Project Report

1. **INTRODUCTION**
    - Project Overview
    - Purpose
2. **LITERATURE SURVEY**
    - Existing problem
    - References
    - Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    - Empathy Map Canvas
    - Ideation & Brainstorming
    - Proposed Solution
    - Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    - Functional requirement
    - Non-Functional requirements
5. **PROJECT DESIGN**
    - Data Flow Diagrams
    - Solution & Technical Architecture
    - User Stories
6. **PROJECT PLANNING & SCHEDULING**
    - Sprint Planning & Estimation
    - Sprint Delivery Schedule
    - Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    - Feature 1
    - Feature 2
    - Database Schema (if Applicable)
8. **TESTING**
    - Test Cases
    - User Acceptance Testing
9. **RESULTS**
    - Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
    - Source Code
    - GitHub & Project Demo Link

INTRODUCTION

1.1 Overview

This project is used to analyze vehicles based on several fields of data which are collected by various methods, these data are well analyzed by the model created in python and the result derived from it. By utilizing the results generated one can improve their performance.

1.2 Purpose

The main purpose of the project is to depict the current performance of the vehicle accurately so that the user may upgrade accordingly to achieve better performance.

1. LITERATURE SURVEY

2.1 Existing problem

Some of the existing solutions for solving this problem are:

1. **Modelling and performance analysis of a vehicle with kinetic dynamic suspension system:**

   The proposed KDS system consists of two hydraulic circuits acting on two pairs of torsional rods and levers, which can be treated as novel anti-roll bars. Hence, these anti-roll bars do not work independently, but are coupled to merely respond to particular motion modes. The results show that the KDS system considerably improves the vehicle's anti-roll ability.

2. **Improved vehicle performance using combined suspension and braking forces:**

   The specific focus of this research is the integration of active suspension components with anti-lock braking (ABS) mechanisms. Simulations of the integrated controller and an ABS system demonstrate a significant increase in performance.

2.2 References

1. **Environment Setup:** https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

2. **Handling missing values:** https://youtu.be/xkRz6R0FlQ4

3. **Splitting dataset into trainset:** https://youtu.be/-KYiefj2wuw

4. **Integrating Flask:** https://www.analyticsvidhya.com/blog/2020/04/how-to-deploy-machine-learning-model-flask/

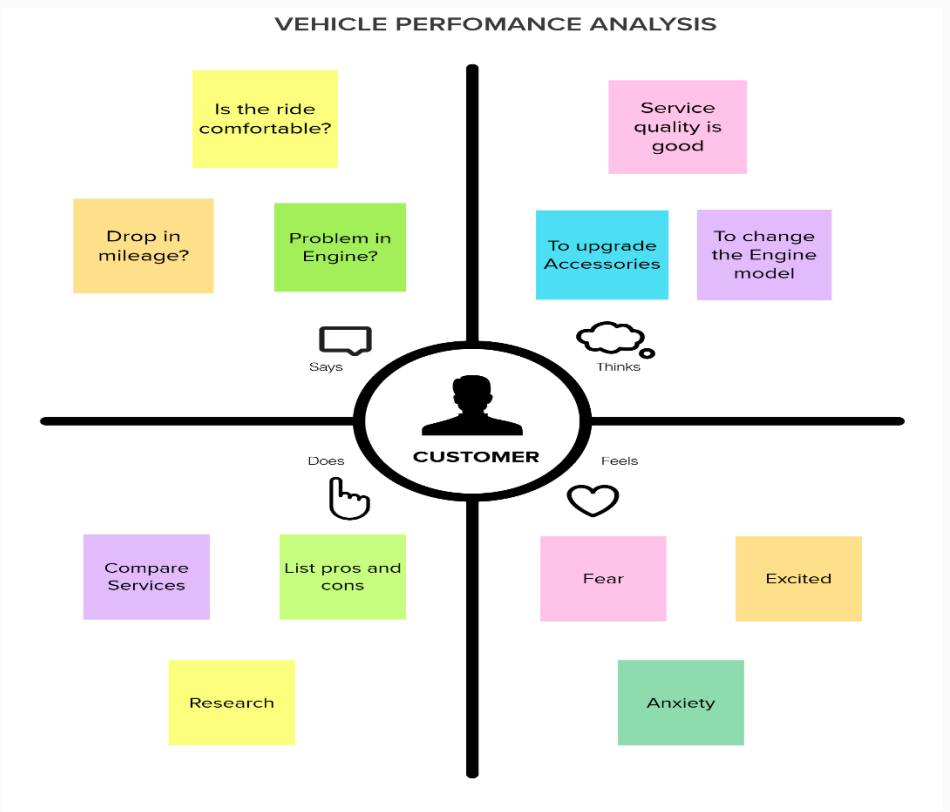
2.3 Problem Statement Definition

Predicting the performance level of cars is an important and interesting problem. The main goal is to predict the performance of the car to improve certain behaviours of the vehicle. This can significantly help to improve the system's fuel consumption and increase efficiency.

