Python ML Internship

Project Report

Automobile Price prediction

Submitted by:

PRAJWAL.M.S

1VE18EC070

Gowdaprajwal009@gmail.com

Online Internship Organizes By:

IC Solutions

In association with **Takeiteasy_Engineers(TIE)**

Under the guidance of

Mr.Abhishek C

Acknowledgement

Firstly I would like to express my special thanks of gratitude to **Take It Easy Engineers**(**TIE**) for arranging this internship program.Also I would really like to thank **IC Solutions** for giving the students such a golden opportunity to do **Python ML internship** at just **₹799**. Providing such a quality training at low price is really appreciable.As doing an internship is a must for all the VTU students, it was really difficult to find good internship program during the pandemic. This online internship has really helped me.

I would like to extend my gratitude to my instructor **Mr Abhishek C.** I'm really fortunate that such a good trainer was assigned to me. He has so much knowledge in this area, so all the eleven sessions of this internship program were really informative. He shared his experience in the field of ML during the sessions which was really great. He used to clear all the doubts asked by each & every student, due to which all the concepts taught by him are crystal clear.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives.

Hope to continue cooperation with all of you in the future.

Sincerely,
PRAJWAL.M.S

Place: Bangalore

Date: 01/11/2020

Abstract

The current project is to predict the price of automobile using ML. An automobile price prediction requires noticeable effort and knowledge of the field expert. Considerable number of distinct attributes is examined for the reliable and accurate prediction. In this project, we were asked to experiment with a real world dataset, and to explore how machine learning algorithms can be used to find the patterns in the data. We were expected to gain experience using a common data-mining and machine learning library, and were expected to submit a report about the dataset and the algorithm used. After performing the required tasks on the dataset of price data of automobile, here lies my final report.To build a model for predicting the price of the automobile, three machine learning techniques (Linear Regression, Support Vector Regression and AdaBoost Regression) have been applied. The data used for the prediction is ‘Automobile price data _Raw_.csv’.

About the company

IC Solutions(ICS) is a digital service provider that aims to provide software, designing and marketing solutions to individuals and businesses. ICS believes that service and quality is the key to success.

They 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!

Development - They develop responsive, functional and super fast websites. They 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 - They offer a wide range of professional Android, iOS & Hybrid app development services for global clients, from a start-up to a large enterprise.

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

Consultancy - They provide expert advice on the client's design and development requirement.

Videos - They create a polished professional video that impresses the audience..

INDEX

Introduction

Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data.

Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range.

An ML algorithm is supposed to perform task and gain experience with the passage of time. The measure which tells whether ML algorithm is performing as per expectation or not is its performance (P). P is basically a quantitative metric that tells how a model is performing the task, T, using its experience, E. There are many metrics that help to understand the ML performance, such as accuracy score, r2_score, confusion matrix, precision, recall, sensitivity etc.

From this internship program I learned the basics of Artificial Intelligence (AI), Machine Learning using Python, Data Analysis & Data Visualization using different libraries, Training & Testing the models using ML algorithms like Linear Regression, Logistic Regression, Support Vector Machines, Decision Trees, Random Forest & K Nearest Neighbors.

Using the knowledge gained by this internship I have completed a ML project which involved Exploratory Data Analysis, Training & testing the model using three different algorithms.

Problem Statement

To predict the price of different automobiles based on the given data set. Using these data set we have to train a Machine Learning model to find efficiency and price of the car.

Objective

1. Data Analysis is done to analyse the given data set & summarize their main characteristics.
2. To predict the price of the automobile, we need to apply Regression algorithm. After training & testing the model the r2_score has to be evaluated for all the three algorithms.

System Requirements

Hardware Specifications (Minimum Requirement):-

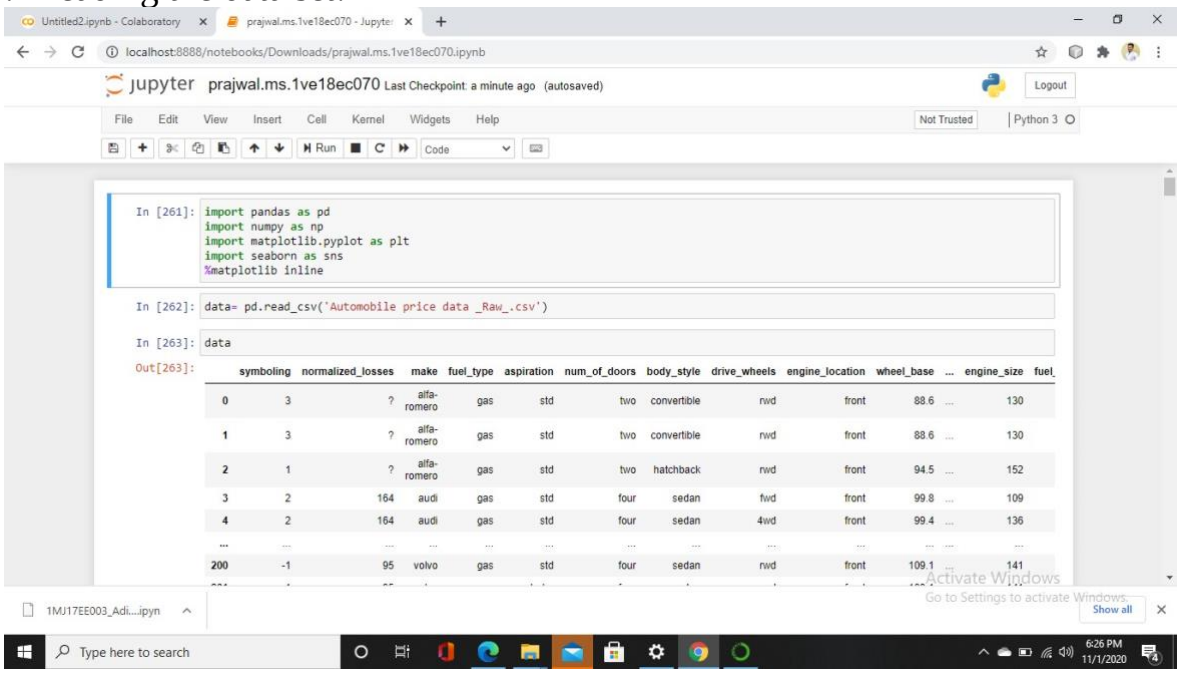
- RAM: 4 GB
- CPU: Processor above Intel Corei3 8th Gen
- OS: Windows 10/Mac OS

