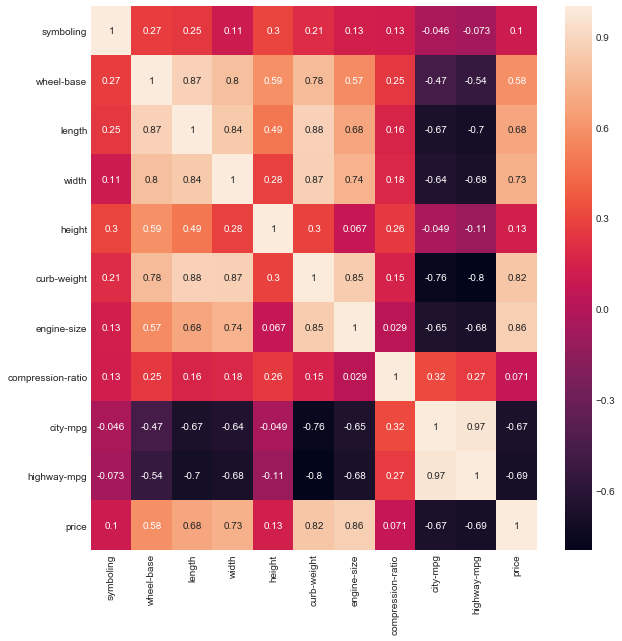
1. CountPlot for Engine Size and Price



Countplot is a general plot that allows you to aggregate the categorical data based off some function, by default the mean. This is essentially the same as except the estimator is explicitly counting the number of occurrences. Which is why we only pass the x value:

This Engine Size vs Price is not making any sense, It is hard to derive conclusions from it due to so anomalies and outliers. So we will leave it for now