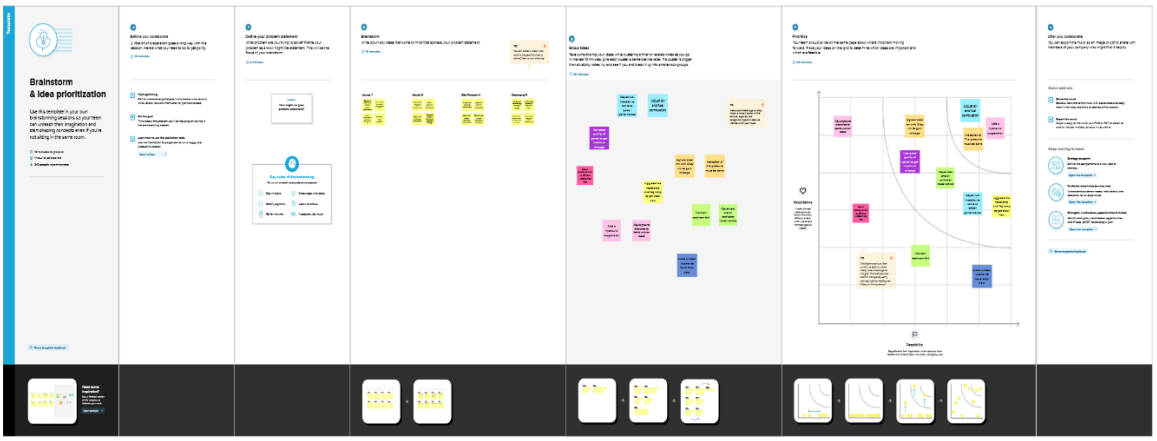
The performance analysis of the car is based on the engine type, no of engine cylinders, fuel type, horsepower, etc. These are the factors on which the health of the car can be predicted. It is an ongoing process of obtaining, researching, analysing, and recording health based on the above three factors. The performance objectives like mileage, dependability, flexibility and cost can be grouped together to play a vital role in the prediction engine and engine management system. This approach is a very important step towards understanding the vehicle's performance.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