Software Requirements:-

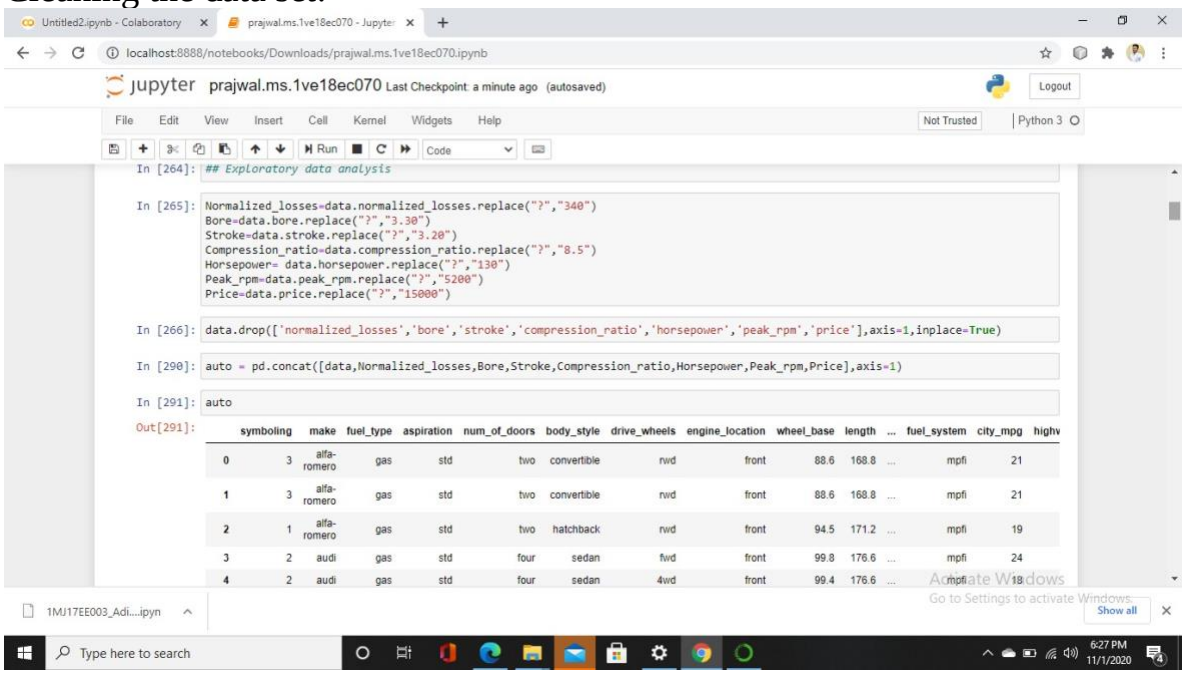
- Jupyter Notebook
- Pandas
- NumPy
- Scikit-learn

Exploratory Data Analysis

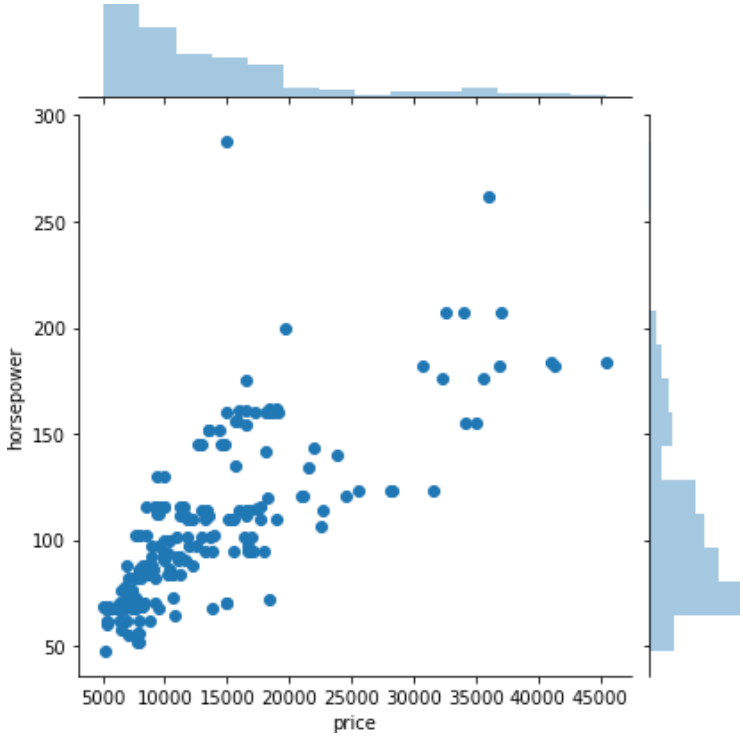
1. Reading the data set:



2. Cleaning the data set:

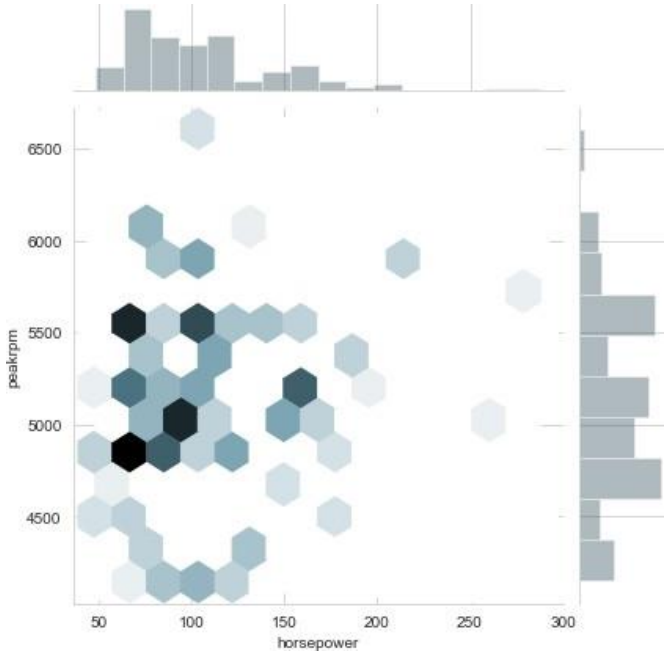


3. Data Analysis:

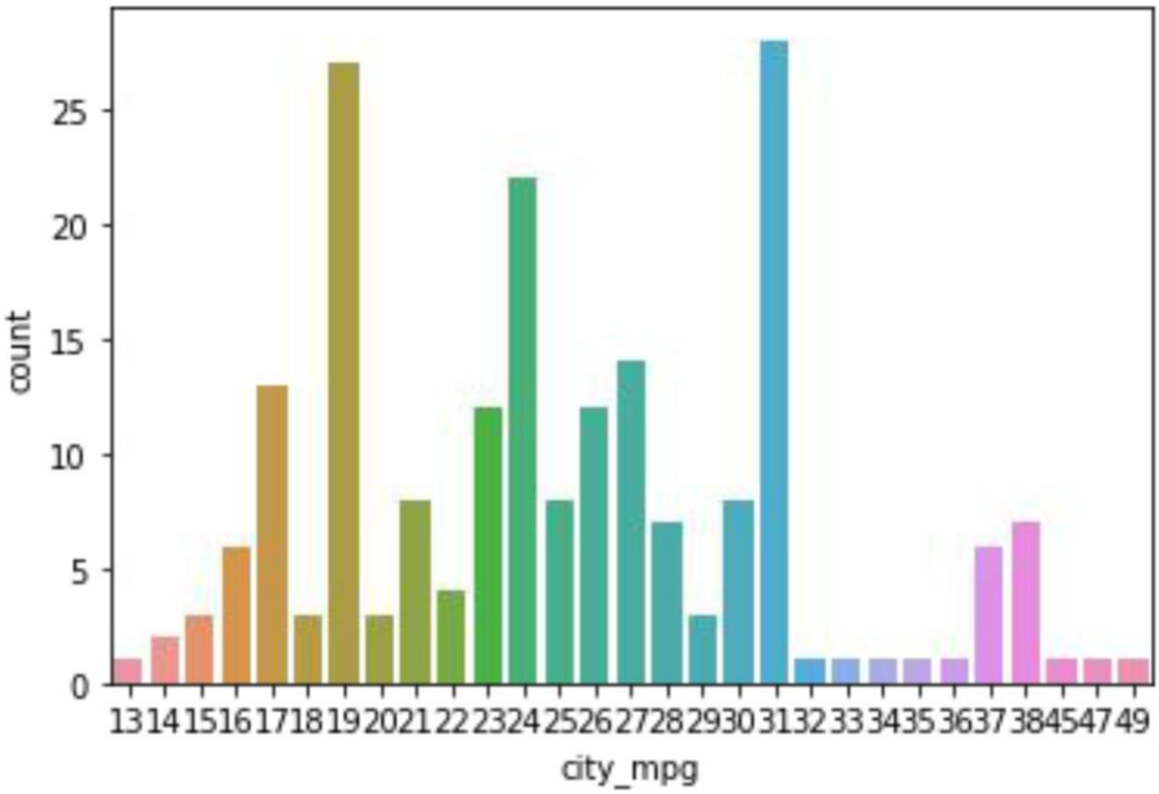


i) From the above graph we can conclude that cars having an engine with 70-100 HP falls under the price range of 6000 to 10000 USD.

ii) There are very few cars which have an engine with 180-210 HP & they cost around 32000 to 45000 USD.



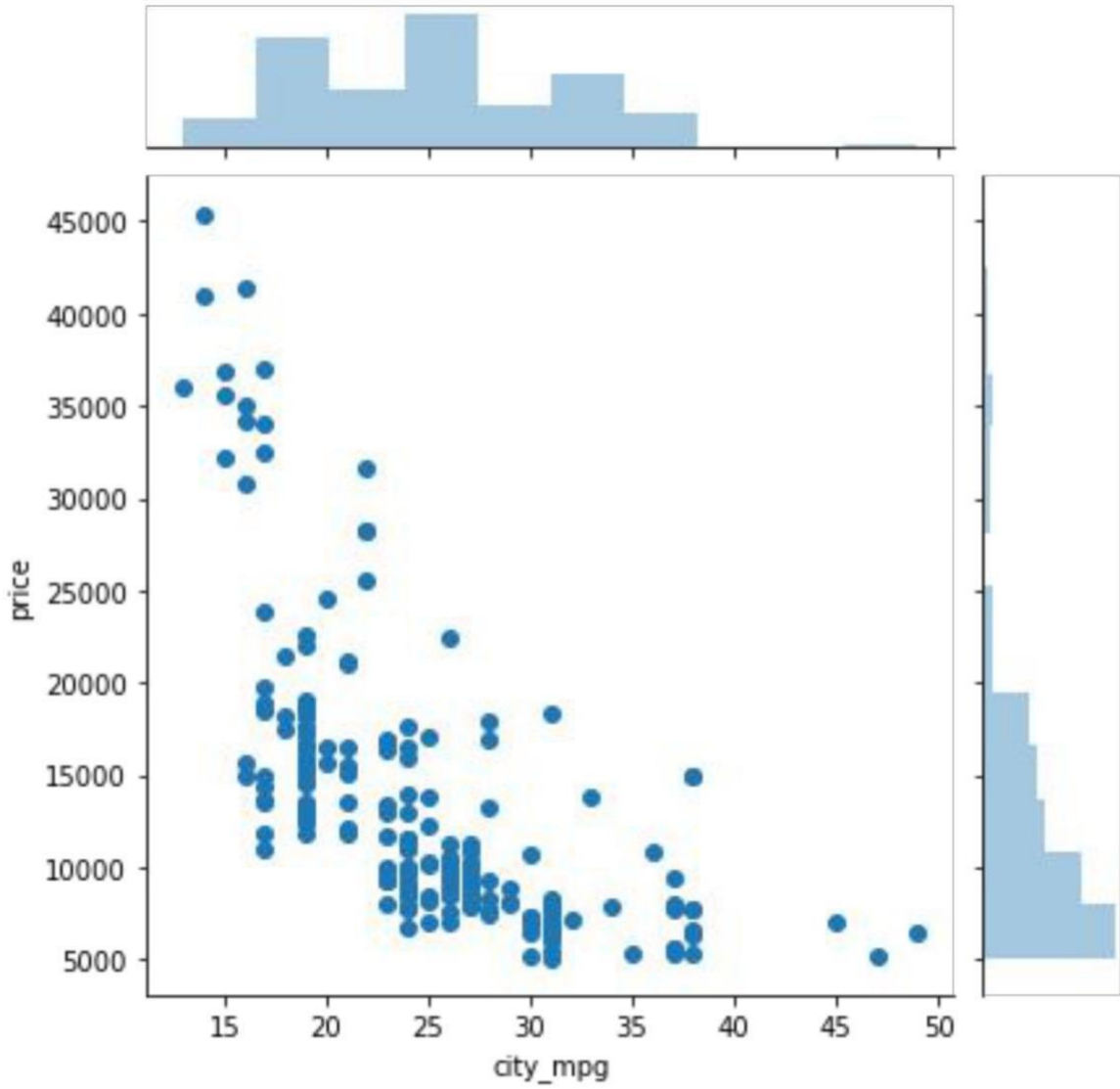
i)The above graph is between peakrpm vs horsepower, horsepower decreases with increases in peakrpm.



i) From the above graph we can conclude that the number of cars which gives 31mpg in city is 30, which is the highest among the others.

