MACHINE LEARNING BASED VEHICLE PERFORMANCE ANALYZER

TEAM ID:PNT2022TMID37180

BACHELOR OF ENGINEERING
COMPUTER SCIENCE AND ENGINEERING

ANAND INSTITUTE OF HIGHER TECHNOLOGY

INDUSTRY MENTOR : NIDHI
FACULTY MENTOR : ANGEL PETRICIA V

Team Members:
NIRMAL A V - 310119104302
NAVEEN KS - 310119104054
MUSHARRAF U - 310119104701
HARI VIGNESH K - 310119104303

Introduction:

1.1 Project overview:

The monitoring of car performance, especiallygas consumption, has so far been approached only very superficially. A typical fuel gauge, when closely monitored,shows an extremelynon-linear relationship betweenneedle movement and fuel consumption. Inaccuracies occur especially in the range of critical low fuel values of 510% or more. In the past, due to this limitation, some luxury cars had an audible and flashing light alarm functionto indicate a low fuel condition. These systems, which add to the existingfuel level, have no more accuracy than the fuel level monitor alone.In recent years, with the availability of computer techniquesand reliable and less expensive computer equipment, a number of systems have been developed to provide some what more accurate information about vehicle performance.

1.2 PURPOSE :

The solution mainly aim on to predict the miles per gallon value based on the given input values that effect the performance of the vehicle.

2.LITERATURE SURVEY

S.No.	Topic	Methodology	References
1	**Topic:**Car Sales Prediction Using Machine Learning Algorithms **Author:** K. Madhuvanthi	An analytic hierarchy methodology is implemented in order to get varied idea about how well the various criteria's in our dataset works and after this we apply the machine learning algorithms such as Linear regression, Random tree to get the best clusters and we process them in to random forest to get best accurate feature out of it. which is ultimately followed by Technique for Order of preference by similarity to ideal	1. Keivan Kianmehr a, Reda Alhajj a,b,*2008 , Calling communities analysis and identification using machine learning techniques, journal homepage: www.elsevier.com/locate/eswa. 2. Yaya Xie a, Xiu Li a,*, E.W.T. Ngai b, Weiyun Ying c,2008, Customer churn prediction using improved balanced random forests, journal homepage: www.elsevier.com/locate/eswa.

		solution (TOPSIS) .	
2	**Topic**:Machine Learning Based Real-Time Vehicle Data Analysis for Safe Driving Modeling **Author:** Pamul Yadav	Supervised learning based linear regression model that is used as an estimator for Driver's Safety Metrics and Economic Driving Metrics.	[1] Singh D, Singh M., "Internet of Vehicles for Smart and Safe Driving", International Conference on Connected Vehicles and Expo (ICCVE), Shenzhen, 19-23 Oct., 2015. [2] Zhang, Y., Lin, W., and Chin, Y., "Data-Driven Driving Skill Characterization: Algorithm Comparison and Decision Fusion," SAE Technical Paper 2009-01- 1286, 2009, https://doi.org/10.4271/2009-01-1286.Azevedo, C. L Cardoso.

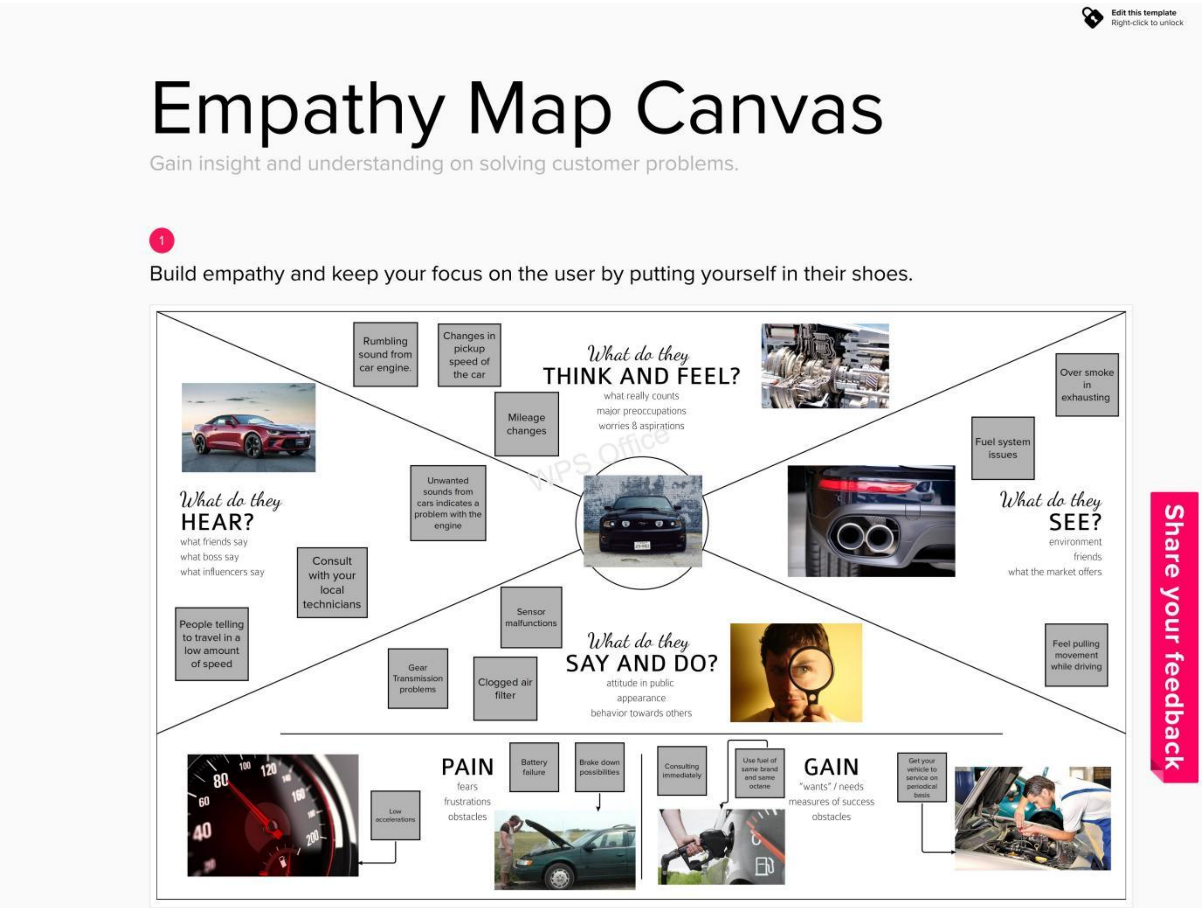
3	**Topic:** Performance of Motor Vehicle based on Driving and Vehicle Data using Machine Learning **Author:** Punith Kumar Nagaraje Gowda	The data for this analysis was taken from the the OBD of the car and models are built using techniques like Multiple Linear Regression, XGBoost, Support Vector Machine and Artificial Neural Network .	Cortes, C. and Vapnik, V. (1995). Support-vector networks, Machine learning, 20(3), pp. 273-297 . Fayyad, U. M., Haussler, D. and Stolorz, P. E. (1996). Kdd for science data analysis: Issues and examples., KDD pp. 50-56. Freedman, D. A. (2009). Statistical models: theory and practice, cambridge university press.

2.3 Problem statement definition:

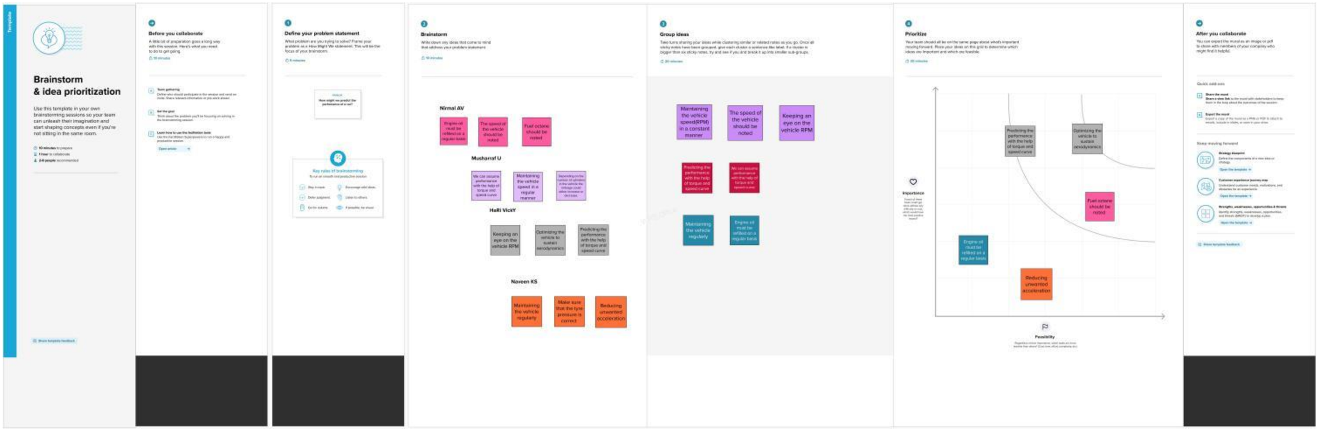
It is an important to analyse the factors using number of well known approaches of machine learning algorithms like linear regression, decision tree and random forest to improve the vehicle performance efficiency.The range, durability and longevity of automotive traction batteries are 'hot topics' in automotive engineering. And here we consider a performance in mileage. To solve this problem, we will develop the models, using the different algorithms and neural networks. We will then see which algorithm predicts car performance(Mileage) with higher accuracy.

3.Ideation and proposed solution:

3.1 Empathy map canvas



3.2 IDEATION ANDBRAIN STORMING :



3.3 PROPOSED SOLUTION:

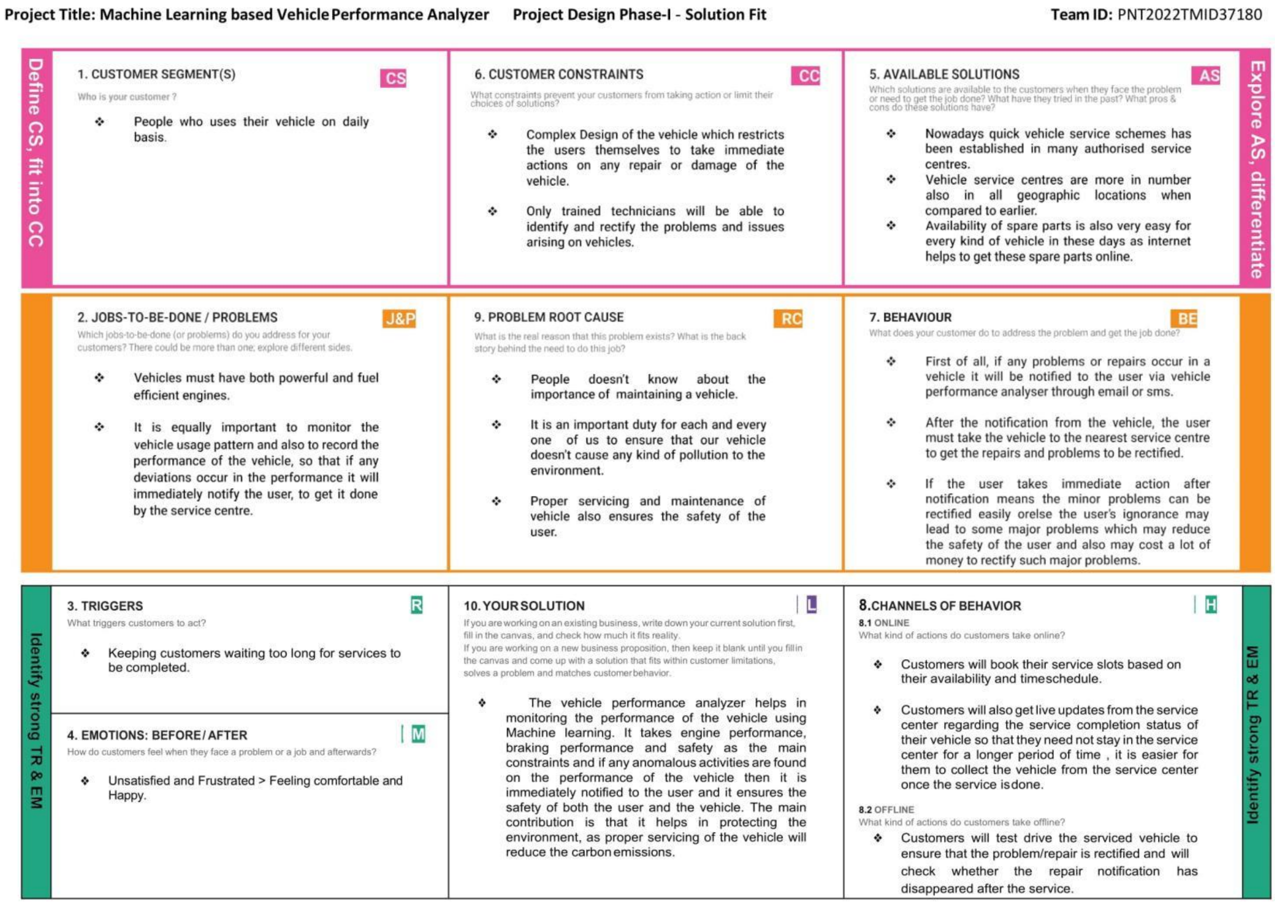
Project Design Phase-I
Proposed Solution Template

Date	15 October 2022
Team ID	PNT2022TMID37180
Project Name	Machine Learning based vehicle performance Analyzer
Maximum Marks	2 Marks

Proposed Solution :

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Vehicle user or manufacture trying to analyse the performance of the vehicle But, it is hard to analysis. Because it needs a knowledge of the engineering and vehicle, it takes time to do it manually, which makes users feel fear, worried about the vehicle
2.	Idea / Solution description	Dataset of the Vehicle performance need to be collected and need to analyse the data. Based on the data analysis Machine Learning Model should be created and need to test the accuracy of the model and the error of the model.
3.	Novelty / Uniqueness	Using this Machine Learning project we can develop the app in that app we can frequently update the dataset and train the model, So the user can get the accurate data
4.	Social Impact / Customer Satisfaction	The Social impact for this product is good, It make people life easier by perform analyse of the vehicle
5.	Business Model (Revenue Model)	Alige Model, MVP (Minimum Viable Product) Model
6.	Scalability of the Solution	It can be further developed to provide app integration, We can further develop the project to bring more accuracy.

3.4 Problem Solution fit



Project Title: Machine Learning based Vehicle Performance Analyzer **Project Design Phase-I - Solution Fit** **Team ID:** PNT2022TMID37180

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer ?