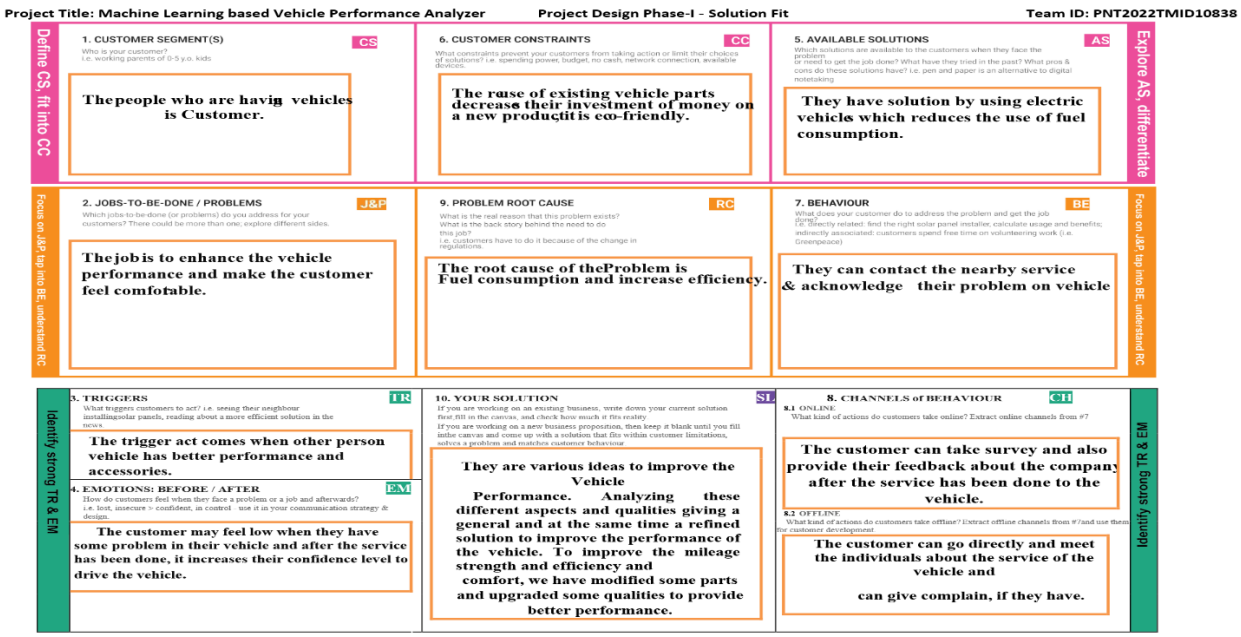


3.3 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	**Machine Learning based Vehicle Performance Analyzer** Predicting the performance level of cars is an important and interesting problem. The main goal isto predict the performance of the car to improve certain behaviours of the vehicle. This can significantly help to improve the system's fuel consumption and increase efficiency. The performance analysis of the car is based on the engine type, no of engine cylinders, fuel type, 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 health based on the above three factors. The performance objectives like mileage, dependability, flexibility and cost can be grouped together to play a vital role in the prediction engine and engine management system. This approach is a very important step towards understanding the vehicle's performance.
2.	Idea / Solution description	They are various ideas to improve the Vehicle Performance. Analyzing these different aspects and qualities giving a general and at the same time a refined solution to improve the performance of the vehicle. To improve the mileage strength and efficiency and comfort, we have modified some parts and upgraded some qualities to provide better performance.
3.	Novelty / Uniqueness	Generally many vehicle analyzer, focus on single quality or particular part of the vehicle. Whereas we have focused on all the domains ofthe vehicle and upgraded their quality and infrastructure to provide better performance for customer satisfaction.

4.	Social Impact / Customer Satisfaction	The petrol/diesel cost can become lower due to a better mileage performance and the existing vehicle parts can be reused which increases the reusability thus decreases the cost on new products and the physically abled people have better seat comfort because of accessories work. Better mileage and better engine maintenance provides complete combustion thus emitting less harmful gases.
5.	Business Model (Revenue Model)	Due to reusable of parts, we will spend only limited amount for modification/alteration. Therefore many customers could prefer the product as we are selling at low cost with a profit. Using this idea, we can make a stable business and get a profitable revenue.
6.	Scalability of the Solution	Our project has better scalability since our model analyses all information provides betterrefined solution. With less change to the vehicle we could achieve maximum performance.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Linked In.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Reset Password	Reset password through Gmail Reset password through Mobile number
FR-4	Feedback	The user can submit the feedback through a contact form in the website or through Gmail.

4.2 Non-Functional requirements

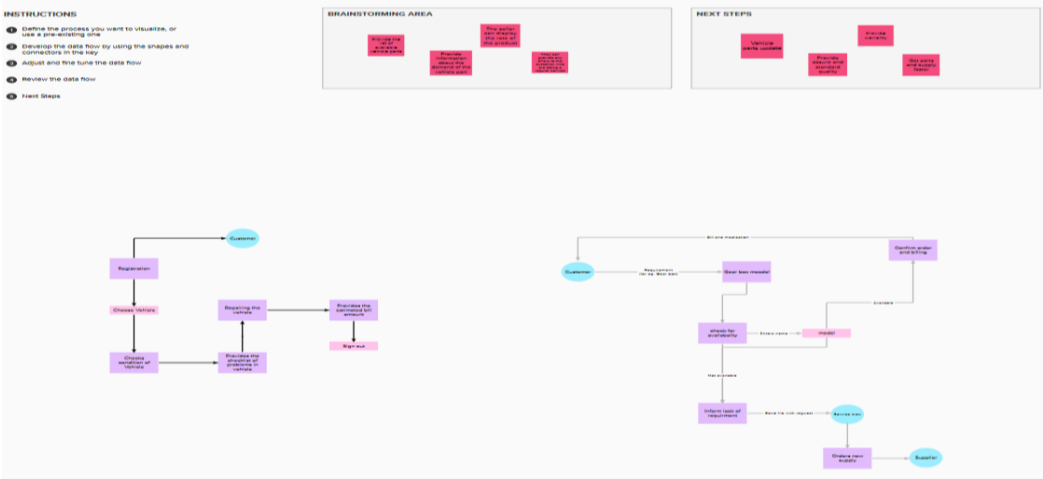
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The analyzer allows the user to improve performance based on the results provided. It is easy to use with just the data required.
NFR-2	**Security**	The security is improved by using vehicle alarm, wheel lock, vehicle lock and also GPS tracker.
NFR-3	**Reliability**	The reliability rating is good due to best performance, less frequency of problem occurrence and cost for repairing is low.
NFR-4	**Performance**	The vehicle is upgraded in their quality and infrastructure to provide better performance likegood mileage, smooth travel due to good suspension and better engine performance.
NFR-5	**Availability**	The data required is collected by research persons and this data can be used to provide better results.

NFR-6	**Scalability**	Our project has better scalability since our model analyses all information provides better refined solution. With less change to the vehicle, we couldachieve maximum performance.