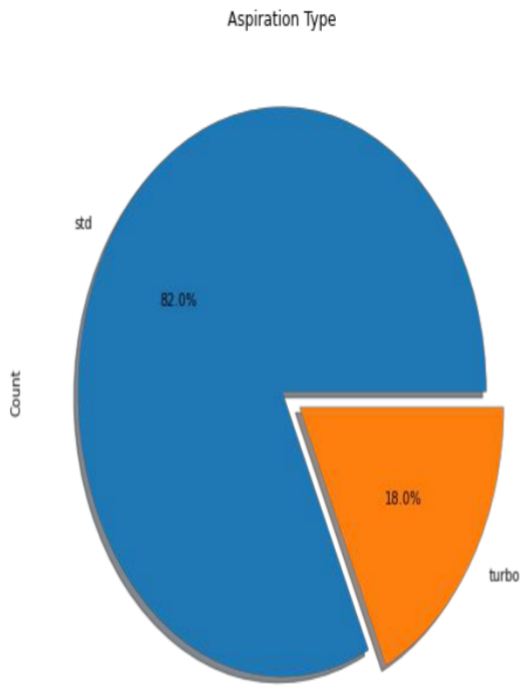
ii) There are just one or two cars which gives 13mpg in city.

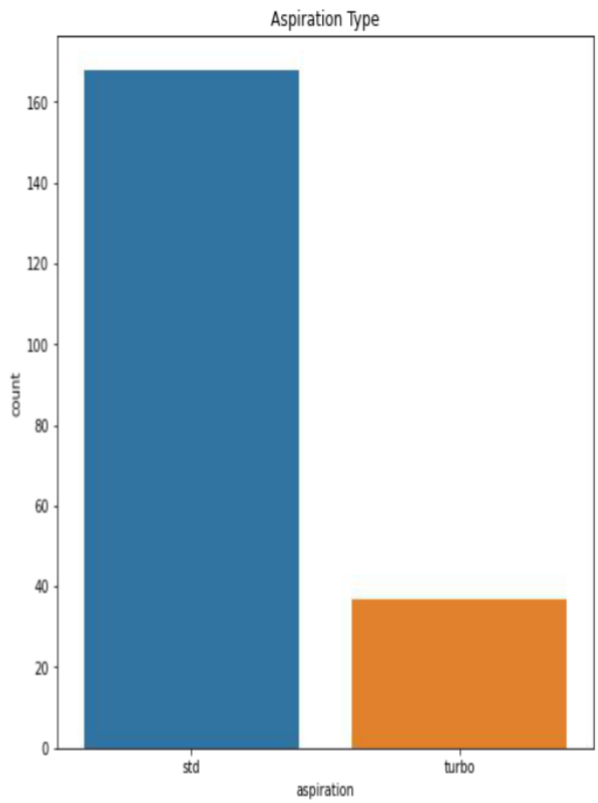
iii) Likewise, the number of cars which gives 45,47 & 49mpg in city is just 1.



i) The cars which gives 24-27mpg in city costs around 7000 to 11000 USD.

ii) The super cars which gives just 15mpg in city cost around 30000 to 41000 USD.