- People who uses their vehicle on daily basis.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

- Complex Design of the vehicle which restricts the users themselves to take immediate actions on any repair or damage of the vehicle.
- Only trained technicians will be able to identify and rectify the problems and issues arising on vehicles.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have?

- Nowadays quick vehicle service schemes has been established in many authorised service centres.
- Vehicle service centres are more in number also in all geographic locations when compared to earlier.
- Availability of spare parts is also very easy for every kind of vehicle in these days as internet helps to get these spare parts online.

Explore AS, differentiate

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

- Vehicles must have both powerful and fuel efficient engines.
- It is equally important to monitor the vehicle usage pattern and also to record the performance of the vehicle, so that if any deviations occur in the performance it will immediately notify the user, to get it done by the service centre.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?

- People doesn't know about the importance of maintaining a vehicle.
- It is an important duty for each and every one of us to ensure that our vehicle doesn't cause any kind of pollution to the environment.
- Proper servicing and maintenance of vehicle also ensures the safety of the user.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?

- First of all, if any problems or repairs occur in a vehicle it will be notified to the user via vehicle performance analyser through email or sms.
- After the notification from the vehicle, the user must take the vehicle to the nearest service centre to get the repairs and problems to be rectified.
- If the user takes immediate action after notification means the minor problems can be rectified easily orelse the user's ignorance may lead to some major problems which may reduce the safety of the user and also may cost a lot of money to rectify such major problems.

Identify strong TR & EM

3. TRIGGERS R

What triggers customers to act?

- Keeping customers waiting too long for services to be completed.

4. EMOTIONS: BEFORE / AFTER M

How do customers feel when they face a problem or a job and afterwards?

- Unsatisfied and Frustrated > Feeling comfortable and Happy.

10. YOUR SOLUTION L

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behavior.

- The vehicle performance analyzer helps in monitoring the performance of the vehicle using Machine learning. It takes engine performance, braking performance and safety as the main constraints and if any anomalous activities are found on the performance of the vehicle then it is immediately notified to the user and it ensures the safety of both the user and the vehicle. The main contribution is that it helps in protecting the environment, as proper servicing of the vehicle will reduce the carbon emissions.

8. CHANNELS OF BEHAVIOR H

8.1 ONLINE

What kind of actions do customers take online?

- Customers will book their service slots based on their availability and timeschedule.
- Customers will also get live updates from the service center regarding the service completion status of their vehicle so that they need not stay in the service center for a longer period of time , it is easier for them to collect the vehicle from the service center once the service is done.

8.2 OFFLINE

What kind of actions do customers take offline?

- Customers will test drive the serviced vehicle to ensure that the problem/repair is rectified and will check whether the repair notification has disappeared after the service.

Identify strong TR & EM

4. REQUIREMENT ANALYSIS :

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	• The system doesn't require any prior technical knowledge from the user, thus even a novice user can access it. • The user interface would prioritize recognition over recall. • User friendly • Pay attention to internal sources of control • It wouldn't take long for the content to load and show (30 seconds). • The fields in the site would be selfexpla
NFR-2	Security	• Only the authenticated user will be able to use

		the site's services. • The database should be backed up every hour.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Details	No of cylinders, Displacement, Horsepower, Weight, Model year, and Origin
FR-4	User Requirements	• Upload all essential details to the website's appropriate. • The system would extract all essential data based on the uploads. • Based on the information that was scraped, a list of every potential potential results will be delivered.

Non-functional Requirements:

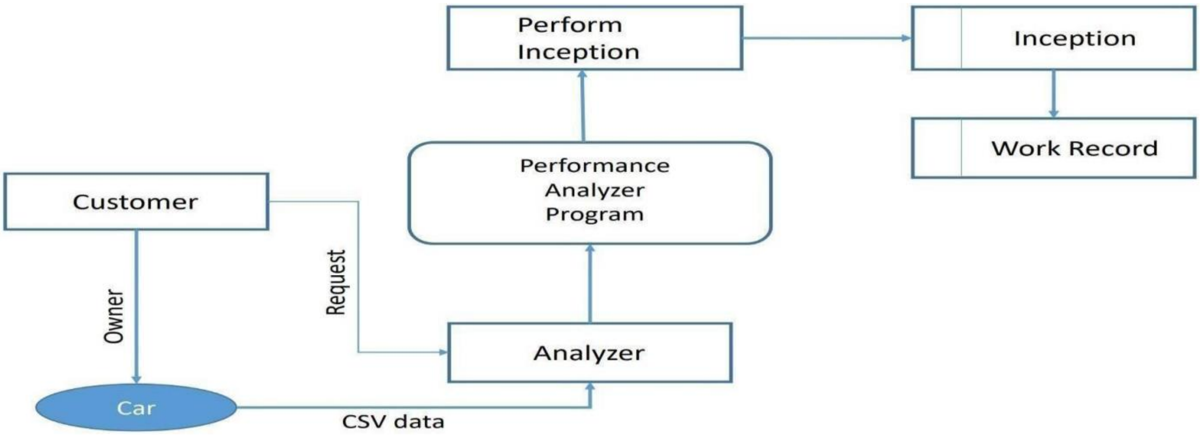
Following are the non-functional requirements of the proposed solution.

NFR-3	Reliability	• Due to the value of data and the potential harm that inaccurate or incomplete data could do, the system will always strive for optimum reliability. • The system will be operational every day of the week, 24 hours a day.
NFR-4	Performance	• The website can efficiently handle traffic by responding to requests right away. • A 64-kbps modem connection would take no longer than 30 seconds to see this webpage (quantitatively, the mean time)
NFR-5	Availability	• Low data redundancy • reduced error risk, quick and effective
NFR-6	Scalability	• A significant number of users must be able to access the system simultaneously because an academic portal is essential to the courses that use it. • The system will likely be most stressed during the admissions season. • Therefore, it must be able to handle several users at once.

5. PROJECT DIAGRAM :

5.1 DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

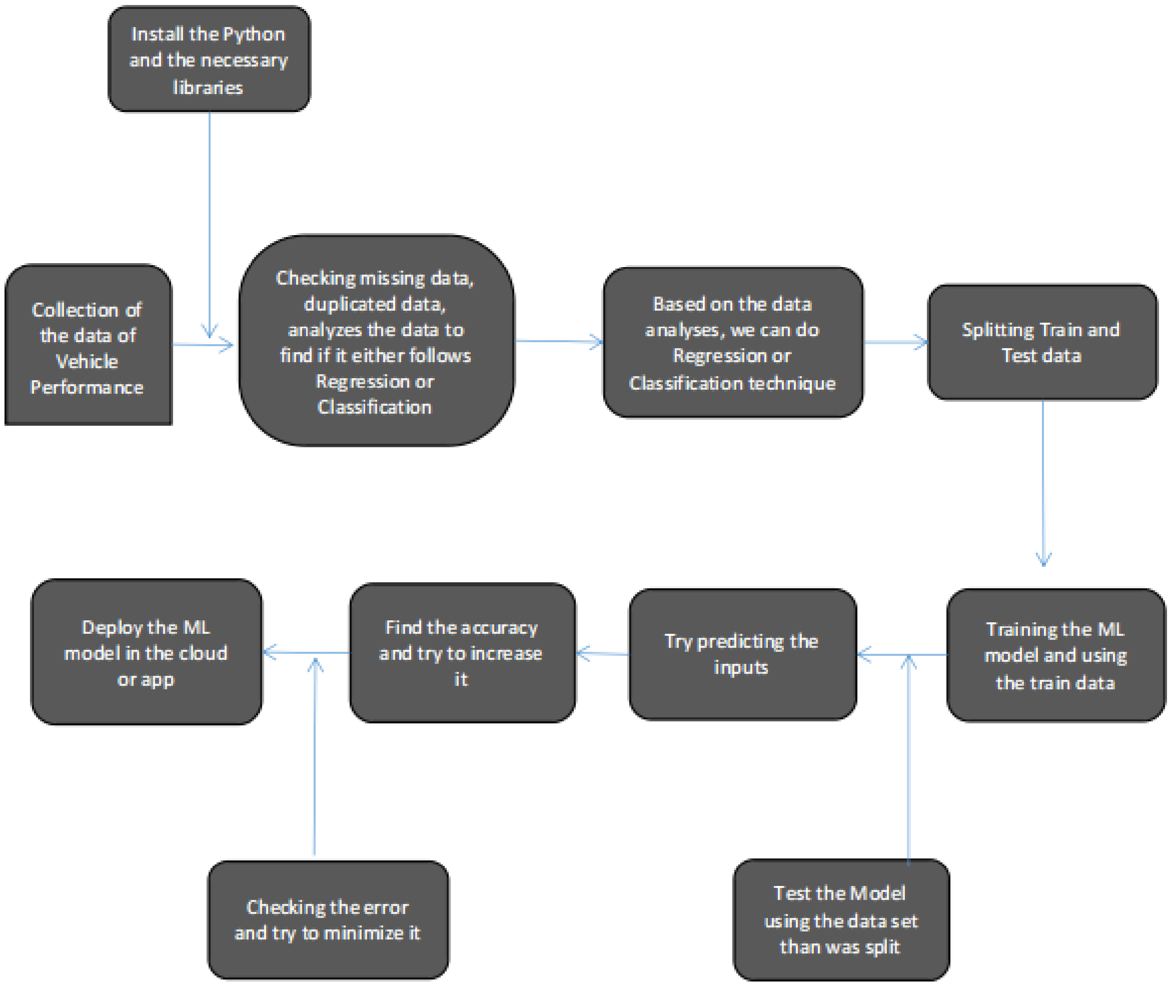


5.2 SOLUTION AND TECHNICAL ARCHITECTURE :

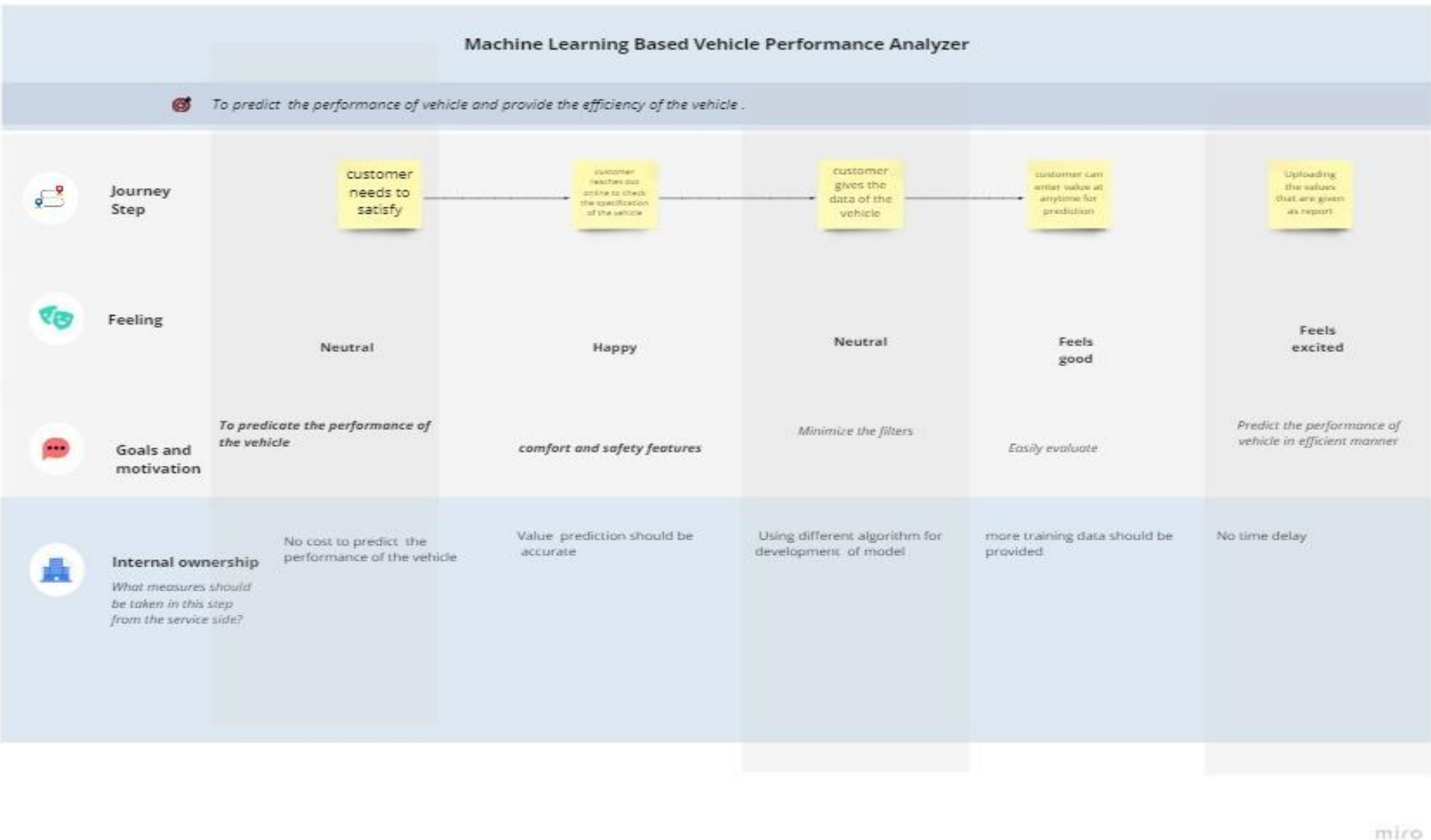
Solution:

predicting the performance level of cars is an important and interesting problem. The main goal of the current study is to predict the performance of the car to improve certain behavior of the vehicle. This can significantly help to improve the systems fuel consumption and increase the efficiency. The performance analysis of the car based on theengine type, no of engine cylinders, fuel type and horsepower etc.These are the factors on which the health of the car can be predicted. It is an on-going process of obtaining, researching, analyzing and recording thehealth based on the above three factors. The performance objectives likemileage, dependability, flexibility and cost can be grouped together to play a vital role in prediction engine and engine management system.
This approach is the very important step towards understanding thevehicles performance

It is an important to analyse the factors using number of well-known approaches of machine learning algorithms like linear regression, decision tree and random forest to improve the vehicle performance efficiency.The range, durability and longevity of automotive traction batteries are ‘hot topics’ in automotive engineering. And here we consider a performance in mileage. To solve this problem, we will develop the models, using the different algorithms and neural networks. We will then see which algorithm predicts car performance(Mileage) withhigher accuracy.



5.3 USER STORIES :



6. PROJECT PLANNING AND SCHEDULING :

6.1 SPRINT PLANNING

StepsTo Perform Predictive Analysis:

Some basicsteps should be performed in order to perform predictive analysis.