5. PROJECT DESIGN

5.2 5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture:

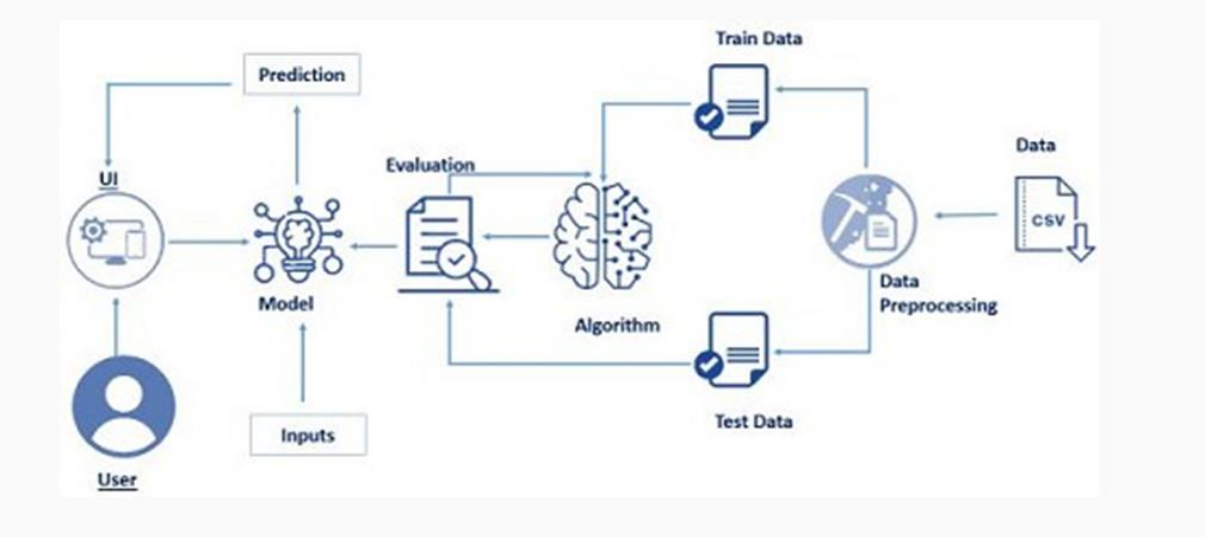


Table-5.2.1: Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	With the help of web UI, user has better experience And can access the website user-friendly.	HTML, CSS, JavaScript, React JS.
2.	Application Logic-1	Customer can login with username and password.	Java / Python
3.	Application Logic-2	Customer can give their vehicle faults.	IBM Watson STT service
4.	Application Logic-3	Customer can check their vehicle performance and can check the vehicle after the service.	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration:	Local, Cloud Foundry, Kubernetes, etc.

Table-5.2.2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

5.3 User Stories:

User Type	Functional Requirements	User Story Number	User Story/Task	Acceptance Criteria	Priority	Release
Customer	Access the Webpage	USN -1	Anyone can access the webpage to check the specifications of the vehicle	I can access my webpage online at any time	High	Sprint-1
Customer	Performance of the Vehicle	USN - 2	As per the usageof the user, the performance of the vehicle should be predictable.	Prediction can be done in an easy way.	High	Sprint-2

Customer	Accuracy to check the performance and health of the car	USN -3	By using our prediction, it helps to check the health of the car.	The efficiency of the car can be predicted.	High	Sprint-3

6. PROJECT PLANNING & SCHEDULING

6.2 6.1 Sprint Planning

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Required Data	USN-1	Model Building	2	High	Arvind
Sprint-2		USN-2	Application Building	1	High	Arvind, Harish, Dharmaraj
Sprint-3		USN-3	Train the model on IBM	2	Low	Arvind, Harish, Dharmaraj, Gita Prakash
Sprint-4		USN-4	Integrate flask with scoring end-point	2	Medium	Arvind
Sprint-2			In Application Development: Enter the number of cylinders	1	High	Harish, Arvind
Sprint-2			Enter the displacement Enter the Horse Power	2	Medium	Harish, Arvind
Sprint-2			Enter the Weight Enter acceleration	1	High	Dharmaraj, Gita Prakash
Sprint-2			Enter model year and origin Check the performance		Medium	Harish, Arvind

6.2 **Estimation:**

Pre-Requisites	M-01	The following software concepts and packages, including Machine learning, Python, KNN, Python Flask, IBM Cloudland DB, and Watson Studio, should have been familiar to us by the time we finished this project.	Yes
Data Collection	M-02	To create a project structure, create a Dataset.	Yes
Data Preprocessing	M-03	The dataset collection is separated into a various collection, first reading the dataset, handling the missing values, label encoding and one hot coding, splitting the dataset into dependent and independent variable, and into trainset and test set and normalizing and finally importing libraries.	Yes
Model Building	M-04	Build the model with the random forest regressor, predict the values and model the evaluation	Yes
Application Building	M-05	First, build an Index, HTML file, python code and python code-II, Run the app and finally output.	Yes
Train the model on IBM	M-06	Register on cloud IBM, train the model on IBM and integrate with the flask with scoring end point.	Yes