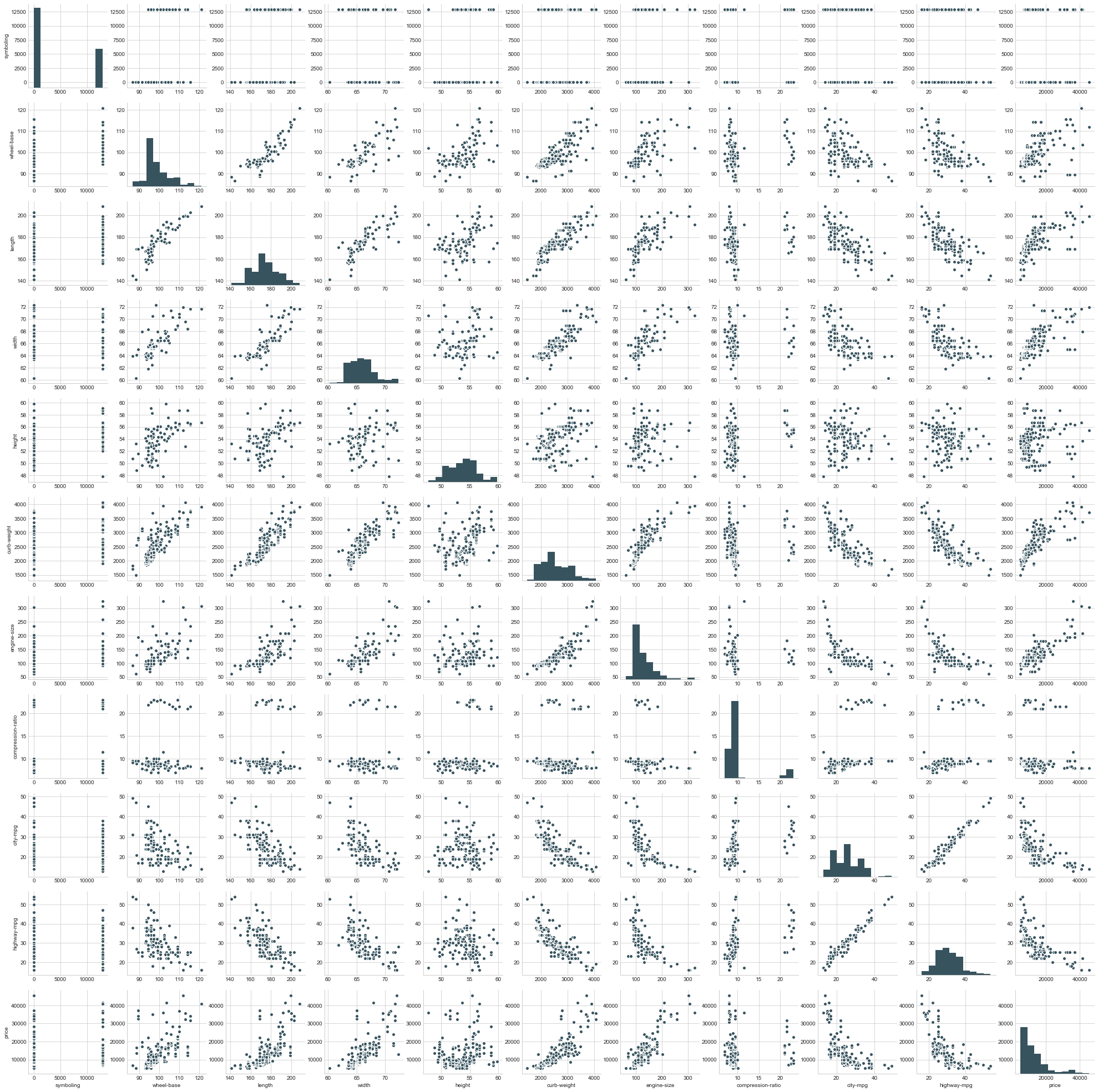
1. Heatmap



A heatmap is a two-dimensional graphical representation of data where the individual values that are contained in a matrix are represented as colors.

We can see the correlation between the different entities and their contribution in the model.

1. PairPlot



Pairplot will plot pairwise relationships across an entire dataframe (for the numerical columns) and supports a color hue argument (for categorical columns). It plots the numerical values only so we can see here all the numeric values attributes.

Now it is easy to identify the relationship among different attributes and by the graphs it can be seen as the majority of the entities follow a linear increasing pattern. Hence it led to say that Linear Regression will be the optimum choice for the model.

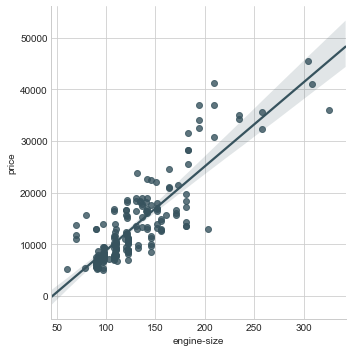
[Preparing Machine Learning model](#index)

1. By seeing the relation in model it can be found out that the datasets shows a direct relationship with the target attribute.
2. Also, we needed a continuous value in this case as the price will be a whole number ranging from (0 – ∞) both exclusive.
3. And, the target function is to be minimized as lower for prediction.
4. Thus, forming a conclusion after this Linear Regression would be the best Algorithm that suits our demand.

Linear Regression being very subtle in these cases will be much helpful and will provide significant contribution in the model prediction.

The Linear Algorithms that I chose for my project are

1. Linear Regression with scikit learn
2. Decision Trees Linear Regression
3. Random Forest Linear Regression



This is our Regression Line which Fits the Data perfectly.