Define Problem Statement:

Define the project outcomes, the scope of the effort, objectives, identifythe da tasets that are going to be used.

Data Collection:

Data collection involves gathering the necessary details required for the analysis. It involves the historical or past data from an authorized source over which predictive analysisis to be performed.

Data Cleaning:

Data Cleaning is the process in which we refine our data sets. In the process o f datacleaning, we remove un-
necessary and erroneous data. It involvesremoving the redundantdata and dupl icatedata from our data sets.

Data Analysis:

It involves the exploration of data. We explore the data and analyze it thoroug hly inorder to identify some patterns or new outcomesfrom the data set. In thi s stage, we discoveruseful information and conclude by identifying some patt erns or trends.

Build Predictive Model:

In this stage of predictiveanalysis, we use various algorithmsto build predictiv emodels based on the patternsobserved. It requiresknowledge of python, R, St atistics and MATLAB and so on. We also test our hypothesis using standard s tatisticmodels.

Validation: It is a very importantstep in predictive analysis. In this step, we c heck the efficiencyof our model by performingvarious tests. Here we provides ample input sets to check the validityof our model. The model needs to be ev aluatedforits accuracy in this stage.

Deployment:

In deployment we make our model work in a real environment and it helps in everyday discussion making and make it available to use.

Model Monitoring:

Regularly monitor your models to check performance and ensure that we ha ve proper results.It is seeing how model predictions are

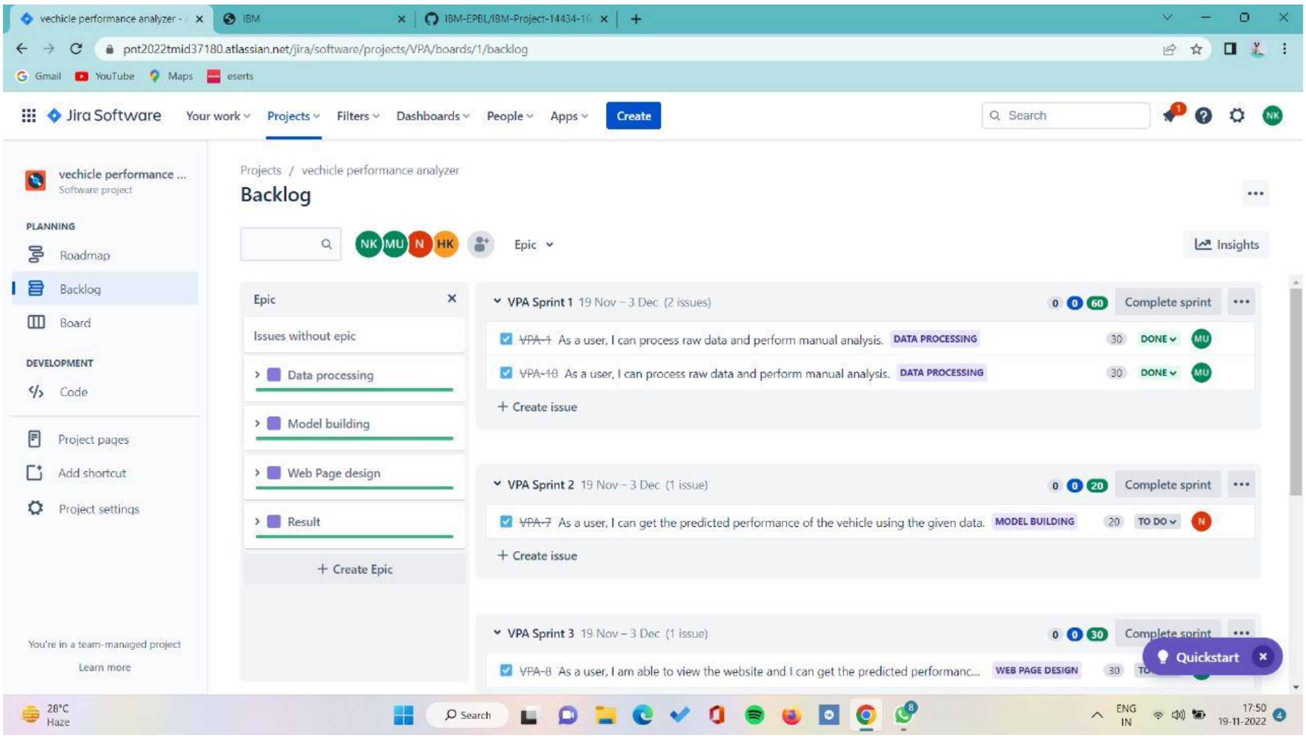
6.2 SPRINT DELIVERY SCHEDULE :

Sprint	Total story point	Duration	Sprint start date	Sprint end date	Story points completed	Sprint release date
Sprint-1	20	6 days	27-oct-2022	28-oct-2022	20	04-nov-2022
Sprint-2	20	6 days	02-nov-2022	05-nov-2022	20	07-nov-2022
Sprint-3	20	6 days	08-nov-2022	12-nov-2022	20	12-nov-2022
Sprint-4	20	6 days	14-nov-2022	19-nov-2022	20	19-nov-2022

Velocity:

Average Velocity = 80/20 =4 story points per day

6.3 Reports from JIRA



7.CODING & SOLUTIONING :

Feature 1 :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.formula.api as smf
```

Importing Dataset

In [5]:

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It
includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='dz2s3XtvfbKpVl_3GgyaPVUrM201qDt4nQUtLNAxaBkY',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-
storage.appdomain.cloud')

bucket = 'machinelearningbasedvehicleperfor-donotdelete-pr-yyybztasqpcsdt'
object_key = 'car performance.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType(
__iter__, body )

dataset = pd.read_csv(body)
dataset.head()
```

Out[5]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino
...	...	...	...	...	...	...	...	...	...
393	27.0	4	140.0	86	2790	15.6	82	1	ford mustang gl
394	44.0	4	97.0	52	2130	24.6	82	2	vw pickup
395	32.0	4	135.0	84	2295	11.6	82	1	dodge rampage
396	28.0	4	120.0	79	2625	18.6	82	1	ford ranger
397	31.0	4	119.0	82	2720	19.4	82	1	chevy s-10

398 rows × 9 columns

Finding missing data

In [6]:

```
dataset.isnull().any()
```

Out[6]:

```
mpg             False
cylinders       False
displacement    False
horsepower      False
weight          False
acceleration    False
model year      False
origin          False
car name        False
dtype: bool
```

There are no null characters in the columns but there is a special character '?' in the 'horsepower' column. So we we replaced '?' with nan and replaced nan values with mean of the column.

In [7]:

```
dataset['horsepower']=dataset['horsepower'].replace('?',np.nan)
```

In [8]:

```
dataset['horsepower'].isnull().sum()
```

Out[8]:

```
0
```

In [9]:

```
dataset['horsepower']=dataset['horsepower'].astype('float64')
```

In [10]:

```
dataset['horsepower'].fillna((dataset['horsepower'].mean()),inplace=True)
```

In [11]:

```
dataset.isnull().any()
```

Out[11]:

```
mpg             False
cylinders       False
displacement    False
horsepower      False
weight          False
acceleration    False
model year      False
origin          False
car name        False
dtype: bool
```

In [12]:

```
dataset.info() #Pandas dataframe.info() function is used to get a quick overview of the dataset.
```

```
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64
 2   displacement  398 non-null    float64
 3   horsepower    398 non-null    float64
 4   weight        398 non-null    int64
 5   acceleration  398 non-null    float64
 6   model year    398 non-null    int64
 7   origin        398 non-null    int64
 8   car name      398 non-null    object
dtypes: float64(4), int64(4), object(1)
memory usage: 28.1+ KB
```

In [13]:

```
dataset.describe() #Pandas describe() is used to view some basic statistical details of a data frame or a series of numeric values.
```

Out[13]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin
count	398.000000	398.000000	398.000000	398.000000	398.000000	398.000000	398.000000	398.000000

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin
mean	23.514573	5.454774	193.425879	104.165829	2970.424623	15.568090	76.010050	1.572864
std	7.815984	1.701004	104.269838	38.298676	846.841774	2.757689	3.697627	0.802055
min	9.000000	3.000000	68.000000	46.000000	1613.000000	8.000000	70.000000	1.000000
25%	17.500000	4.000000	104.250000	75.000000	2223.750000	13.825000	73.000000	1.000000
50%	23.000000	4.000000	148.500000	92.000000	2803.500000	15.500000	76.000000	1.000000
75%	29.000000	8.000000	262.000000	125.000000	3608.000000	17.175000	79.000000	2.000000
max	46.600000	8.000000	455.000000	230.000000	5140.000000	24.800000	82.000000	3.000000

There is no use with car name attribute so drop it

In [14]:

```
dataset=dataset.drop('car name',axis=1) #dropping the unwanted column.
```

In [15]:

```
corr_table=dataset.corr()#Pandas dataframe.corr() is used to find the pairwise correlation of all columns in the dataframe.
corr_table
```

Out[15]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin
mpg	1.000000	-0.775396	-0.804203	-0.777501	-0.831741	0.420289	0.579267	0.563450
cylinders	-0.775396	1.000000	0.950721	0.842437	0.896017	-0.505419	-0.348746	-0.562543
displacement	-0.804203	0.950721	1.000000	0.897082	0.932824	-0.543684	-0.370164	-0.609409

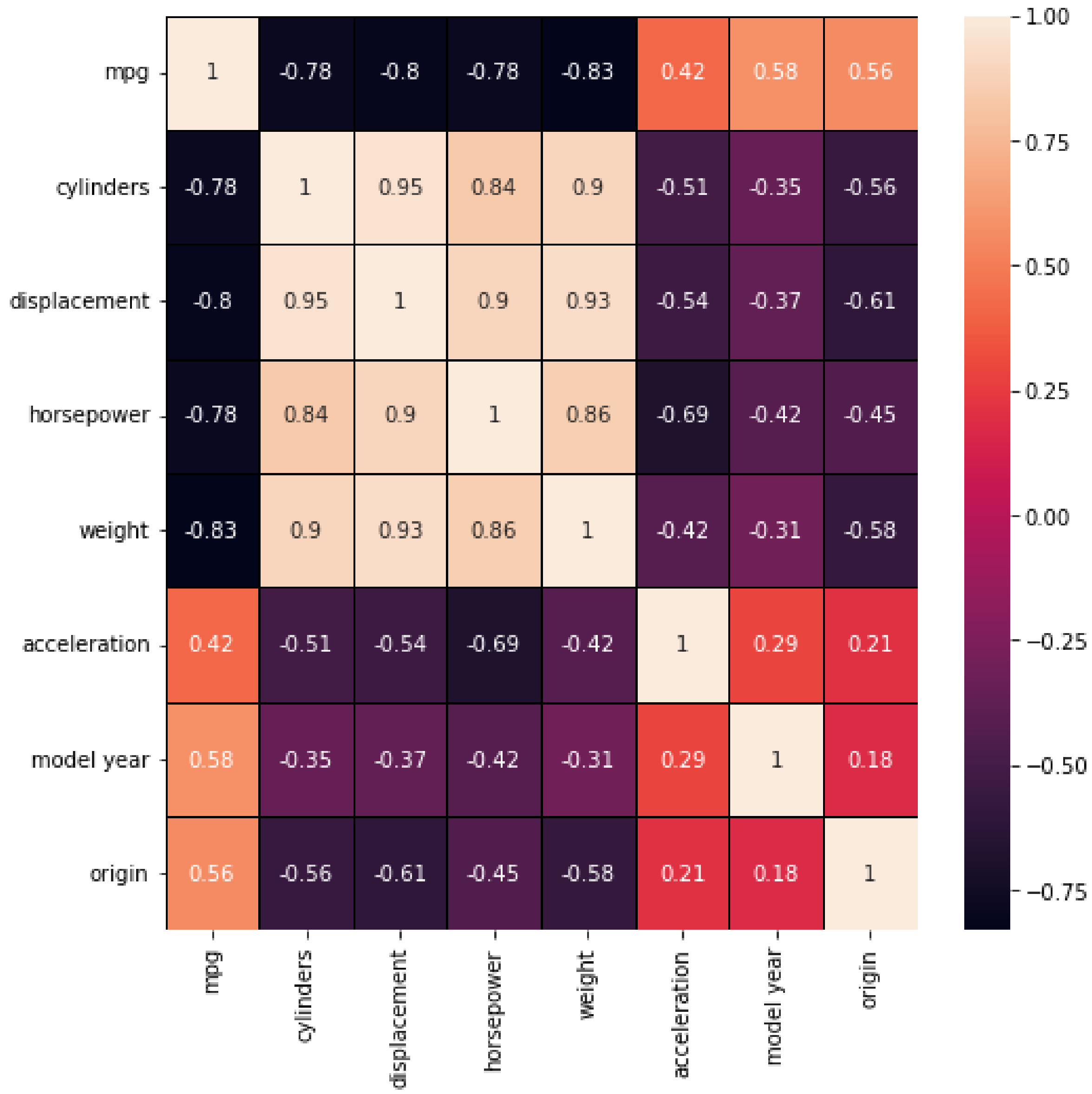
	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin
horsepower	-0.777501	0.842437	0.897082	1.000000	0.863990	-0.686436	-0.417081	-0.452386
weight	-0.831741	0.896017	0.932824	0.863990	1.000000	-0.417457	-0.306564	-0.581024
acceleration	0.420289	-0.505419	-0.543684	-0.686436	-0.417457	1.000000	0.288137	0.205873
model year	0.579267	-0.348746	-0.370164	-0.417081	-0.306564	0.288137	1.000000	0.180662
origin	0.563450	-0.562543	-0.609409	-0.452386	-0.581024	0.205873	0.180662	1.000000

Data Visualizations

Heatmap : which represents correlation between attributes

In [16]:

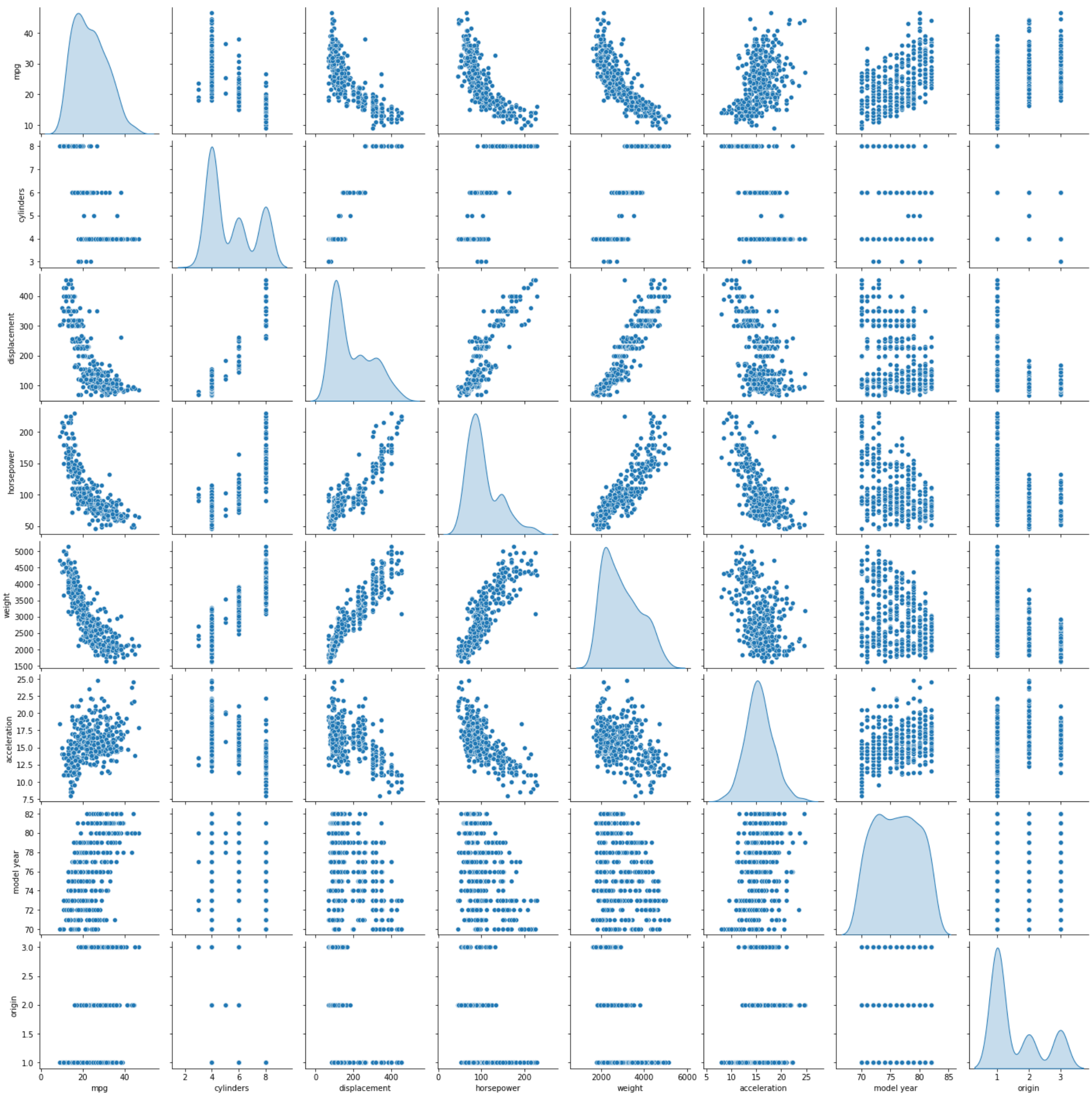
```
sns.heatmap(dataset.corr(),annot=True,linecolor ='black', linewidths =
1)#Heatmap is a way to show some sort of matrix plot,annot is used for
correlation.
fig=plt.gcf()
fig.set_size_inches(8,8)
```



Visualizations of each attributes w.r.t rest of all attributes

In [17]:

```
sns.pairplot(dataset,diag_kind='kde') #pairplot represents pairwise
relation across the entire dataframe.
plt.show()
```

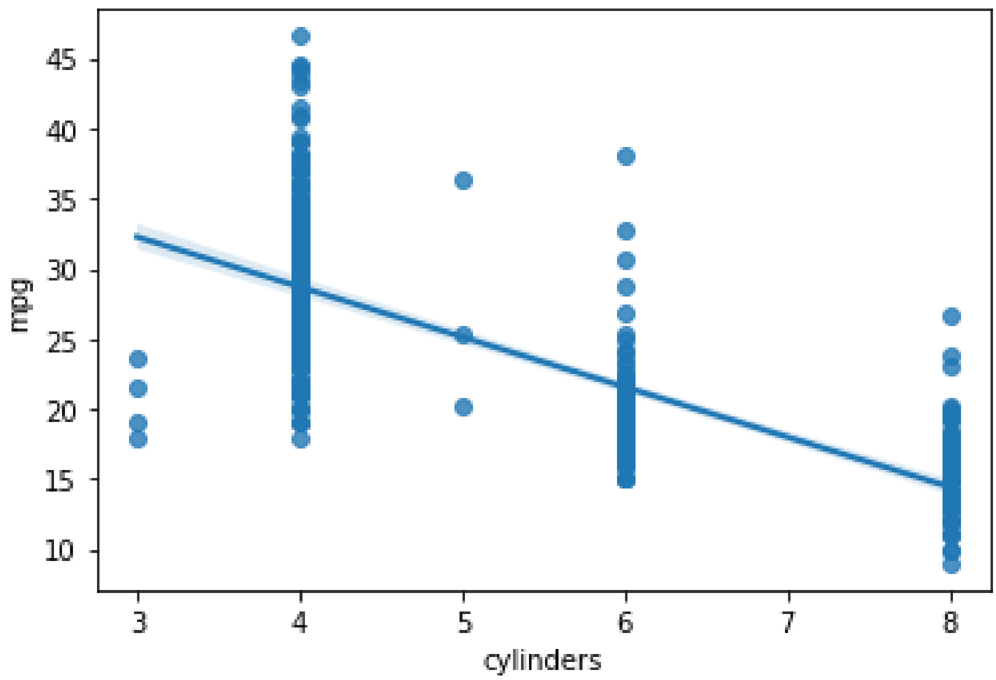


Regression plots(regplot()) creates a regression line between 2 parameters and helps to visualize their linear relationships.

In [18]:

```
sns.regplot(x="cylinders", y="mpg", data=dataset)
```

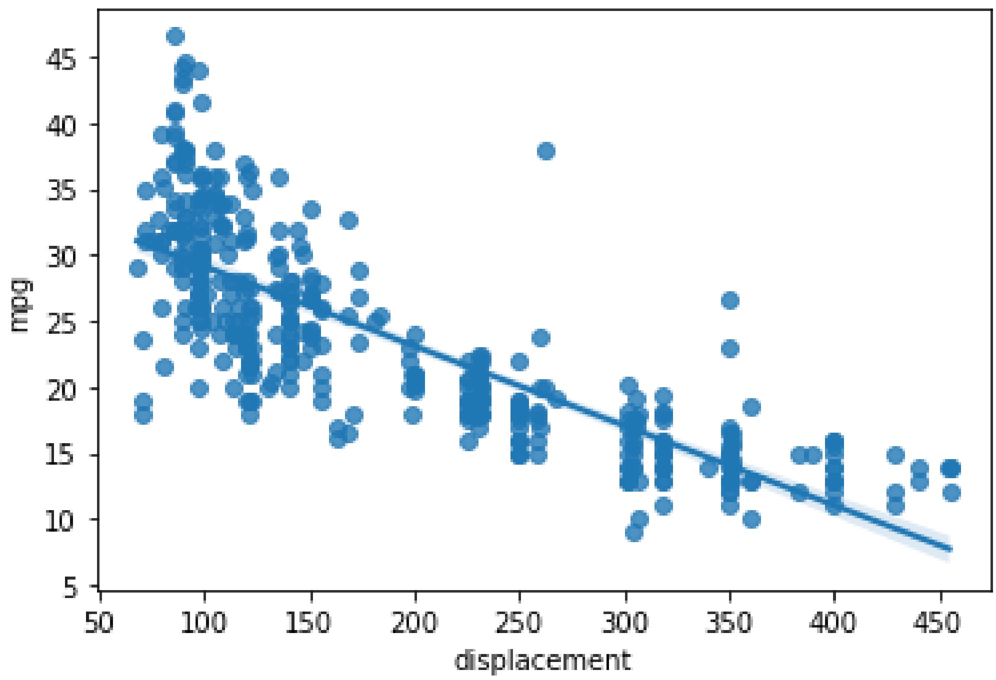
Out[18]:



In [19]:

```
sns.regplot(x="displacement", y="mpg", data=dataset)
```

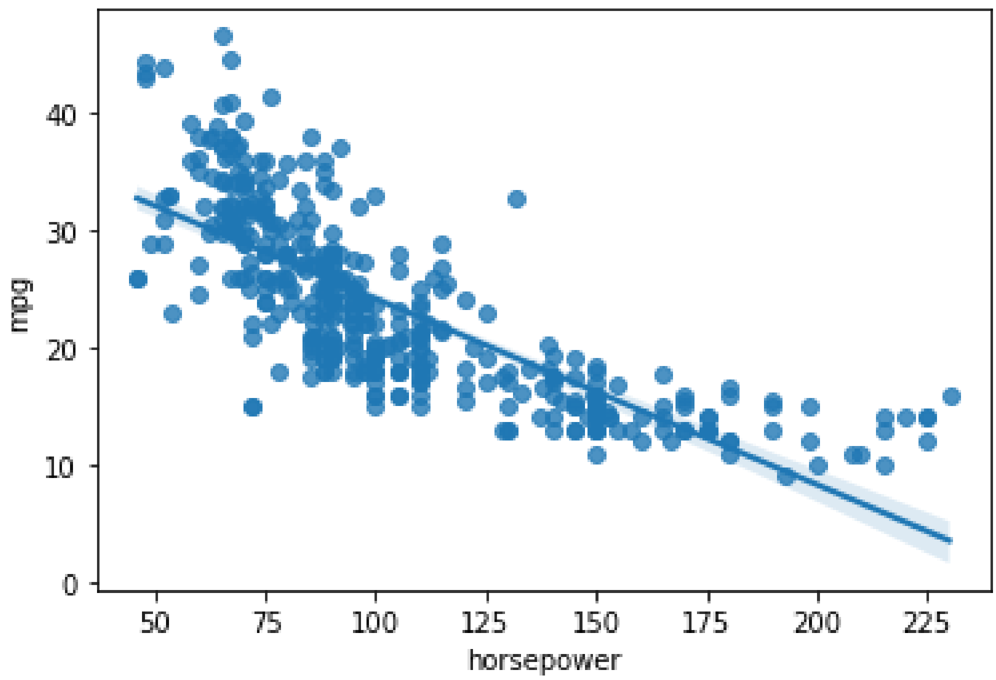
Out[19]:



In [20]:

```
sns.regplot(x="horsepower", y="mpg", data=dataset)
```

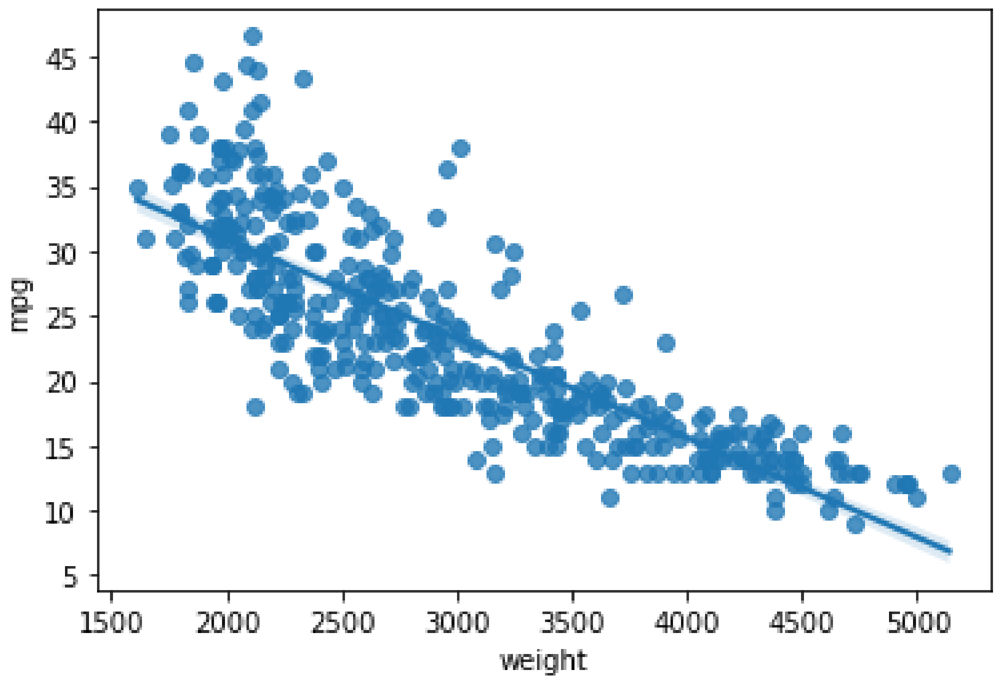
Out[20]:



In [21]:

```
sns.regplot(x="weight", y="mpg", data=dataset)
```

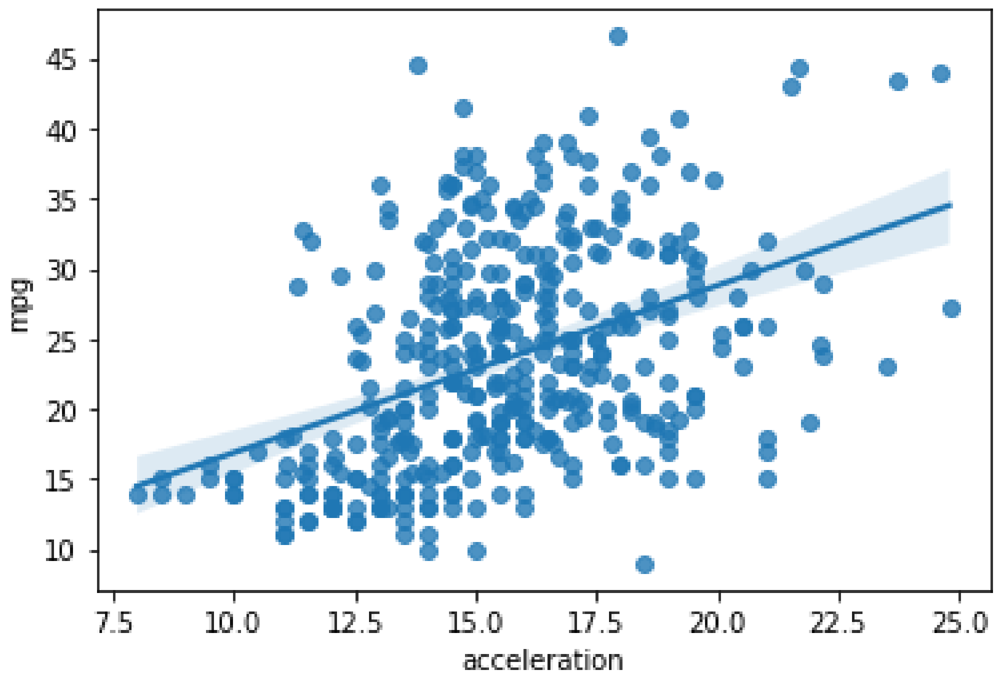
Out[21]:



In [22]:

```
sns.regplot(x="acceleration", y="mpg", data=dataset)
```

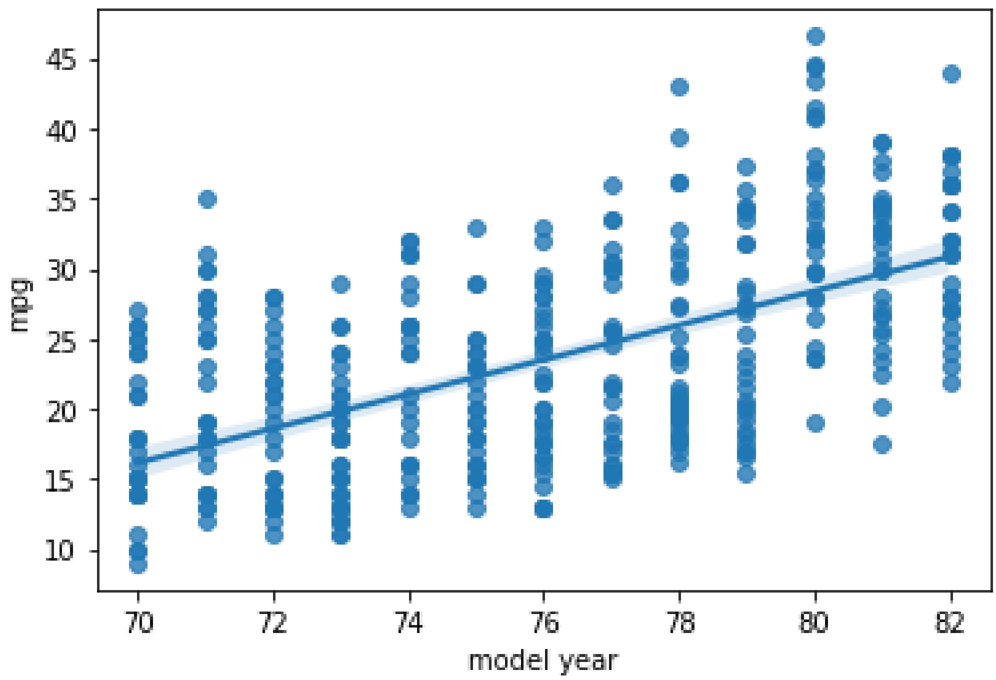
Out[22]:



In [23]:

```
sns.regplot(x="model year", y="mpg", data=dataset)
```

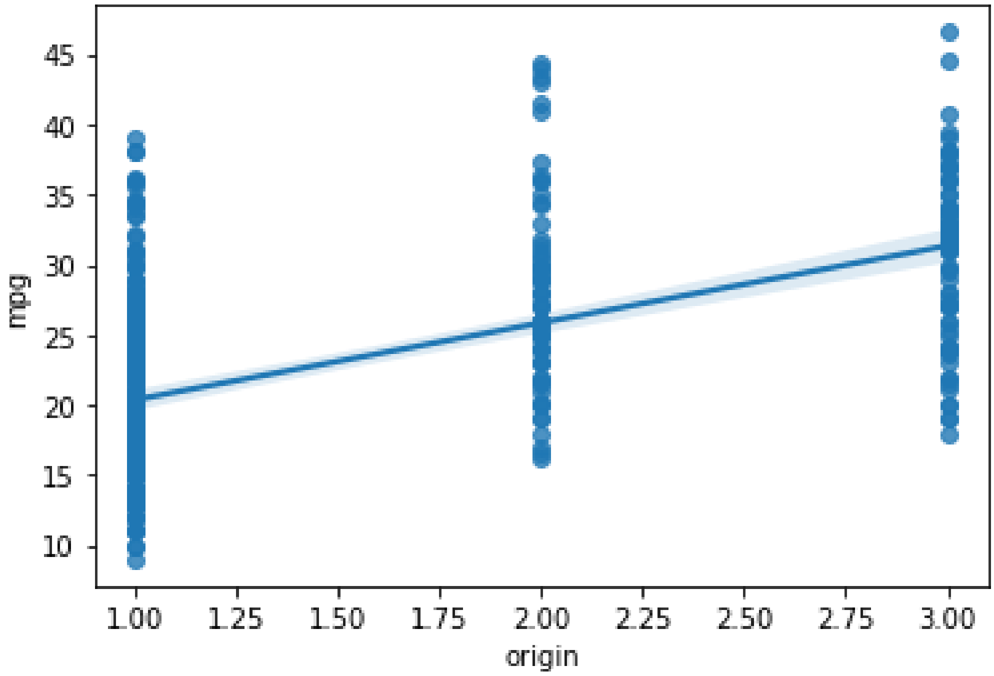
Out[23]:



In [24]:

```
sns.regplot(x="origin", y="mpg", data=dataset)
```

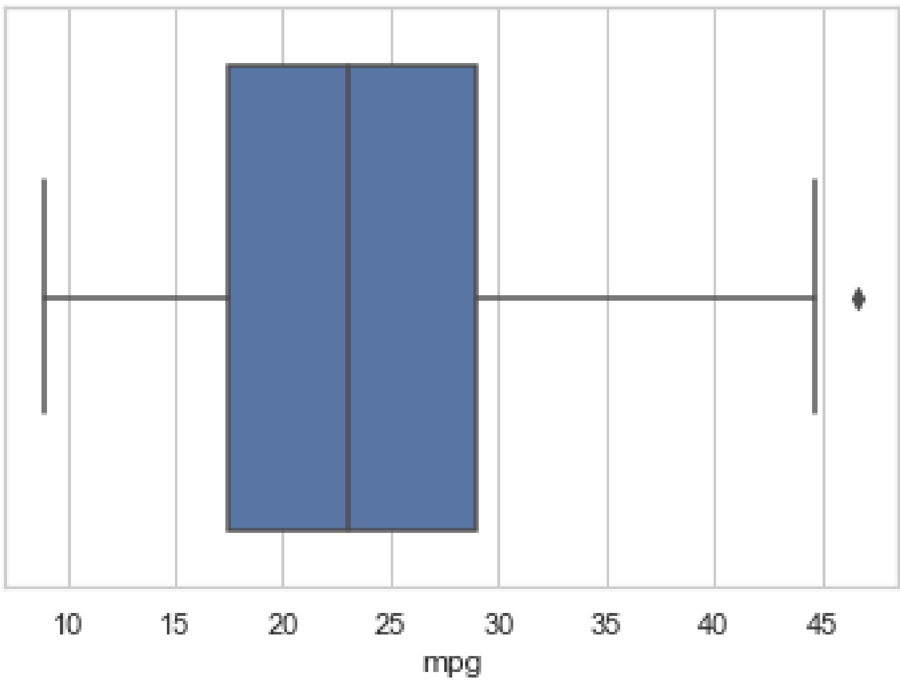
Out[24]:



In [25]:

```
sns.set(style="whitegrid")
sns.boxplot(x=dataset["mpg"])
```

Out[25]:



Finding quartiles for mgp

The P-value is the probability value that the correlation between these two variables is statistically significant.

Normally, we choose a significance level of 0.05, which means that we are 95% confident that the correlation between the variables is significant.

By convention, when the

- p-value is < 0.001: we say there is strong evidence that the correlation is significant.
- the p-value is < 0.05: there is moderate evidence that the correlation is significant.
- the p-value is < 0.1: there is weak evidence that the correlation is significant.
- the p-value is > 0.1: there is no evidence that the correlation is significant.

In [26]:

```
from scipy import stats
```

Cylinders vs mpg

Let's calculate the Pearson Correlation Coefficient and P-value of 'Cylinders' and 'mpg'.

In [27]:

```
pearson_coef, p_value = stats.pearsonr(dataset['cylinders'],
dataset['mpg'])
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-
value of P =", p_value)
```

```
The Pearson Correlation Coefficient is -0.7753962854205543  with a P-value
of P = 4.503992246176927e-81
```

Conclusion:

Since the p-value is < 0.001, the correlation between cylinders and mpg is statistically significant, and the coefficient of ~ -0.775 shows that the relationship is negative and moderately strong.

Displacement vs mpg

Let's calculate the Pearson Correlation Coefficient and P-value of 'Displacement' and 'mpg'.

In [28]:

```
pearson_coef, p_value = stats.pearsonr(dataset['displacement'],
dataset['mpg'])
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-
value of P =", p_value)
```

```
The Pearson Correlation Coefficient is -0.804202824805898  with a P-value o
f P = 1.655888910192639e-91
```

Conclusion:

Since the p-value is < 0.1, the correlation between displacement and mpg is statistically significant, and the linear negative relationship is quite strong (~-0.809, close to -1)

Horsepower vs mpg

Let's calculate the Pearson Correlation Coefficient and P-value of 'horsepower' and 'mpg'.

In [29]:

```
pearson_coef, p_value = stats.pearsonr(dataset['horsepower'], dataset['mpg'])
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value of P =", p_value)
```

```
The Pearson Correlation Coefficient is -0.7775013636276671  with a P-value of P = 8.802190914914203e-82
```

Conclusion:

Since the p-value is < 0.001, the correlation between horsepower and mpg is statistically significant, and the coefficient of ~ -0.771 shows that the relationship is negative and moderately strong.

Weight vs mpg

Let's calculate the Pearson Correlation Coefficient and P-value of 'weight' and 'mpg'.

In [30]:

```
pearson_coef, p_value = stats.pearsonr(dataset['weight'], dataset['mpg'])
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value of P =", p_value)
```

```
The Pearson Correlation Coefficient is -0.831740933244335  with a P-value of P = 2.9727995640500577e-103
```

Conclusion:

Since the p-value is < 0.001, the correlation between weight and mpg is statistically significant, and the linear negative relationship is quite strong (~-0.831, close to -1)

Acceleration vs mpg

Let's calculate the Pearson Correlation Coefficient and P-value of 'Acceleration' and 'mpg'.

In [31]:

```
pearson_coef, p_value = stats.pearsonr(dataset['acceleration'], dataset['mpg'])
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value of P =", p_value)
```

```
The Pearson Correlation Coefficient is 0.42028891210165065  with a P-value of P = 1.823091535078553e-18
```

Conclusion:

Since the p-value is > 0.1, the correlation between acceleration and mpg is statistically significant, but the linear relationship is weak (~0.420).

Model year vs mpg

Let's calculate the Pearson Correlation Coefficient and P-value of 'Model year' and 'mpg'.

In [32]:

```
pearson_coef, p_value = stats.pearsonr(dataset['model year'], dataset['mpg'])
```

```
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value of P =", p_value)
```

```
The Pearson Correlation Coefficient is 0.5792671330833096  with a P-value of P = 4.844935813365483e-37
```

Conclusion:

Since the p-value is < 0.001, the correlation between model year and mpg is statistically significant, but the linear relationship is only moderate (~0.579).

Origin vs mpg

Let's calculate the Pearson Correlation Coefficient and P-value of 'Origin' and 'mpg'.

In [33]:

```
pearson_coef, p_value = stats.pearsonr(dataset['origin'], dataset['mpg'])
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value of P =", p_value)
```

```
The Pearson Correlation Coefficient is 0.5634503597738432  with a P-value of P = 1.0114822102335907e-34
```

Conclusion:

Since the p-value is < 0.001, the correlation between origin and mpg is statistically significant, but the linear relationship is only moderate (~0.563).

Ordinary Least Squares Statistics

In [34]:

```
test=smf.ols('mpg~cylinders+displacement+horsepower+weight+acceleration+origin',dataset).fit()
test.summary()
```

Out[34]:

OLS Regression Results

Dep. Variable:	mpg	**R-squared:**	0.720
Model:	OLS	**Adj. R-squared:**	0.716
Method:	Least Squares	**F-statistic:**	167.6
Date:	Tue, 15 Nov 2022	**Prob (F-statistic):**	8.18e-105
Time:	20:09:09	**Log-Likelihood:**	-1129.2
No. Observations:	398	**AIC:**	2272.
Df Residuals:	391	**BIC:**	2300.

Df Model:	6
Covariance Type:	nonrobust

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	43.4941	2.690	16.171	0.000	38.206	48.782
cylinders	-0.5537	0.402	-1.377	0.169	-1.344	0.237
displacement	0.0125	0.009	1.335	0.183	-0.006	0.031
horsepower	-0.0628	0.017	-3.797	0.000	-0.095	-0.030
weight	-0.0049	0.001	-6.168	0.000	-0.006	-0.003
acceleration	-0.0402	0.121	-0.332	0.740	-0.278	0.198
origin	1.4880	0.345	4.315	0.000	0.810	2.166

Omnibus:	31.632	**Durbin-Watson:**	0.901
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	41.557
Skew:	0.613	**Prob(JB):**	9.46e-10
Kurtosis:	4.002	**Cond. No.**	4.00e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
Inference as in the above summary the p value of the accelaration is maximum(i.e 0.972) so we can remove the acc variable from the dataset

Seperating into Dependent and Independent variables

Independent variables

In [35]:

```
x=dataset[['cylinders','displacement','horsepower','weight','model year','origin']].values
x
```

Out[35]:

```
array([[8.000e+00, 3.070e+02, 1.300e+02, 3.504e+03, 7.000e+01, 1.000e+00],
       [8.000e+00, 3.500e+02, 1.650e+02, 3.693e+03, 7.000e+01, 1.000e+00],
       [8.000e+00, 3.180e+02, 1.500e+02, 3.436e+03, 7.000e+01, 1.000e+00],
       ...,
       [4.000e+00, 1.350e+02, 8.400e+01, 2.295e+03, 8.200e+01, 1.000e+00],
       [4.000e+00, 1.200e+02, 7.900e+01, 2.625e+03, 8.200e+01, 1.000e+00],
       [4.000e+00, 1.190e+02, 8.200e+01, 2.720e+03, 8.200e+01, 1.000e+00]])
```