Ideation Phase	M-07	Prepare empathy map, take literature survey and Ideation.	Yes
Project Design Phase-I	M-08	Proposed Solution, Problem solution fit, Solution architecture.	Yes
Project Design Phase-II	M-09	Preparation of the technological stack architecture, functional requirements, data flow diagrams, and customerjourney mapping.	Yes
Project PlanningPhase	M-10	Prepare Milestone & Activity List and Sprint Delivery Plan.	Yes
Project developme nt phase	M-11	Develop Sprint 1, Sprint 2, Sprint 3, Sprint 4.	Yes

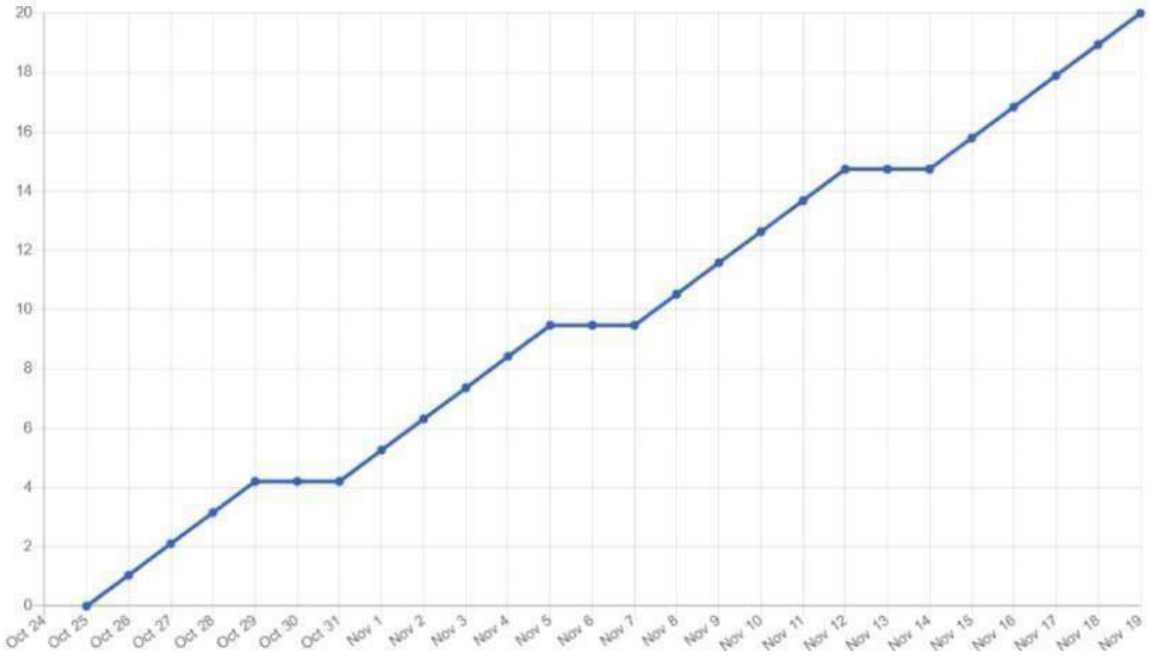
6.2 Sprint Delivery Schedule:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 ov 2022

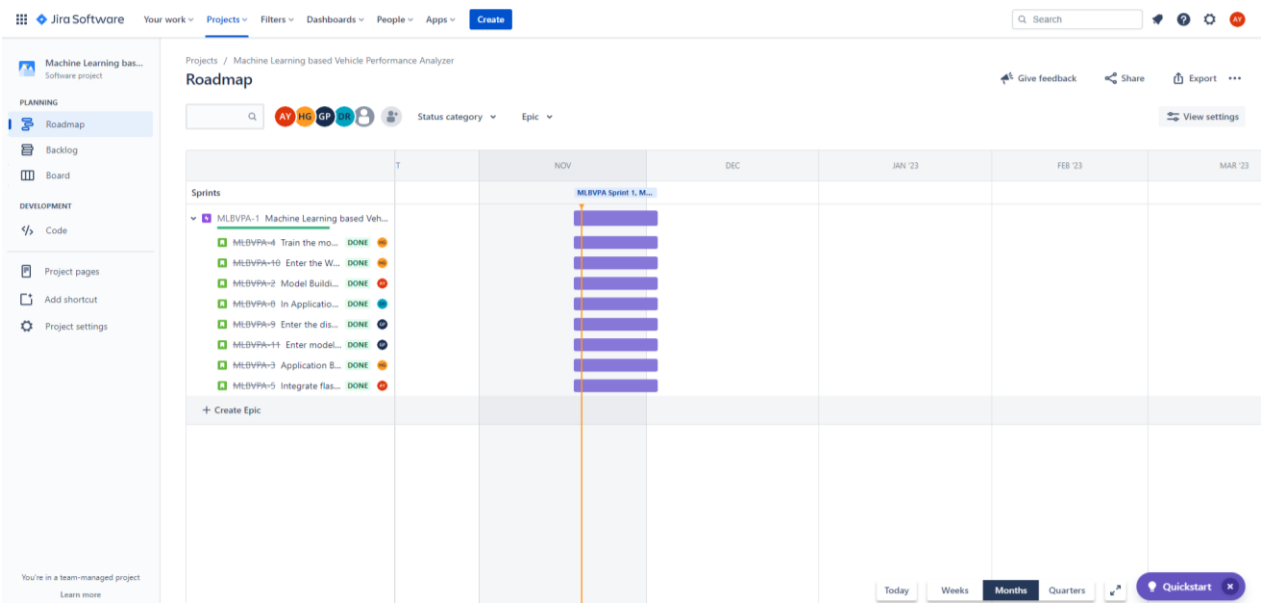
6.3 Reports from JIRA:

Burndown Chart: A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down chartscan be applied to any project containing measurable progress over time.

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$



STATUS:

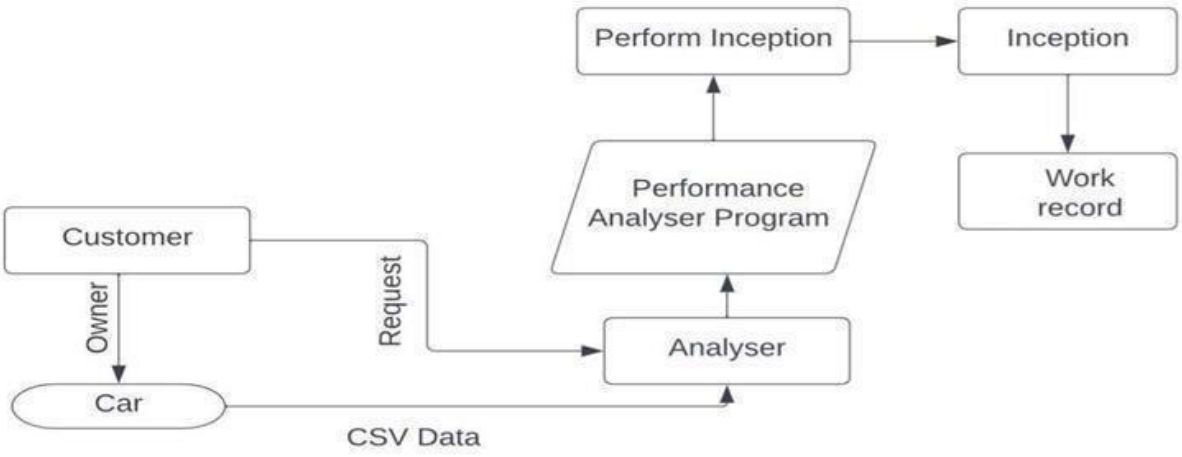


7. CODING & SOLUTIONING:

7.1 Features:

FR No.	Feature	Description
FR-1	Enter the input	Get input through the form
FR-2	User Essential	Predict the performance of the vehicle
FR-3	Data preprocessing	Sample dataset for training purpose
FR-4	User input Evaluation	Evaluating the given user values
FR-5	Prediction	Fuel consumption and efficiency of the vehicle

DFD:



8. Testing:

8.1 TEST CASES:

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
HomePage_TC_OO 1	Functional	Home Page	Verify if the user is able to enter the data into the text field in the webpage and click the button		1. Enter the URL 2. Enter the values	[8,307.130,3504.70,1]	Page refresh	Working as expected	Pass				Sandeep
HomePage_TC_OO 2	Functional	Home page	Verify is the user is able to view the output after the submit button has been clicked		1. Click the submit button		Low performance with mileage 17.1	Working as expected	Pass				Sandeep

8.2 User Acceptance Testing:

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	1	0	0	2
Duplicate	1	0	0	0	1
External	1	0	0	0	1
Fixed	1	1	1	1	4
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	4	2	1	1	13

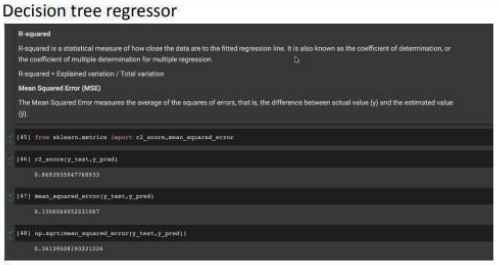
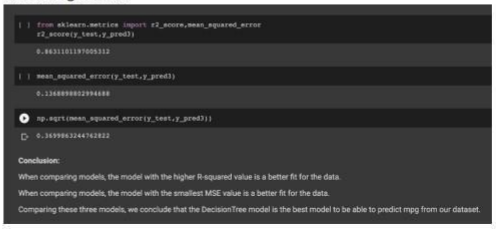
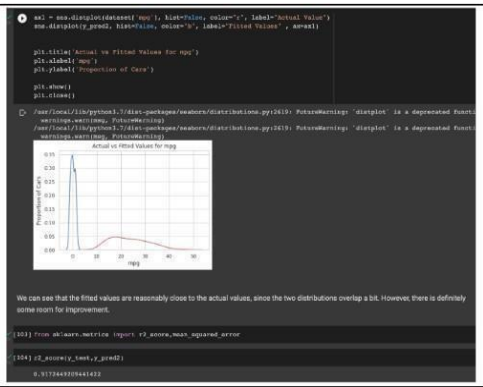
3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	4	0	0	4
Client Application	4	0	0	4
Security	1	0	0	1
Outsource Shipping	0	0	0	0
Exception Reporting	1	0	0	1
Final Report Output	4	0	0	4
Version Control	1	0	0	1