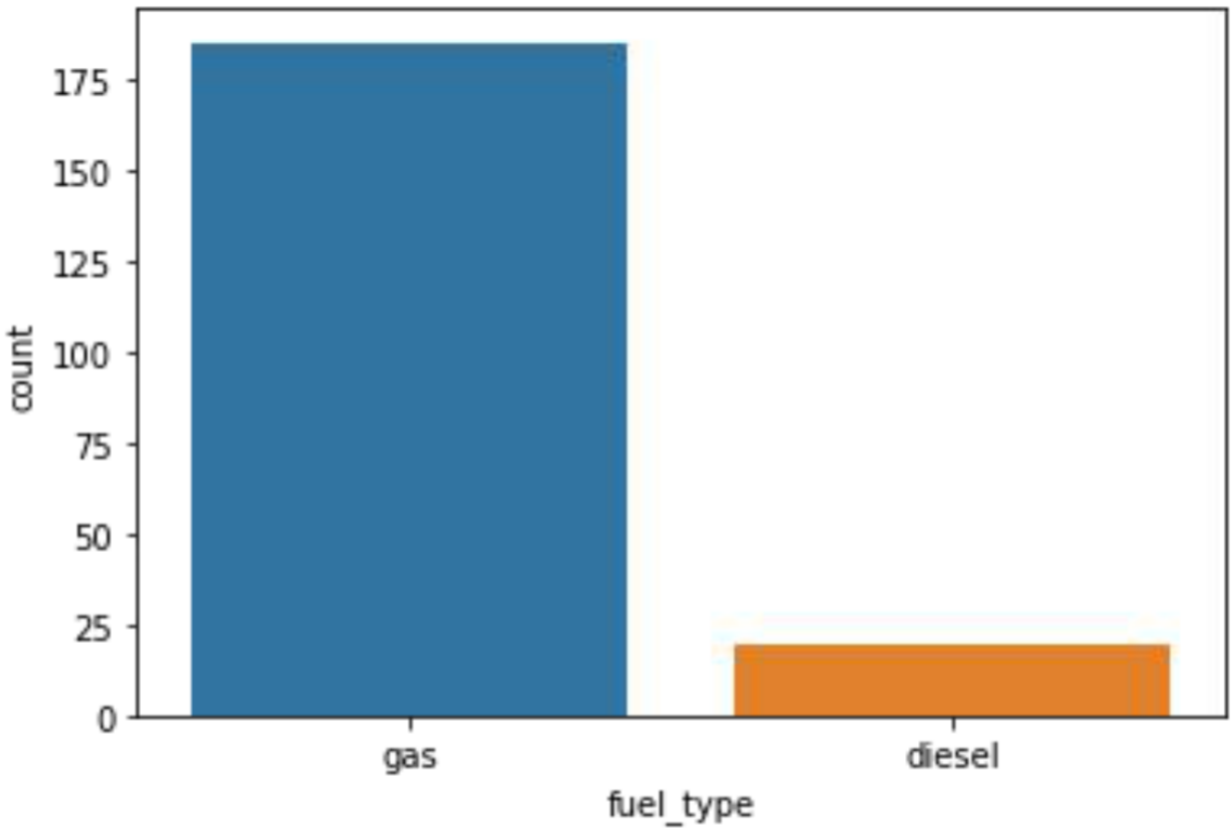




i) Most of the vehicles have standard aspiration.

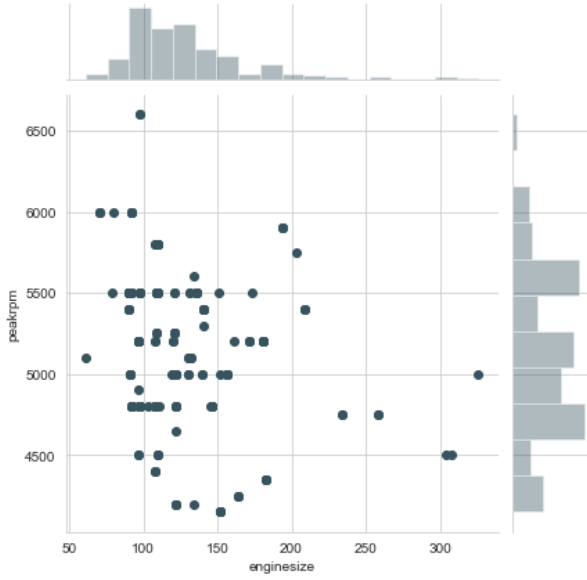
ii) There are only 38 cars which have turbo aspiration.

iii) The number of vehicles with standard aspiration is 166.

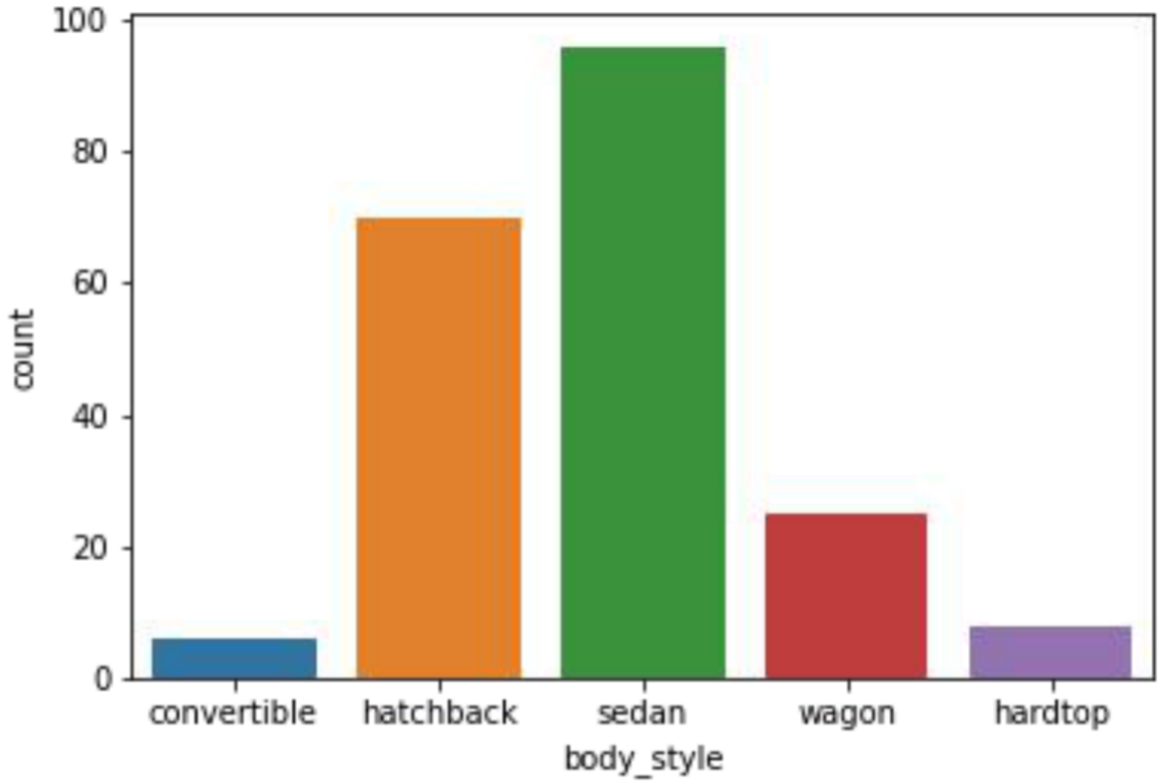


i) Most of the vehicles use gasoline as their fuel.

ii) There are only 24 diesel based vehicles.



i)From the above graph we can say that engine with small size will have high peak rpm



i) Most of cars have Sedan body style. There are 95 Sedan shaped cars.

ii) There are just 5 convertible type cars.

iii) There are 25 wagon style cars.

iv) There are 9 hardtop type of cars.

Machine Learning Models

We need to use Regression algorithms on the given data set in order predict the price of the car.