1. Linear Regression with scikit learn

R2 SCORE

import sklearn.metrics as metric

metric.r2\_score(y\_test,prediction) #r2 score

0.83565206787682855

Coding Notebook

Training

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y, test\_size=0.3, random\_state=101)

from sklearn.linear\_model import LinearRegression

lm = LinearRegression()

lm.fit(x\_train,y\_train)

Testing

prediction = lm.predict(x\_test)

from sklearn import metrics #LinearRegression from sciket learn

print('MAE:', metrics.mean\_absolute\_error(y\_test, prediction))

print('MSE:', metrics.mean\_squared\_error(y\_test, prediction))

print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test, prediction)))

1. Linear Regression with Decision Trees

Coding Notebook

Training

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.3)

# Fitting Decision Tree Regression to the dataset

from sklearn.tree import DecisionTreeRegressor

regressor = DecisionTreeRegressor()

regressor.fit(X\_train.reshape(-1,1), y\_train.reshape(-1,1))

Testing

y\_pred = regressor.predict(X\_test.reshape(-1,1))

dt = pd.DataFrame({'Engine-size':y\_test.reshape(-1), 'Price':y\_pred.reshape(-1)})

# Visualising the Decision Tree Regression Results

X\_grid = np.arange(min(X), max(X), 0.01)

X\_grid = X\_grid.reshape((len(X\_grid), 1))

plt.scatter(X\_test, y\_test, color = 'red')

plt.scatter(X\_test, y\_pred, color = 'green')

plt.title('Decision Tree Regression')

plt.xlabel("Engine-size")

plt.ylabel('Price')

plt.show()

R2 SCORE

import sklearn.metrics as metric

metric.r2\_score(y\_test,prediction) #r2 score

0.67211037802025575

1. Linear Regression with Random Forest

Coding Notebook

Training

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.3)

# Fitting Random Forest Regression to the dataset

from sklearn.ensemble import RandomForestRegressor

regressor = RandomForestRegressor(n\_estimators = 10, random\_state = 0)

regressor.fit(X\_train.reshape(-1,1), y\_train.reshape(-1,1))

Testing

y\_pred = regressor.predict(X\_test.reshape(-1,1))

dt = pd.DataFrame({'Engine-size':y\_test.reshape(-1), 'Price':y\_pred.reshape(-1)})

# Visualising the Random Forest Regression Results

X\_grid = np.arange(min(X), max(X), 0.01)

X\_grid = X\_grid.reshape((len(X\_grid), 1))

plt.scatter(X\_test, y\_test, color = 'blue')

plt.scatter(X\_test, y\_pred, color = 'purple')

plt.title('Random Forest Regression')

plt.xlabel('engine-size')

plt.ylabel('price')

plt.show()plt.show()

R2 SCORE

import sklearn.metrics as metric

metric.r2\_score(y\_test,prediction) #r2 score

0.84355058115325099

[ML model chart](#index)

|  |  |  |
| --- | --- | --- |
| **Serial no** | **Algorithm** | **R2\_Score** |
| 1. | Random Forest | 0.84355058115325099 |
| 2. | Linear Regression | 0.83565206787682855 |
| 3. | Decision Trees | 0.67211037802025575 |

These are the respective r2 score of the algorithms for the linear regression.

So why this trend?

If we analyze the r2 score of the algos we find that Random forest tops the chart and works best for the model, this is because -

* Random forest are an ensemble of decision trees where the final prediction is taken by the cumulative average mean (or average pooling in regression) of the other trees as well and then the result is predicted.
* Then comes the Linear Regression which uses scikit learn a vast library which beautifully analyses the datasets and does the necessary backend function predicting a sensible outcome.
* And at the last the Decision trees, wasn’t these decision trees only did the whole job in random forest so how they ended up in so low score, so that can be explained as random forest is a totally different algorithm and has uses a different approach it asks multiples trees and irrespective of the bias, whereas in decision trees if one decision tree agrees for a prediction then the another may not agree and hence it conflicts thus bringing not a good outcome.

Yet these results are highly susceptible as it varies with the number of datasets having attributes and examples or by the track or theme of the project.

[Hurdles](#index)

After such well explained class of Abhishek Sir, it was hardly any concepts that he missed within the time frame as constrain. But challenges are something which pushes one for the betterment and dig out the real ideas and knowledge within.