9. RESULTS:

9.1 PERFORMANCE METRICS:

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** MAE - , MSE - , RMSE - , R2 score - **Classification Model:** Confusion Matrix - , Accuray Score- & Classification Report -	Decision tree regressor 
			Random forest regressor  Linear regression 
	Accuracy	Training Accuracy - 0.91724492094	

10. PROS AND CONS:

10.1 PROS:

- Using the Random Forest Algorithm in the model helps to perform both classification aswell as regression tasks.
- A random forest produces good predictions that can be easily understood
- It can handle large datasets easily Random Forest Algorithm provides a higher-level accuracy in predicting outcomes.

10.2 CONS:

- The main limitation of using random forest algorithm in the model is that a large numberof trees can make the algorithm too slow and ineffective for real-time predictions.
- The random forest algorithm is quite slow to create predictions once it is trained.

11. CONCLUSION:

The ability to estimate a car's performance level presents a big and fascinating challenge. Forecasting vehicle performance in order to improve particular vehicle behavior was our main goal. performance evaluation of the car considering its horsepower, cylinder count, fuel type, and engine type, among other things. Based on the factors, like horsepower, cylinder count, fueltype, and engine type, the health of the car is forecasted. We analyzed the components using a number of well-known machine learning approaches, like linear regression, decision trees, and random forests, in order to optimize the performance efficiency of the vehicle. The power, longevity, and range of automobile traction batteries are now the "hot topics" in automotive engineering. In this case, we additionally consider mileage performance. To answer this problem, we have built the models using a variety of methods and neural networks. We've then

compared which algorithm is most accurate in forecasting car performance (Mileage). A front- end webpage was designed to help give the user an attractive front while they input the values required by the developed machine learning model. The IBM cloud platform was used to develop the model.

12. FUTURE WORKS:

The dataset used for this model is an old vehicle dataset, thus the model's accuracy would drop when the details of vehicles released in recent times are given as input. Thus, in the future we propose to use the latest dataset set containing vehicle information to help train the model. We also plan to use other classification algorithms such as SVM and Decision Tress instead of Random Forest and measure if any accuracy gain occurs. Finally, we propose to scale the machine learning model to also analyze the performance of a larger range of vehicles.

13. APPENDIX:

13.1 Car Performance Prediction.ipbyn:

```
import numpy as np
import pandas as pd
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='wdPOG7CvYRZxYt4sjm8d_Qv7Fzslp7NDy9yWfHWE
    xaSG',
    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-
    eqbab3sfwyugyu'
object_key = 'car performance (1).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 )

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

```
    x=datas.iloc[:,1:8]
        x
    y=datas.iloc[:,0]
        y
from sklearn.model_selection import train_test_split
        x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.2,random_state=0
from sklearn.preprocessing import StandardScaler
        sd=StandardScaler()
        x_train=sd.fit_transform(x_train)
        x_test=sd.fit_transform(x_test)
from sklearn.ensemble import RandomForestRegressor
     d=RandomForestRegressor(n_estimators=30,random_state=0)
     d.fit(x_train,y_train)
!pip install ibm_watson_machine_learning

from ibm_watson_machine_learning import APIClient
     wml_credentials={
                "url":"https://us-south.ml.cloud.ibm.com",

         "apikey":"zDg62IPh9bpRQ06F0TDmtiqqDoQfoiv4z4tcu2RUY9fF"
                }
client=APIClient(wml_credentials)
def guid_from_space_name(client,space_name):
         space=client.spaces.get_details()
         #print(space)
   return(next(item for item in space['resources'] if
item['entity']["name"]==space_name)['metadata']['id'])
space_uid=guid_from_space_name(client,'models')
print("Space UID = "+ space_uid)
client.set.default_space(space_uid)
client.software_specifications.list()
     software_spec_uid =
     client.software_specifications.get_uid_by_name("runtime-22.1-py3.9")
software_spec_uid

     model_details = client.repository.store_model(model=d,meta_props={
        client.repository.ModelMetaNames.NAME : "Model Building",
```