There are many Regression algorithms, like

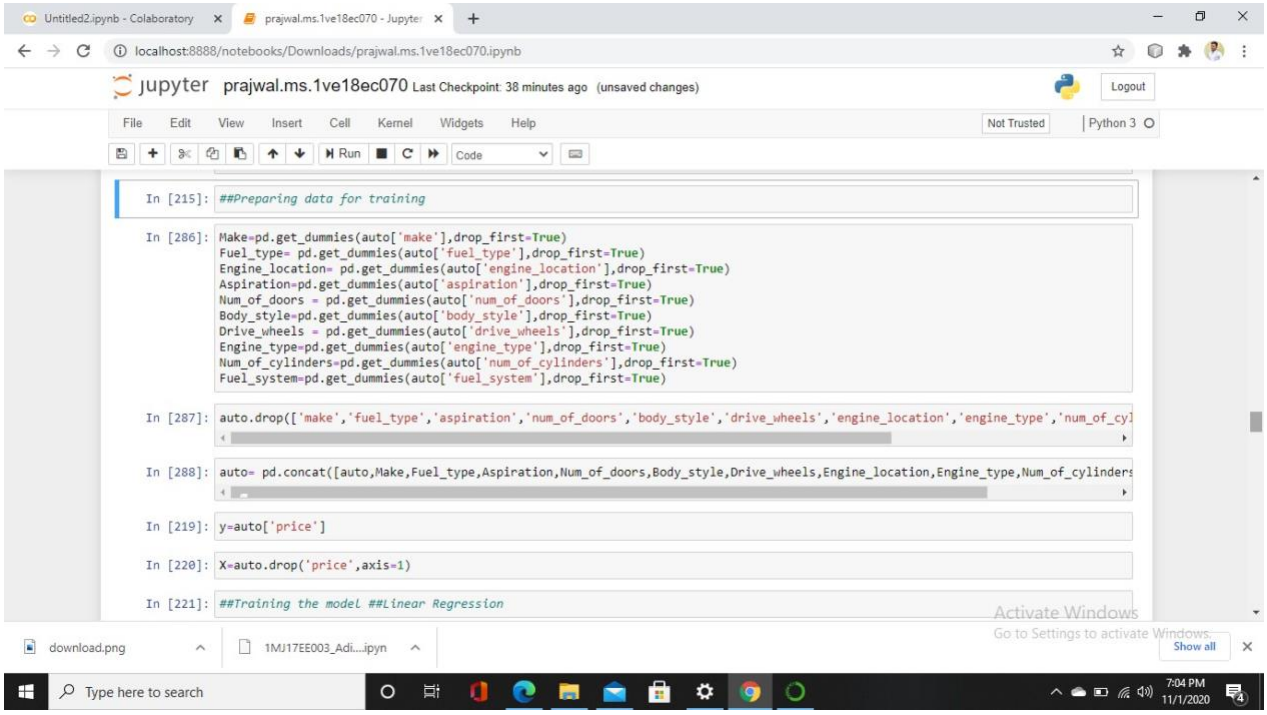
A. Linear Regression

B. Lasso Regression

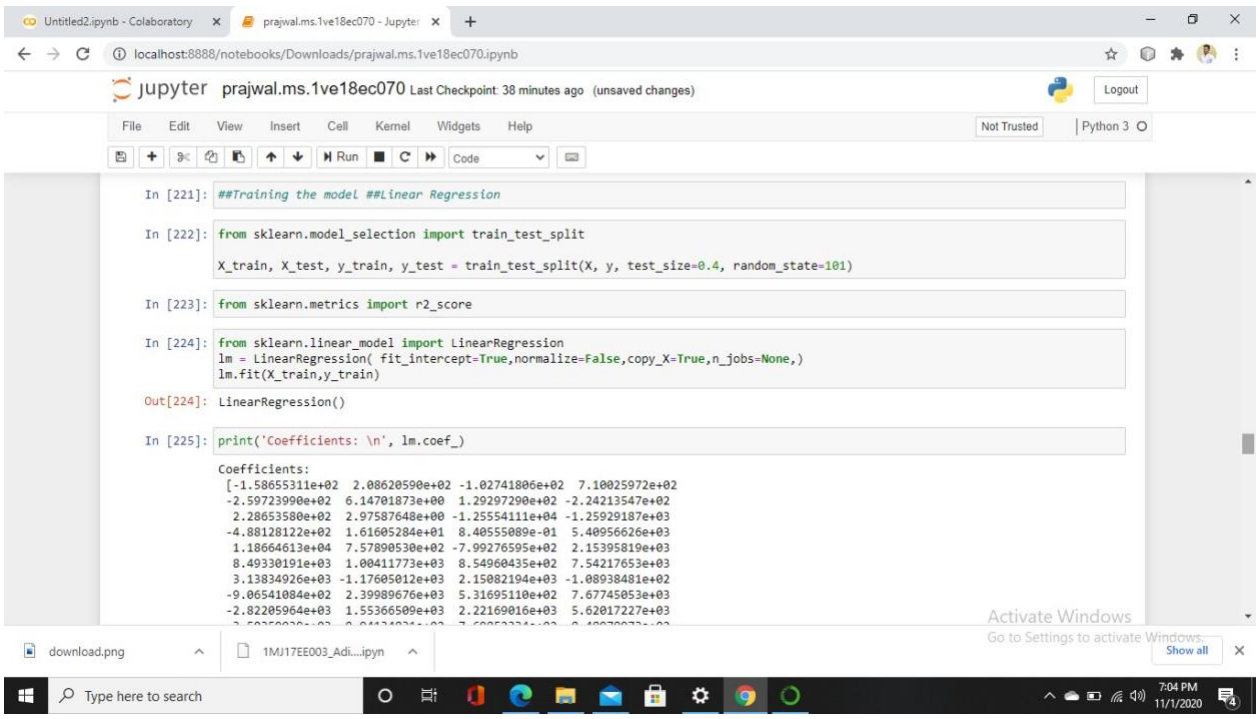
C. Support Vector Regression

D. Decision Tree Regression

1) Linear Regression Model



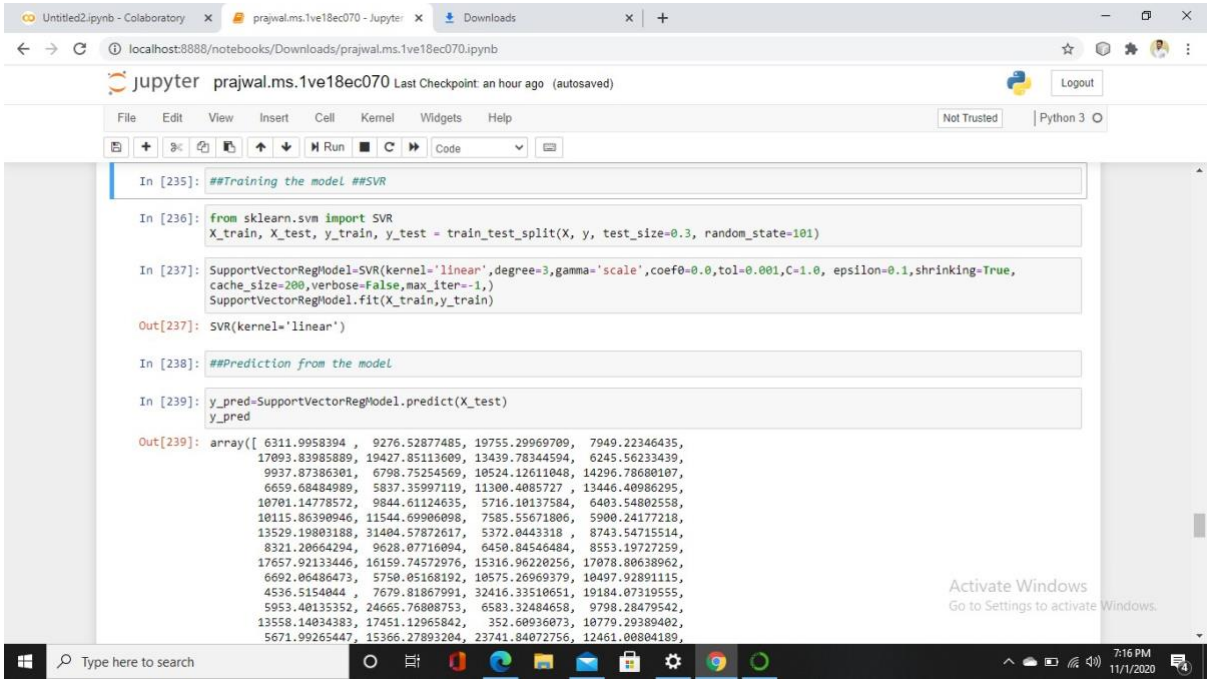
```
In [215]: ##Preparing data for training

In [286]: Make=pd.get_dummies(auto['make'],drop_first=True)
          Fuel_type= pd.get_dummies(auto['fuel_type'],drop_first=True)
          Engine_location= pd.get_dummies(auto['engine_location'],drop_first=True)
          Aspiration=pd.get_dummies(auto['aspiration'],drop_first=True)
          Num_of_doors = pd.get_dummies(auto['num_of_doors'],drop_first=True)
          Body_style=pd.get_dummies(auto['body_style'],drop_first=True)
          Drive_wheels = pd.get_dummies(auto['drive_wheels'],drop_first=True)
          Engine_type=pd.get_dummies(auto['engine_type'],drop_first=True)
          Num_of_cylinders=pd.get_dummies(auto['num_of_cylinders'],drop_first=True)
          Fuel_system=pd.get_dummies(auto['fuel_system'],drop_first=True)

In [287]: auto.drop(['make','fuel_type','aspiration','num_of_doors','body_style','drive_wheels','engine_location','engine_type','num_of_cyl

In [288]: auto= pd.concat([auto,Make,Fuel_type,Aspiration,Num_of_doors,Body_style,Drive_wheels,Engine_location,Engine_type,Num_of_cylinders

In [219]: y=auto['price']

In [220]: X=auto.drop('price',axis=1)