Dependent variables

In [36]:

```
y=dataset.iloc[:,0:1].values
y
```

Out[36]:

```
array([[18. ],
       [15. ],
       [18. ],
       [16. ],
       [17. ],
       [15. ],
       [14. ],
       [14. ],
       [14. ],
       [15. ],
       [15. ],
       [14. ],
       [15. ],
       [14. ],
       [24. ],
       [22. ],
       [18. ],
       [21. ],
       [27. ],
       [26. ],
       [25. ],
       [24. ],
       [25. ],
       [26. ],
       [21. ],
       [10. ],
       [10. ],
       [11. ],
       [ 9. ],
       [27. ],
       [28. ],
       [25. ],
       [25. ],
       [19. ],
       [16. ],
       [17. ],
       [19. ],
```

```
[18. ],
[14. ],
[14. ],
[14. ],
[14. ],
[12. ],
[13. ],
[13. ],
[18. ],
[22. ],
[19. ],
[18. ],
[23. ],
[28. ],
[30. ],
[30. ],
[31. ],
[35. ],
[27. ],
[26. ],
[24. ],
[25. ],
[23. ],
[20. ],
[21. ],
[13. ],
[14. ],
[15. ],
[14. ],
[17. ],
[11. ],
[13. ],
[12. ],
[13. ],
[19. ],
[15. ],
[13. ],
[13. ],
[14. ],
[18. ],
[22. ],
[21. ],
[26. ],
[22. ],
[28. ],
[23. ],
[28. ],
[27. ],
[13. ],
[14. ],
[13. ],
[14. ],
[15. ],
[12. ],
[13. ],
[13. ],
[14. ],
```

```
[13. ],
[12. ],
[13. ],
[18. ],
[16. ],
[18. ],
[18. ],
[23. ],
[26. ],
[11. ],
[12. ],
[13. ],
[12. ],
[18. ],
[20. ],
[21. ],
[22. ],
[18. ],
[19. ],
[21. ],
[26. ],
[15. ],
[16. ],
[29. ],
[24. ],
[20. ],
[19. ],
[15. ],
[24. ],
[20. ],
[11. ],
[20. ],
[21. ],
[19. ],
[15. ],
[31. ],
[26. ],
[32. ],
[25. ],
[16. ],
[16. ],
[18. ],
[16. ],
[13. ],
[14. ],
[14. ],
[14. ],
[29. ],
[26. ],
[26. ],
[31. ],
[32. ],
[28. ],
[24. ],
[26. ],
[24. ],
[26. ],
```

```
[31. ],
[19. ],
[18. ],
[15. ],
[15. ],
[16. ],
[15. ],
[16. ],
[14. ],
[17. ],
[16. ],
[15. ],
[18. ],
[21. ],
[20. ],
[13. ],
[29. ],
[23. ],
[20. ],
[23. ],
[24. ],
[25. ],
[24. ],
[18. ],
[29. ],
[19. ],
[23. ],
[23. ],
[22. ],
[25. ],
[33. ],
[28. ],
[25. ],
[25. ],
[26. ],
[27. ],
[17.5],
[16. ],
[15.5],
[14.5],
[22. ],
[22. ],
[24. ],
[22.5],
[29. ],
[24.5],
[29. ],
[33. ],
[20. ],
[18. ],
[18.5],
[17.5],
[29.5],
[32. ],
[28. ],
[26.5],
[20. ],
```

```
[13. ],
[19. ],
[19. ],
[16.5],
[16.5],
[13. ],
[13. ],
[13. ],
[31.5],
[30. ],
[36. ],
[25.5],
[33.5],
[17.5],
[17. ],
[15.5],
[15. ],
[17.5],
[20.5],
[19. ],
[18.5],
[16. ],
[15.5],
[15.5],
[16. ],
[29. ],
[24.5],
[26. ],
[25.5],
[30.5],
[33.5],
[30. ],
[30.5],
[22. ],
[21.5],
[21.5],
[43.1],
[36.1],
[32.8],
[39.4],
[36.1],
[19.9],
[19.4],
[20.2],
[19.2],
[20.5],
[20.2],
[25.1],
[20.5],
[19.4],
[20.6],
[20.8],
[18.6],
[18.1],
[19.2],
[17.7],
[18.1],
```

```
[17.5],
[30. ],
[27.5],
[27.2],
[30.9],
[21.1],
[23.2],
[23.8],
[23.9],
[20.3],
[17. ],
[21.6],
[16.2],
[31.5],
[29.5],
[21.5],
[19.8],
[22.3],
[20.2],
[20.6],
[17. ],
[17.6],
[16.5],
[18.2],
[16.9],
[15.5],
[19.2],
[18.5],
[31.9],
[34.1],
[35.7],
[27.4],
[25.4],
[23. ],
[27.2],
[23.9],
[34.2],
[34.5],
[31.8],
[37.3],
[28.4],
[28.8],
[26.8],
[33.5],
[41.5],
[38.1],
[32.1],
[37.2],
[28. ],
[26.4],
[24.3],
[19.1],
[34.3],
[29.8],
[31.3],
[37. ],
[32.2],
```

```
[46.6],
[27.9],
[40.8],
[44.3],
[43.4],
[36.4],
[30. ],
[44.6],
[40.9],
[33.8],
[29.8],
[32.7],
[23.7],
[35. ],
[23.6],
[32.4],
[27.2],
[26.6],
[25.8],
[23.5],
[30. ],
[39.1],
[39. ],
[35.1],
[32.3],
[37. ],
[37.7],
[34.1],
[34.7],
[34.4],
[29.9],
[33. ],
[34.5],
[33.7],
[32.4],
[32.9],
[31.6],
[28.1],
[30.7],
[25.4],
[24.2],
[22.4],
[26.6],
[20.2],
[17.6],
[28. ],
[27. ],
[34. ],
[31. ],
[29. ],
[27. ],
[24. ],
[23. ],
[36. ],
[37. ],
[31. ],
[38. ],
```

```
       [36. ],
       [36. ],
       [36. ],
       [34. ],
       [38. ],
       [32. ],
       [38. ],
       [25. ],
       [38. ],
       [26. ],
       [22. ],
       [32. ],
       [36. ],
       [27. ],
       [27. ],
       [44. ],
       [32. ],
       [28. ],
       [31. ]])
```

Splitting into train and test data.

In [37]:

```
from sklearn.model_selection import train_test_split
```

In [38]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.1,random_sta
te=0)
```

we are splitting as 90% train data and 10% test data

Normalisation

In [39]:

```
from sklearn.preprocessing import StandardScaler
sd = StandardScaler()
x_train = sd.fit_transform(x_train)
x_test = sd.fit_transform(x_test)
y_train = sd.fit_transform(y_train)
y_test = sd.fit_transform(y_test)

x_train
```

Out[39]:

```
array([[ 1.46858608,  2.48230464,  2.97979869,  1.62455076, -1.61295698,
        -0.71873488],
       [ 1.46858608,  1.48729292,  1.55782064,  0.84358808, -1.61295698,
        -0.71873488],
       [-0.86550411, -0.70364636, -0.63978179, -0.36507278,  0.82235108,
        -0.71873488],
       ...,
       [-0.86550411, -1.21071964, -1.44126033, -1.31380657, -0.80118763,
         0.53032865],
       [ 0.30154098,  0.53055088, -0.12269887,  0.35799706, -1.3423672 ,
```

```
 -0.71873488],
[-0.86550411, -1.00023639, -0.87246911, -0.89319732, -0.26000806,
  0.53032865]])
```

Decision tree regressor

In [40]:

```
from sklearn.tree import DecisionTreeRegressor
dt=DecisionTreeRegressor(random_state=0,criterion="mae")
dt.fit(x_train,y_train)
```

```
C:\Users\Max_l\anaconda3\lib\site-packages\sklearn\tree\_classes.py:366: Fu
tureWarning: Criterion 'mae' was deprecated in v1.0 and will be removed in
version 1.2. Use `criterion='absolute_error'` which is equivalent.
  warnings.warn(
```

Out[40]:

```
DecisionTreeRegressor(criterion='mae', random_state=0)
```

In [41]:

```
import pickle
pickle.dump(dt,open('decision_model.pkl','wb'))
```

In [42]:

```
y_pred=dt.predict(x_test)
y_pred
```

Out[42]:

```
array([-1.21248604, -0.17830889, -1.21248604, -0.43685318, -0.82466961,
        0.98514039,  2.07102639, -0.17830889, -1.08321389,  0.20950753,
        2.74324153,  2.74324153, -0.43685318,  0.33877968, -0.95394175,
        1.6315011 ,  0.46805182,  0.31292525, -0.82466961,  1.24368468,
       -0.95394175, -0.04903675, -0.24294497, -0.56612532,  1.39881125,
        0.20950753,  0.79123218,  0.98514039,  0.98514039, -0.43685318,
       -0.88930568,  1.11441253, -1.08321389,  1.04977646, -0.54027089,
        0.05438096, -1.08321389, -0.95394175,  0.79123218, -1.60030246])
```

In [43]:

```
y_test
```

Out[43]:

```
array([[-1.29002284],
       [ 0.03307751],
       [-1.41030469],
       [-0.44804989],
       [-0.80889544],
       [ 1.23589601],
       [ 1.12764234],
       [-1.16974099],
       [-0.14734527],
       [ 1.94555892],
       [ 1.50051608],
       [-0.80889544],
       [-0.20748619],
       [-1.10960007],
       [ 1.3200933 ],
       [ 0.75476861],
       [ 0.27364121],
       [-0.80889544],
```

```
       [ 1.51254426],
       [-1.10960007],
       [-0.20748619],
       [-0.08720434],
       [-0.80889544],
       [ 1.17575508],
       [ 0.08119025],
       [ 1.36820604],
       [ 1.11561416],
       [ 0.63448676],
       [-1.04945914],
       [-0.73672633],
       [ 1.47645971],
       [-1.16974099],
       [ 1.05547323],
       [-0.2796553 ],
       [-0.08720434],
       [-0.68861359],
       [-0.94120548],
       [ 0.86302227],
       [-1.53058654]])
```

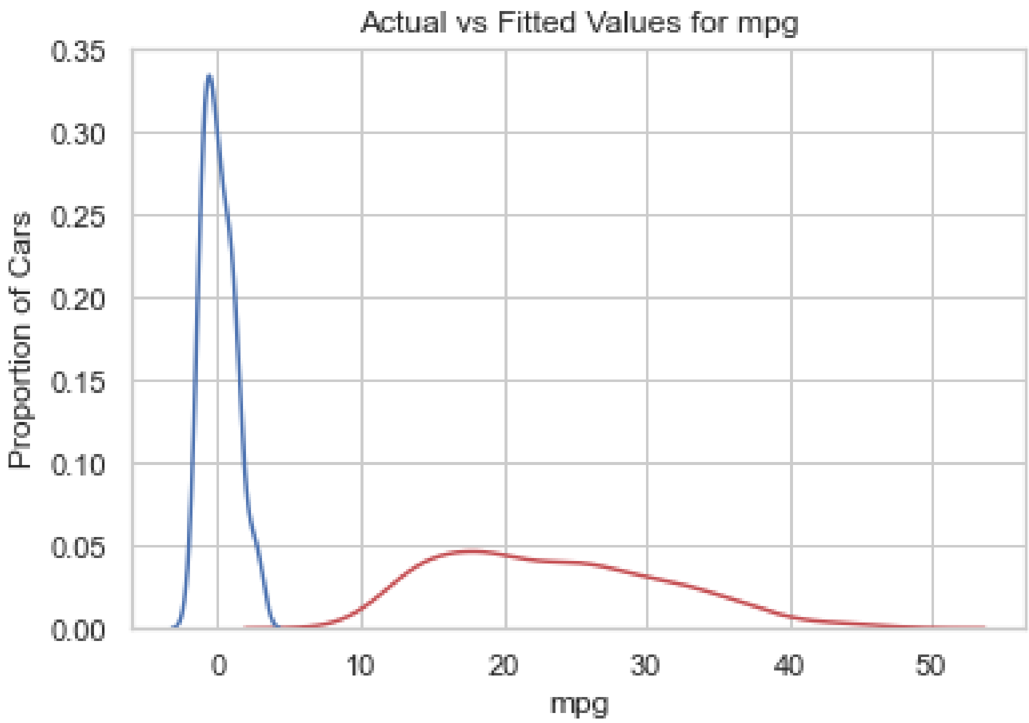
In [44]:

```
ax1 = sns.distplot(dataset['mpg'], hist=False, color="r", label="Actual
Value")
sns.distplot(y_pred, hist=False, color="b", label="Fitted Values" , ax=ax1)


plt.title('Actual vs Fitted Values for mpg')
plt.xlabel('mpg')
plt.ylabel('Proportion of Cars')

plt.show()
plt.close()
```

```
C:\Users\Max_l\anaconda3\lib\site-packages\seaborn\distributions.py:2619: F
utureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-le
vel function with similar flexibility) or `kdeplot` (an axes-level function
 for kernel density plots).
  warnings.warn(msg, FutureWarning)
C:\Users\Max_l\anaconda3\lib\site-packages\seaborn\distributions.py:2619: F
utureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-le
vel function with similar flexibility) or `kdeplot` (an axes-level function
 for kernel density plots).
  warnings.warn(msg, FutureWarning)
```



We can see that the fitted values are reasonably close to the actual values, since the two distributions overlap a bit. However, there is definitely some room for improvement.

R-squared

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

R-squared = Explained variation / Total variation

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).

In [45]:

```
from sklearn.metrics import r2_score,mean_squared_error
```

In [46]:

```
r2_score(y_test,y_pred)
```

Out[46]:

```
0.8578094522360582
```

In [47]:

```
mean_squared_error(y_test,y_pred)
```

Out[47]:

```
0.14219054776394183
```

In [48]:

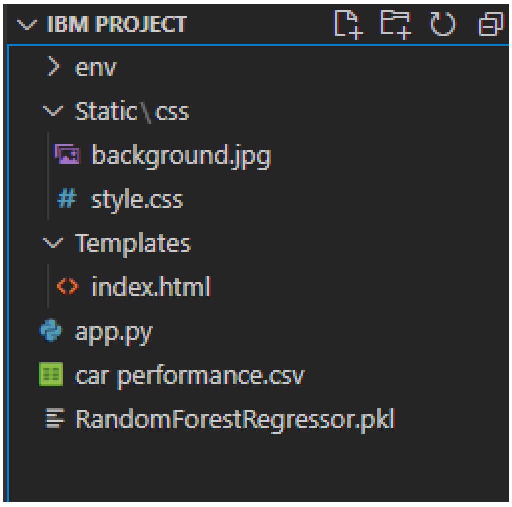
```
np.sqrt(mean_squared_error(y_test,y_pred))
```

Out[48]:

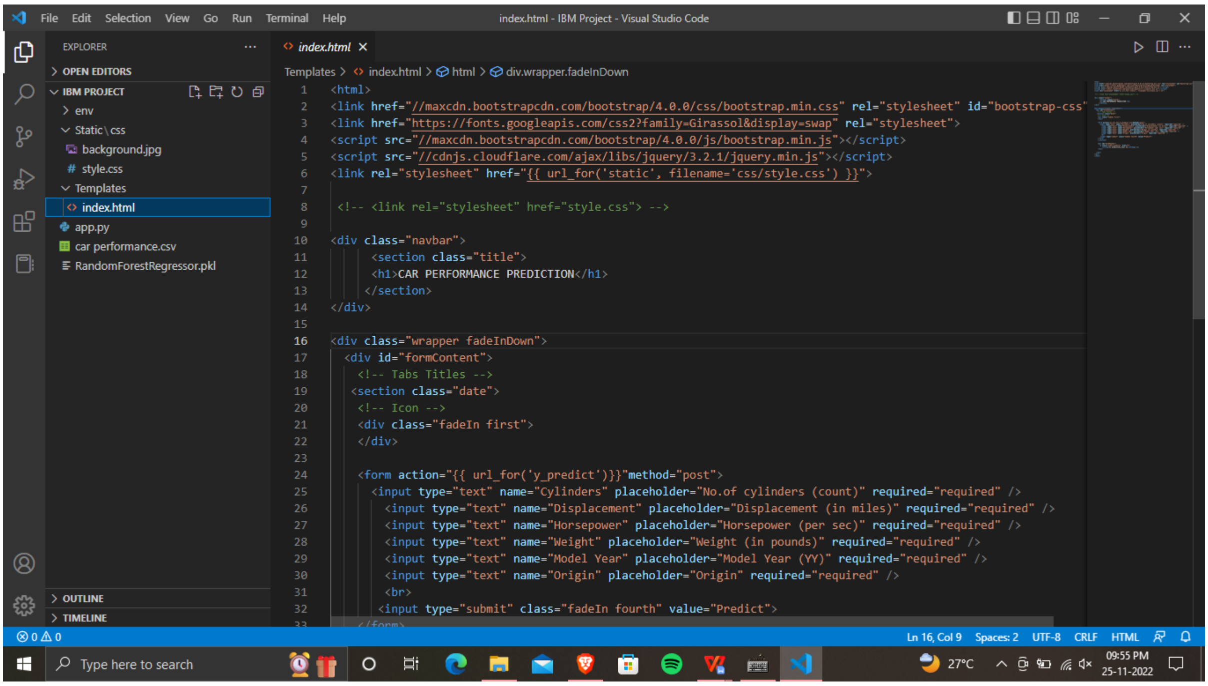
```
0.377081619498938
```

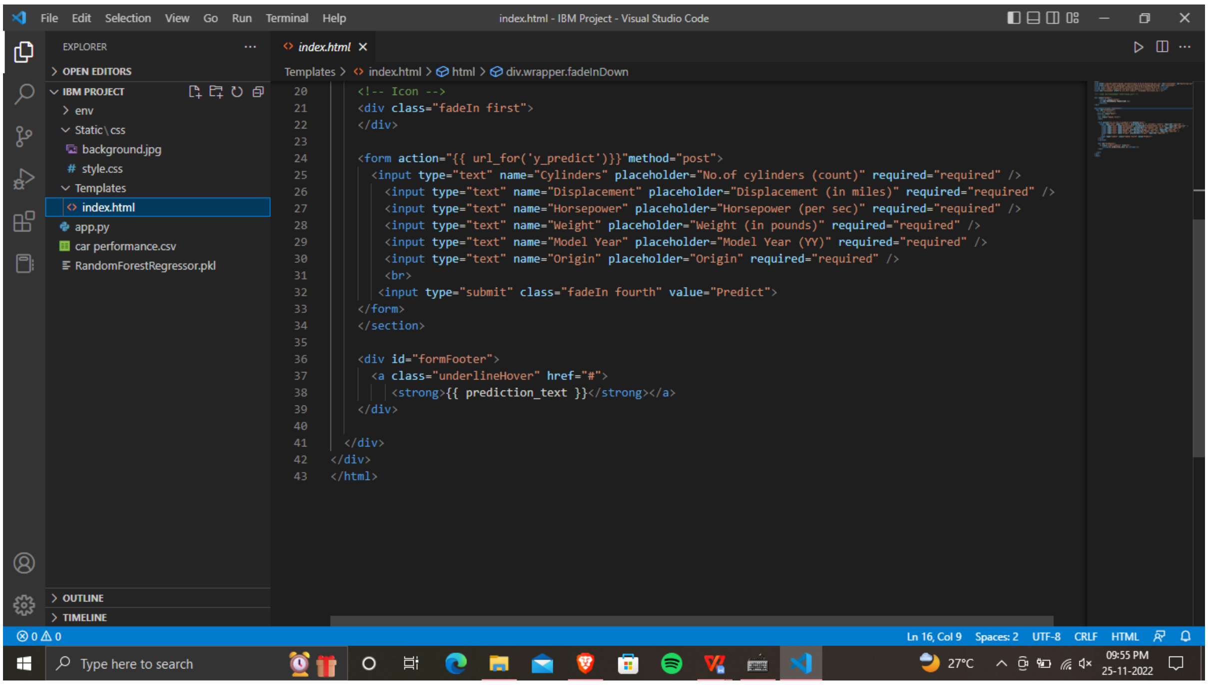
Feature 2 :

Application Building :



Homepage HTML code :



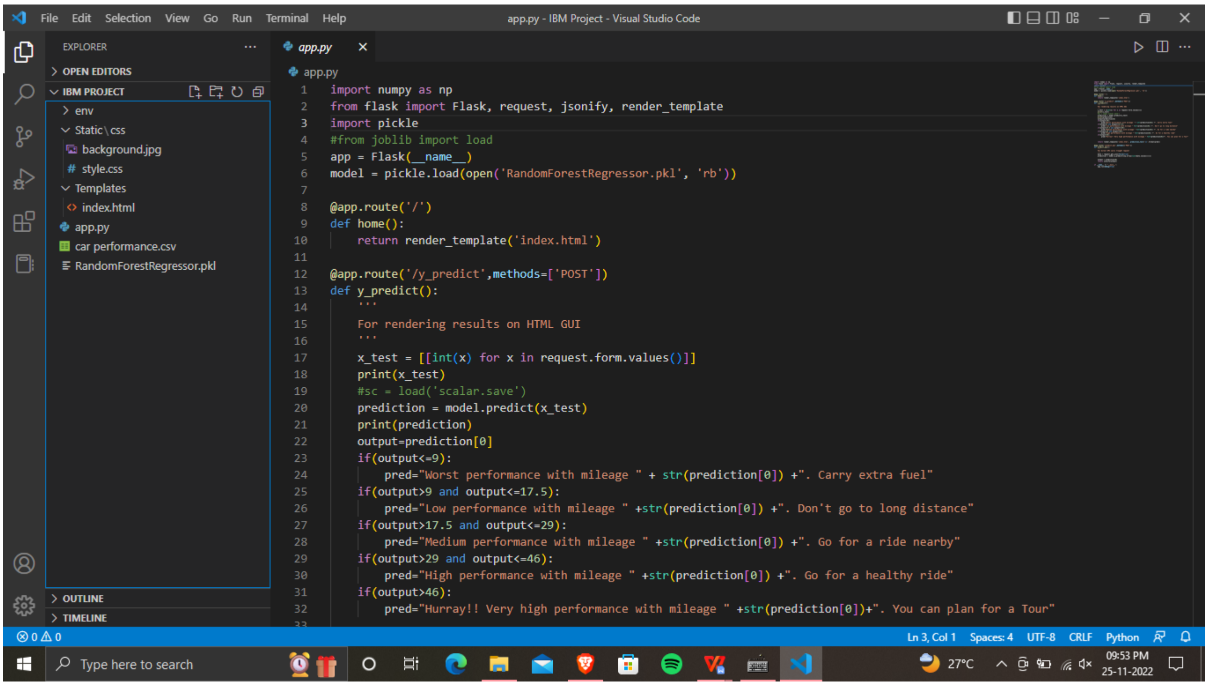


index.html - IBM Project - Visual Studio Code
<!-- Icon -->
<div class="fadeIn first">
</div>
<form action="{{ url_for('y_predict')}}"method="post">
<input type="text" name="Cylinders" placeholder="No.of cylinders (count)" required="required" />
<input type="text" name="Displacement" placeholder="Displacement (in miles)" required="required" />
<input type="text" name="Horsepower" placeholder="Horsepower (per sec)" required="required" />
<input type="text" name="Weight" placeholder="Weight (in pounds)" required="required" />
<input type="text" name="Model Year" placeholder="Model Year (YY)" required="required" />
<input type="text" name="Origin" placeholder="Origin" required="required" />

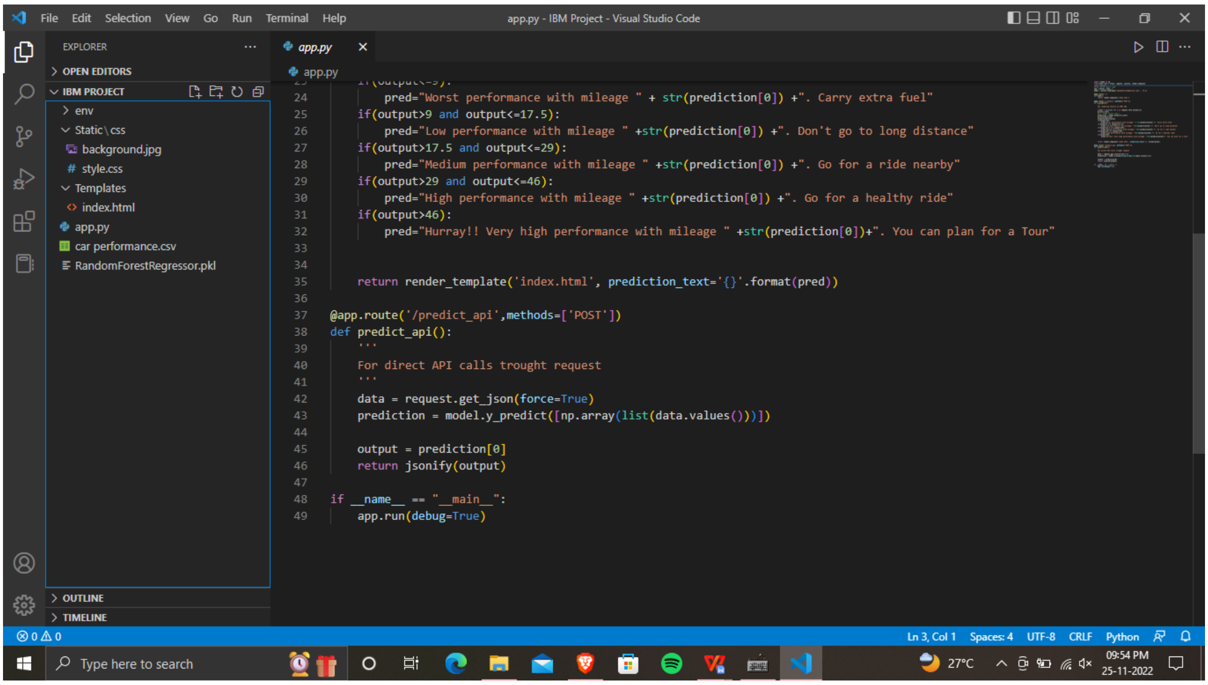
<input type="submit" class="fadeIn fourth" value="Predict">
</form>
</section>
<div id="formFooter">

{{ prediction_text }}
</div>
</div>
</div>
</html>

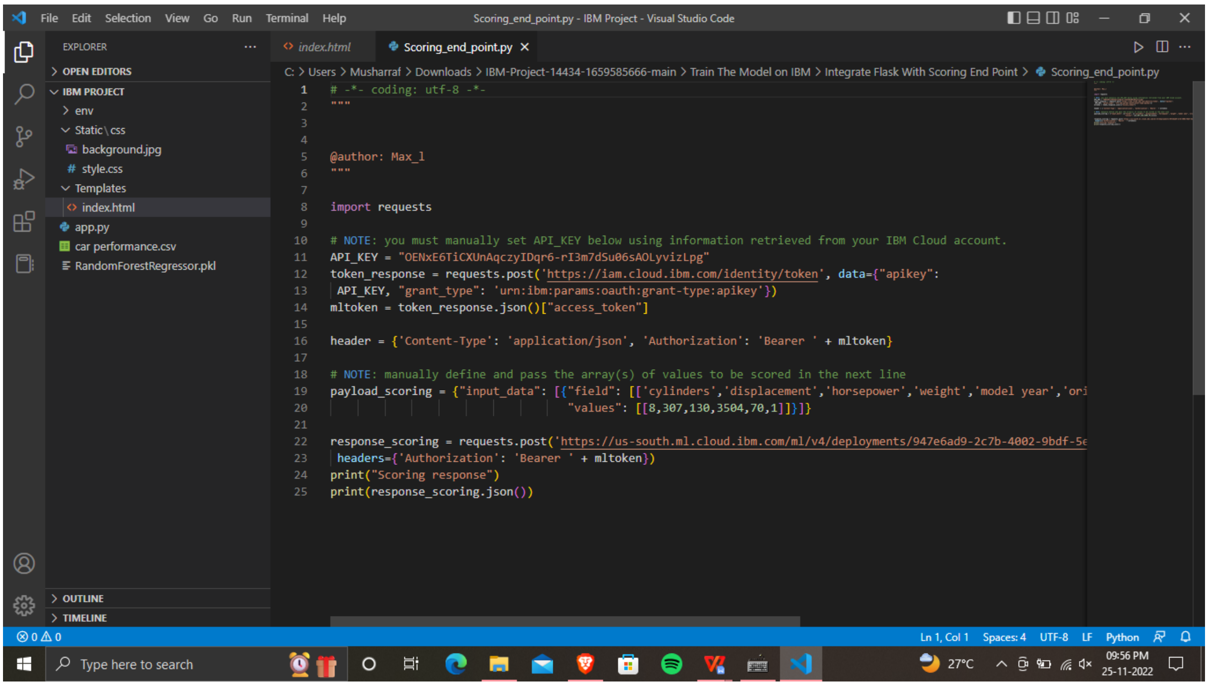
Flask Code :



app.py - IBM Project - Visual Studio Code
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
#from joblib import load
app = Flask(__name__)
model = pickle.load(open('RandomForestRegressor.pkl', 'rb'))
@app.route('/')
def home():
return render_template('index.html')
@app.route('/y_predict',methods=['POST'])
def y_predict():
'''
For rendering results on HTML GUI
'''
x_test = [[int(x) for x in request.form.values()]]
print(x_test)
#sc = load('scalar.save')
prediction = model.predict(x_test)
print(prediction)
output=prediction[0]
if(output<=9):
pred="Worst performance with mileage " + str(prediction[0]) +". Carry extra fuel"
if(output>9 and output<=17.5):
pred="Low performance with mileage " +str(prediction[0]) +". Don't go to long distance"
if(output>17.5 and output<=29):
pred="Medium performance with mileage " +str(prediction[0]) +". Go for a ride nearby"
if(output>29 and output<=46):
pred="High performance with mileage " +str(prediction[0]) +". Go for a healthy ride"
if(output>46):
pred="Hurray!! Very high performance with mileage " +str(prediction[0])+". You can plan for a Tour"



Integrate Flask Code :



UNIT TESTING:

Unit testing is carried out screen-wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to locate errors .This has enabled the detection of errors in coding and logic .This is the first level of testing.In this, codes are written such that from one module ,we can move on to the next module according to the choice we enter.

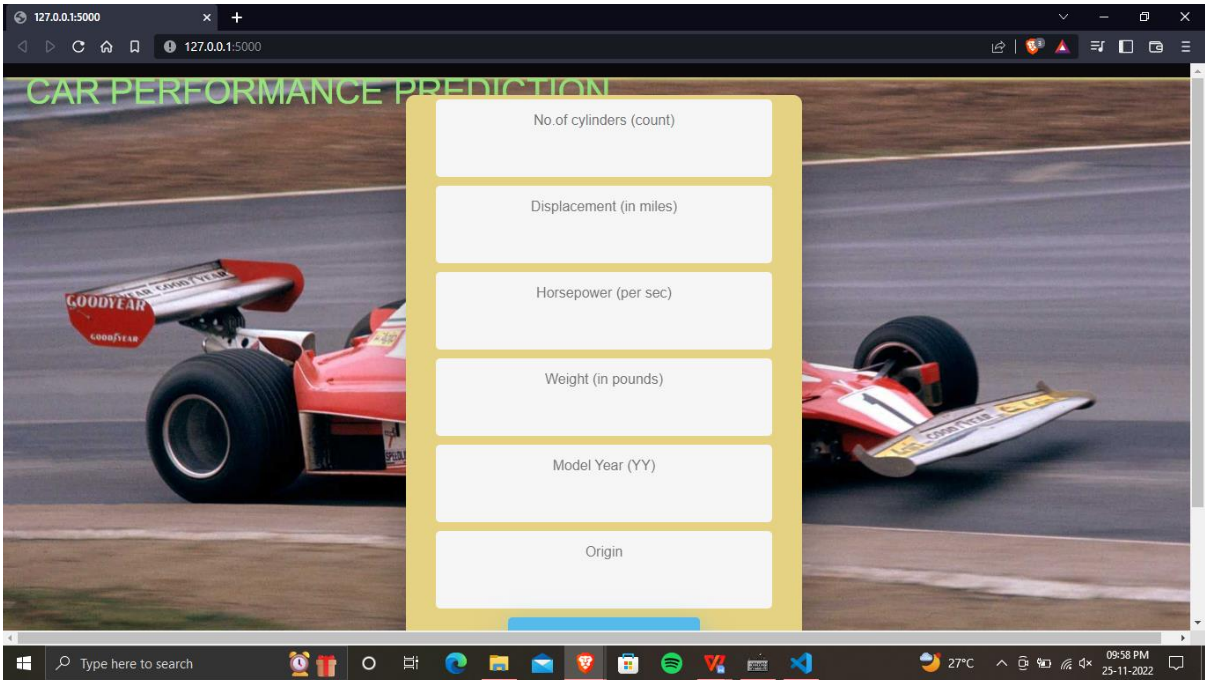
SYSTEM TESTING:

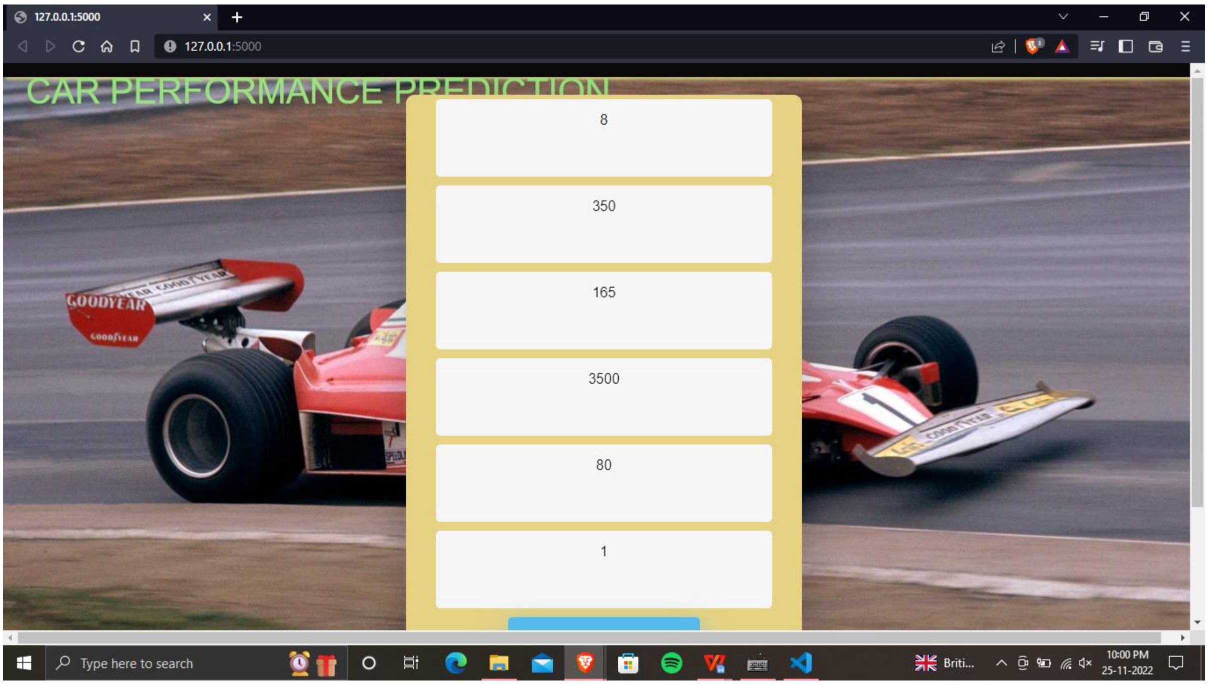
In this, the entire system was tested as a whole with all forms, code, modules and class modules .System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences.It is a series of different tests that verifies that all system elements have been properly integrated and perform allocated functions.System testing makes logical assumptions that if all parts of the system are correct, the goal will be successfully achieved. Testing is the process of executing the program with the intent of finding errors. Testing cannot show the absence of defects, it can only show that software errors are present.

8.1 Test cases :

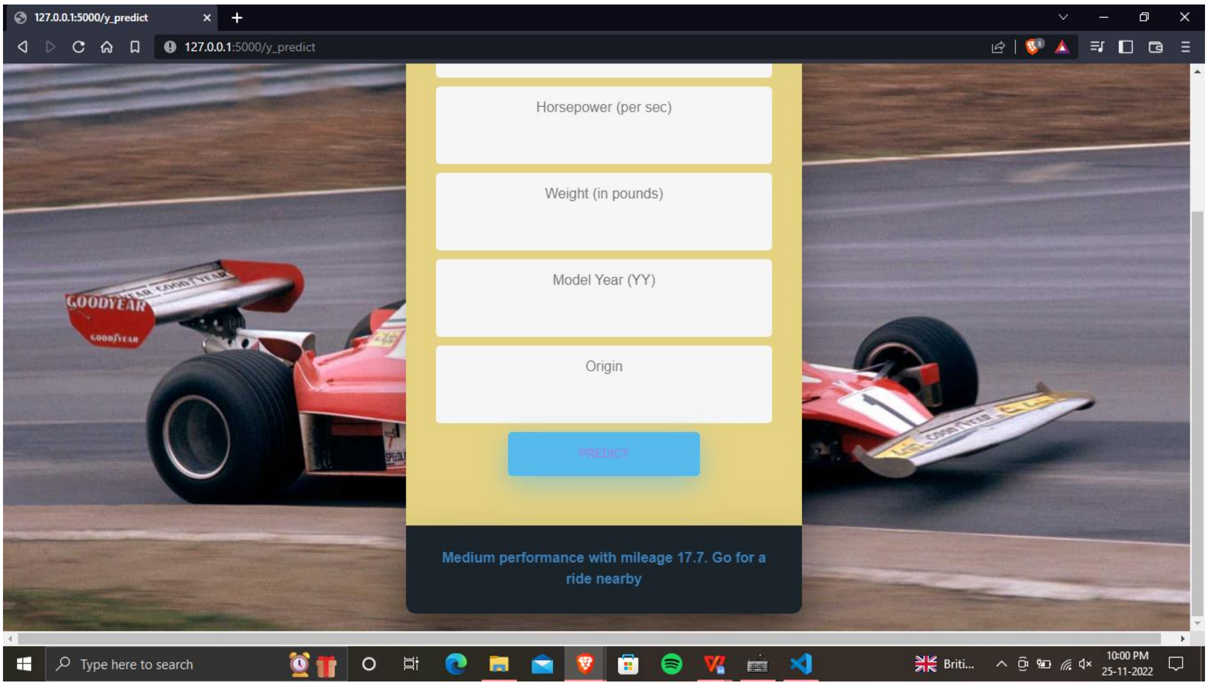
TEST CASE	No of Cylinders	Displacement	HP	Weight	Year	Origin	Predicted Value
1	8	307	130	3504	70	1	18.1
2	8	350	165	3693	70	1	15.2
3	4	130	95	2372	70	3	24.2
4	6	198	95	2833	70	1	22.3
5	4	104	95	2375	70	2	24.2

-


127.0.0.1:5000
CAR PERFORMANCE PREDICTION
No.of cylinders (count)
Displacement (in miles)
Horsepower (per sec)
Weight (in pounds)
Model Year (YY)
Origin
Type here to search
27°C
09:58 PM
25-11-2022


127.0.0.1:5000
CAR PERFORMANCE PREDICTION
8
350
165
3500
80
1
Type here to search
Briti...
10:00 PM
25-11-2022

8.2 User Acceptance Testing



Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Machine Learning-based Vehicle Performance Analyzer project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	9	0	0	9
Client Application	44	0	0	44
Security	2	0	0	2

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

1. Test Case Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	15	6	2	3	26
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	12	3	5	22	42
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	4	2	1	7
Totals	30	16	14	28	88