```
        client.repository.ModelMetaNames.TYPE : "scikit-learn_1.0",

         client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software
         _spec_uid }
                                        )

model_id = client.repository.get_model_uid(model_details)
model_id
     #Prediction
     y_pred=d.predict(x_test)
y_pred
from sklearn.metrics import r2_score
     accuracy=r2_score(y_pred,y_test)
accuracy
    import pickle
        pickle.dump(d,open('regression.pkl','wb'))
x2=[[4,7,58,89,1000,568,70]]
     y=d.predict(x2)
y
```

13.2 Sourcing end point.py:

```
import requests

# NOTE: you must manually set API_KEY below using information
    retrieved from your IBM Cloud account.
API_KEY = " zDg62IPh9bpRQ06F0TDmtiqqDoQfoiv4z4tcu2RUY9fF"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
    data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
    mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in
    the next line
payload_scoring = {"input_data": [{"fields":
    [["Cylinders","Displacement","Horsepower","Weight","Acceleration","Mo
```

```
delYear","Origin"]], "values": [4,7,58,89,1000,568,70]}]}

response_scoring = requests.post(https://us-
    south.ml.cloud.ibm.com/ml/v4/deployments/3c78e47a-ee1f-421d-
    b0e8-aac0a2c49036/predictions?version=2022-11-19,
    json=payload_scoring,
 headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
    print(response_scoring.json())
```

13.3 Index.html:

```
<html>
<head>
<meta charset="ISO-8859-1">

<style>
    body{
        background-size: cover;
    }
</style>
</head>
<body background="cars.jpg">

<div align="center">
        <h1>VEHICLE PERFORMANCE PREDICTION</h1>

        <br><br><br>
        <form action="output.html" method="get">
            <table>
                <tr>
                    <td><b>No of Cylinders:</b></td>
                    <td><input type="text" name="Cylinders" /></td>
                </tr>
                <tr>
                    <td><b>Enter Displacement:</b></td>
                    <td><input type="text" name="Displacement" /></td>
                </tr>
                <tr>
                    <td><b>Enter Horsepower:</b></td>
                    <td><input type="text" name="HorsePower" /></td>
                </tr>
                <tr>
                    <td><b>Weight:</b></td>
                    <td><input type="text" name="Weight" /></td>
                </tr>
                <tr>
```

```
                <td><b>Model Year:</b></td>
                <td><input type="text" name="Model" /></td>
            </tr>
            <tr>
                <td><b>Enter Origin:</b></td>
                <td><input type="text" name="Origin" /></td>
            </tr>

            </table><br>

        <input type="submit" value="PREDICT"/>


    </form>
</div>
```

13.4 app.py:

```
from flask import Flask, request, Response, send_from_directory
import requests
import json
from flask_cors import CORS
import ibm_db

# NOTE: you must manually set API_KEY below using information
    retrieved from your IBM Cloud account.
API_KEY = " zDg62IPh9bpRQ06F0TDmtiqqDoQfoiv4z4tcu2RUY9fF"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
    data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
    mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in
    the next line
#payload_scoring = {"input_data": [{"fields":
    [["Cylinders","Displacement","Horsepower","Weight","Acceleration","Mo
    delYear","Origin"]], "values": [4,7,58,89,1000,568,70]}]}
```

```
#API_KEY_NEW = 'https://us-
    south.ml.cloud.ibm.com/ml/v4/deployments/6ed70a51-2d98-4119-
    a5bf-eda733928a88/predictions?version=2022-11-17'

app=Flask(_name_)

@app.route('/health-check', methods=['GET'])
def health_check_for_user():
    return Response("Running")


@app.route('/get-key', methods=['GET'])
def get_key():
    API_KEY = "zDg62IPh9bpRQ06F0TDmtiqqDoQfoiv4z4tcu2RUY9fF"
    token_response =
     requests.post('https://iam.cloud.ibm.com/identity/token',
     data={"apikey":
      API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
    mltoken = token_response.json()["access_token"]
    return Response(mltoken)


@app.route('/get-users', methods=['GET'])
def get_users():
    return get_user()


@app.route('/get-performance', methods=['POST'])
def get_performance_for_user():
    print(request.args)
    query = request.get_json()
    print(query)
    inp = query.get('inp')
    payload_scoring = {"input_data": [{"field": [["cylinders", "displacement",
     "horsepower", "weight", "acceleration", "model year", "origin"]],
     "values": [
        inp]}]}
```

```
    API_KEY = "zDg62IPh9bpRQ06F0TDmtiqqDoQfoiv4z4tcu2RUY9fF"
    token_response =
     requests.post('https://iam.cloud.ibm.com/identity/token',
     data={"apikey":
                                                          API_KEY,
     "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
    mltoken = token_response.json()["access_token"]

    header = {'Content-Type': 'application/json',
              'Authorization': 'Bearer ' + mltoken}
    url = API_KEY_NEW
    api_response = requests.post(url=url, json=payload_scoring,
     headers=header)
    return Response(api_response)

if _name_ == '_main_':
        app.run(debug=True)
```

GitHub & Project Demo Link:

https://youtu.be/3xhkbnUNGmg