In [221]: ##Training the model ##Linear Regression
```



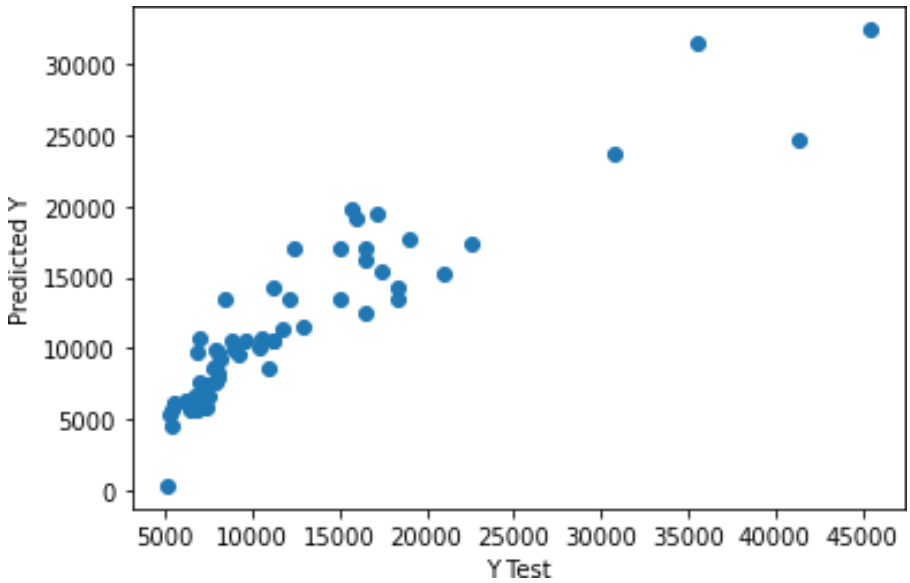
```
In [221]: ##Training the model ##Linear Regression

In [222]: from sklearn.model_selection import train_test_split

          X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=101)

In [223]: from sklearn.metrics import r2_score

In [224]: from sklearn.linear_model import LinearRegression
          lm = LinearRegression( fit_intercept=True,normalize=False,copy_X=True,n_jobs=None,)
          lm.fit(X_train,y_train)

Out[224]: LinearRegression()

In [225]: print('Coefficients: \n', lm.coef_)

          Coefficients:
           [-1.58655311e+02  2.08620590e+02 -1.02741806e+02  7.10025972e+02
           -2.59723990e+02  6.14701873e+00  1.29297290e+02 -2.24213547e+02
            2.28653580e+02  2.97587648e+00 -1.25554111e+04 -1.25929187e+03
           -4.88128122e+02  1.61605284e+01  8.40555089e-01  5.40956626e+03
            1.18664613e+04  7.57890530e+02 -7.99276595e+02  2.15395819e+03
            8.49330191e+03  1.00411773e+03  8.54960435e+02  7.54217653e+03
            3.13834926e+03 -1.17605012e+03  2.15082194e+03 -1.08938481e+02
           -9.06541084e+02  2.39989676e+03  5.31695110e+02  7.67745053e+03
           -2.82205964e+03  1.55366509e+03  2.22169016e+03  5.62017227e+03
```

The r2_score of Linear Regression model is **0.8284**

2) Support Vector Regression Model (SVR)