This report shows the number of test cases that have passed, failed, and untested

Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9 . Results :

9.1 Performance Metrics :
Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot

1.	Metrics	Regression Model:
		
		MAE - 1.7841,
		
		MSE - 6.5057,
		
		RMSE -2.5506 ,
		
		R2 score – 0.9058
			[63] #importing necessary libraries to find evaluation o
			from sklearn.metrics import mean_absolute_error as
			from sklearn.metrics import r2_score
			from sklearn.metrics import mean_squared_error
			import math
			
			# Mean Absolute Error
			MAE = mae(y_test, y_pred)
			print("MAE:",MAE)
			
			MAE: 1.7841771356783922
			
			[65] #mean squared error
			MSE=mean_squared_error(y_test,y_pred)
			print("MSE:",MSE)
			
			MSE: 6.505788848703318
			
			[66] #Root mean squared error
			RMSE=math.sqrt(MSE)
			print("RMSE:",RMSE)
			
			RMSE: 2.550644790774152
			
			[67] #checking the performance of the model using r2_sco
			r2=r2_score(y_test,y_pred)
			print("R2_score:",r2)
			
			R2_score: 0.9058760463516443

2.	Tune the Model	Hyperparameter Tuning –	`[41] from sklearn.ensemble import RandomForestRegres`
			`rf= RandomForestRegressor(n_estimators=10,rando`
			`model=rf.fit(x_train,y_train)`

TEST CASE	No of Cylinders	Displacement	HP	Weight	Year	Origin	Predicted Value
1	4	120	97	2506	72	3	23
2	4	98	80	2164	72	1	28
3	4	97	88	2100	72	3	27
4	8	350	175	4100	73	1	13
5	8	304	150	3672	73	1	14

10. ADVANTAGES & DISADVANTAGES :

ADVANTAGES:

- It helps users for predicting the vehicle performance.
- Here the chance of occurrence of error is less when compared with the existing system.
- It is fast, efficient and reliable.
- Avoids data redundancy and inconsistency.
- Very user-friendly.
- Easy accessibility of data

DISADVANTAGES:

- computer literacy and network access
- Low Computer Literacy
- Security Concerns
- Authenticity

Infrastructural Requirement.

10 . Conclusion :

The monitoring of car performance, especially gas consumption, has so far been approached only very superficially. A typical fuel gauge, when closely monitored, shows an extremely non- linear relationship between needle movement and fuel consumption. In accuracies occur especially in the range of critical low fuel values of 5-10% or more. In the past, due to This limitation, some luxury cars had an audible and flashing light alarm function to indicate a low fuel condition. These systems, which add to the existing fuel level, have no more accuracy than the fuel level monitor alone. In recent years, with the availability of computer techniques and reliable and less expensive computer equipment, a number of systems have been developed to provide somewhatmore accurate information about vehicle performance.

12. FUTURE SCOPE :

This merits exploratory methods based on actual failures to deduct likely failure modes. This thesis presents two methods for data mining the vehicle maintenance records and vehicle usage data to learn usage or wear patterns indicative of failures. This requires detailed maintenance records where the failure root cause can be deducted with accurate date or mileages of the repair.

Further, more wide-spread adoption of predictive maintenance calls for automatic and less human-resource demanding methods, e.g. unsupervised algorithms with lifelong learning. Such methods are easier to scale up and they can thus be ubiquitously applied since much of the modelling is automated and requires little or no human interaction.

Maintenance predictions can be enhanced by combining the deviations in onboard data with off-board data sources such as maintenance records and failure statistics. This is exclusive product knowledge, only accessible to the vehicle manufacturers, which gives them an advantage in predicting maintenance. Still, data mining has yet to become a core competence of vehicle manufacturers, which makes the road to industrialisation long.

The aim of this thesis is to investigate how on-board data streams and off-board data can be used to predict the vehicle maintenance. More specifically, how on-board data streams can be represented and compressed into a transmittable size and still be relevant for maintenance predictions. Further, the most interesting deviations must be found for a given repair which requires new ways of combining semantic maintenance records with deviations based on compressed on-board data.

This can be accessed anytime anywhere, since it is a web application provided only an internet connection.

13. APPENDIX :

https://github.com/IBM-EPBL/IBM-Project-14434-1659585666