I too had some challenging time while making the project, I am listing some of them which troubled a lot.

Challenges

1. Optimizing the dataset

The datasets which I found had many discrepancies and missing values so it was very necessary to make it correct and proper before passing it to machine.

* The data set contained ‘?’ a character datatype among the float numerical values so if it will be passed to the model it sends and error about the mismatch of datatype.
* It was very difficult to segregate and replace all the critical points as the data was vast.
* So, to overcome that I made an instance of the loaded data and then sent that data to pandas to take the numeric values
* hence explicitly making the string as float then the string was converted to null datatype. Then the Null was converted to 0.
* After that I took the mean of the target data (i.e. price) and then replaced the 0 by ‘.replace’ function with the mean of the attribute.
* Thus the dataset were ready to be trained by models.

Coding Notebook

df = pd.DataFrame(auto)

df ['price'] = pd.to\_numeric(df['price'], errors='coerce')

df = df.replace(np.nan, 0, regex=True)

np.average(df.price)

df = df.replace(0, 12949.429268292683, regex=True)

1. Varience and Bais Problem.

While calculating r2\_score it was sometimes seen that the score is less or it is abrupt.

If the training dataset is increased to ratio of 90-95% then the model actually learns the data and it becomes OVERFIT.

While, If the ratio is decreased to 10-30% the datasets loses the track and predict some redundant data hence becoming UNDERFIT.

So, As I had the limited dataset examples so I can’t increase or decrease the available so I had to find a sweet spot which allows the smooth prediction of the model.

Thus, I found the training data size of 0.3 suits the model for the given data in this situation.

Errors

1. **ValueError**: could not convert string to float: '?'

One of the most troublesome error which was persistant due to the redundant data present in the target varible.

I replaced the ? with the mean of the attribute.

1. **ValueError**: The truth value of an array with more than one element is ambiguous. Use a.any() or a.all()

This arised due to the irregular shape of the matrix and presence of redundant data.

Reshaping the matrix solves the issue.

1. **SyntaxError:** keyword can't be an expression

One of the basic errors originated due to improper syntax.

It can be solved if we refer to basics and write proper syntax.

1. **KeyError**: 'engin-size'

Not an error tho but a suggestion please how programmers would say please check you spelling.

1. **TypeError**: 'tuple' object is not callable

One of an interesting an logical error I faced is this. Actually it is the syntax of the python that tuples are present in the format of (a,b) but in our dataset we have only trained x as engine size and y as price no other value so how this tuple is formed?

So the answer to this would be because while passing the data in either x or y it takes the numeric value of the attribute and also adds a serial number to it which is why it is stored in form of tuple.

So to overcome this we can unpack tuples using a for loop and that would be tedious and when talking about a vast dataset it will increase the time and space complexity to the extent where we lose interest

Or, a better method would be pass only the values of tuples to another new list.

[Conclusion](#index)

Growing demands of cars I see this project as something very useful for people whereas it is a good step for a beginner to step in the world of data science and machine learning yet a challenging step to make something which brushes up all the concepts of the theme. Using Linear Regression was the obvious choice because of the nature of the dataset and the value we need to predict and the algorithms was a very cleverly chosen decision as to give a practical aspect as to test the theory which gives the definitions of why and how. At last the measuring the r2\_metric score stood up to be a good choice as though an artificial data it made a lot of sense using that.

It was a great opportunity for me to work on this given project. I feel I have learnt something and will be vigorously looking for such things in the future also.

[Bibliography](#index)

1. Linear Regression with scratch by Abhishek Sir
2. Concepts of Decision Trees by Abhishek Sir
3. Referring Random forests by Abhishek Sir
4. Ideas on Gradient Descent- Medium article
5. Article on Random forest - towards data science
6. Article on Decision trees – Medium , Reddit
7. Stack overflow (for error)
8. [Github](https://github.com/Aashutosh748/carmvp) for hosting repository