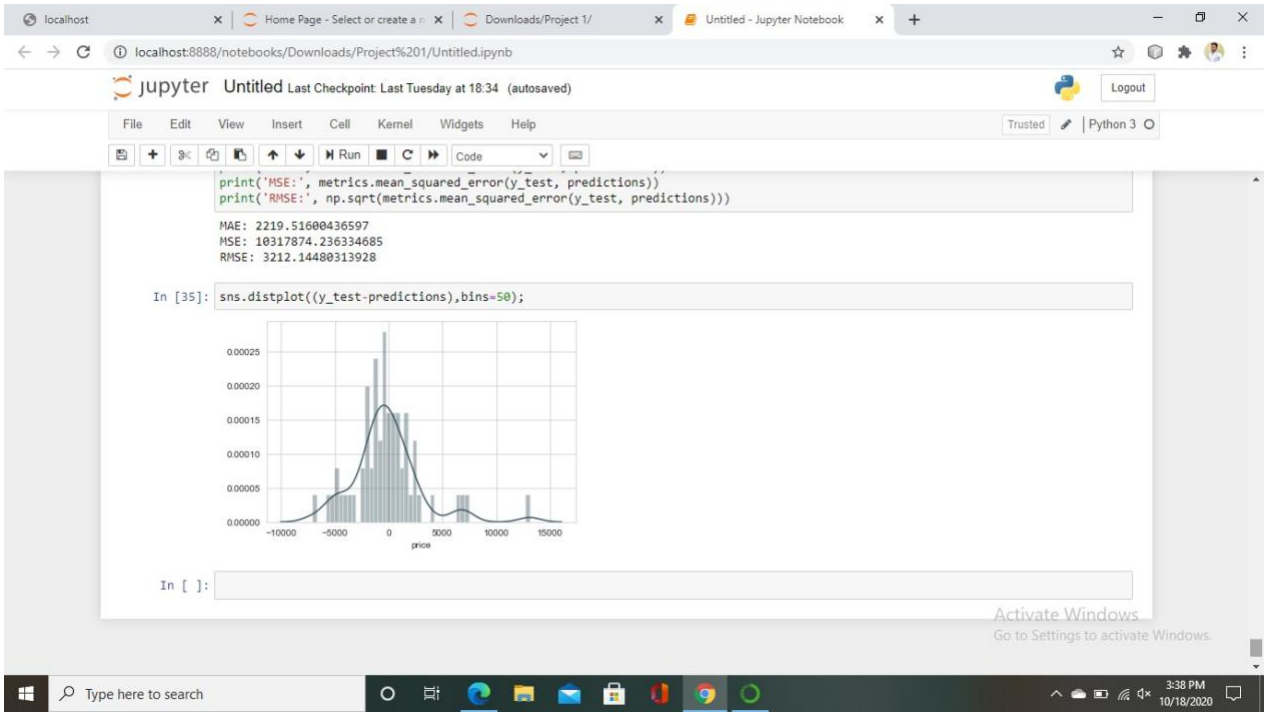


```
In [235]: ##Training the model ##SVR

In [236]: from sklearn.svm import SVR
          X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)

In [237]: SupportVectorRegModel=SVR(kernel='linear',degree=3,gamma='scale',coef0=0.0,tol=0.001,C=1.0, epsilon=0.1,shrinking=True,
          cache_size=200,verbose=False,max_iter=-1,)
          SupportVectorRegModel.fit(X_train,y_train)

Out[237]: SVR(kernel='linear')

In [238]: ##Prediction from the model

In [239]: y_pred=SupportVectorRegModel.predict(X_test)
          y_pred

Out[239]: array([ 6311.9958394 ,  9276.52877485, 19755.29969709,  7949.22346435,
                 17093.83985889, 19427.85113609, 13439.78344594,  6245.56233439,
                  9937.87386301,  6798.75254569, 10524.12611048, 14296.78680107,
                  6659.68484989,  5837.35997119, 11300.4085727 , 13446.40986295,
                 10701.14778572,  9844.61124635,  5716.10137584,  6403.54802558,
                 10115.86390946, 11544.69906098,  7585.55671806,  5900.24177218,
                 13529.19803188, 31404.57872617,  5372.0443318 ,  8743.54715514,
                  8321.20664294,  9628.07716094,  6450.84546484,  8553.19727259,
                 17657.92133446, 16159.74572976, 15316.96220256, 17078.80638962,
                  6692.06486473,  5750.05168192, 10575.26969379, 10497.92891115,
                  4536.5154044 ,  7679.81867991, 32416.33510651, 19184.07319555,
                  5953.40135352, 24665.76808753,  6583.32484658,  9798.28479542,
                 13558.14034383, 17451.12965842,   352.60936073, 10779.29389402,
                  5671.99265447, 15366.27893204, 23741.84072756, 12461.00804189,
```



The r2_score of SVR is **0.81**

3) Decision Tree Model



Decision tree regressor:-

model=DecisionTreeRegressor()

model.fit(X_train,y_train)

DecisionTreeRegressor()

DTree=prediction.astype(int)

DTree

array([18802, 22216, 9749, 21454, 15776, 7736, 11299, 15190, 6171,

5297, 5995])

pred = model.predict(X_test)

from sklearn.metrics import r2_score

print(r2_score(y_test, pred))

0.8527946742442184

ML Model Chart

Serial Number	Algorithm Name	r2_score
1	**Decision Tree**	**0.852**
2	**Linear Regression**	**0.8284**
3	**Support Vector Regression**	**0.8083**

Hurdles

a) I was getting a negative r2_score for SVR model. Then I realized that if I change the parameter, then the r2_score will improve. After doing so I got better r2_score.

b) While performing Grid search I had given the Kernel value as Linear & it was taking forever to run that particular line. Linear gives the best fit, but it takes too much time to run. So I changed the Kernel to its default value, then there was no issue.

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 preprocessing of the data.

Data cleaning is one of the processes that increases prediction performance.

On the whole, this internship was a useful experience. I have gained new knowledge, skills and met many new people. I achieved several of my learning goals.

The internship was also good to find out what my strengths and weaknesses are. This helped me to define what skills and knowledge I have to improve in the coming time.

Bibliography

1. Seaborn Jupyter Notebook given by the tutor.

2. Support Vector Machines Jupyter Notebook given by the tutor.
3. Linear Regression with Sklearn Jupyter Notebook given by the tutor.