Project Report

# INTRODUCTION

* 1. Project Overview
  2. Purpose

# LITERATURE SURVEY

* 1. Existing problem
  2. References
  3. Problem Statement Definition

# IDEATION & PROPOSED SOLUTION

* 1. Empathy Map Canvas
  2. Ideation & Brainstorming
  3. Proposed Solution
  4. Problem Solution fit

# REQUIREMENT ANALYSIS

* 1. Functional requirement
  2. Non-Functional requirements

# PROJECT DESIGN

* 1. Data Flow Diagrams
  2. Solution & Technical Architecture
  3. User Stories

# PROJECT PLANNING & SCHEDULING

* 1. Sprint Planning & Estimation
  2. Sprint Delivery Schedule
  3. Reports from JIRA

# CODING & SOLUTIONING (Explain the features added in the project along with code)

* 1. Feature 1
  2. Feature 2
  3. Database Schema (if Applicable)

# TESTING

* 1. Test Cases
  2. User Acceptance Testing

# RESULTS

* 1. Performance Metrics

# ADVANTAGES & DISADVANTAGES

1. **CONCLUSION**

# FUTURE SCOPE

1. **APPENDIX**

Source Code

GitHub & Project Demo Link

# INTRODUCTION

## 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.

## 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.

1. **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.

* 1. **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/>

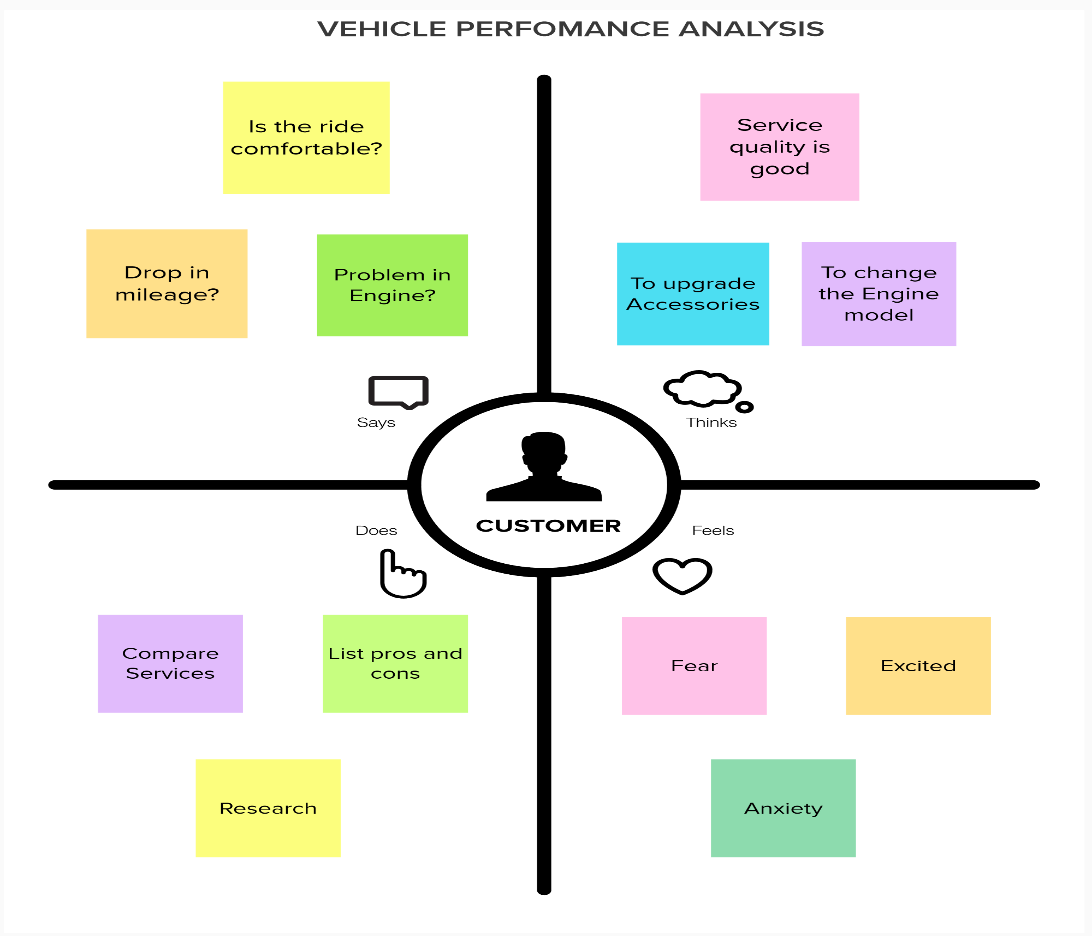
**2.3 Problem Statement Definition**

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.

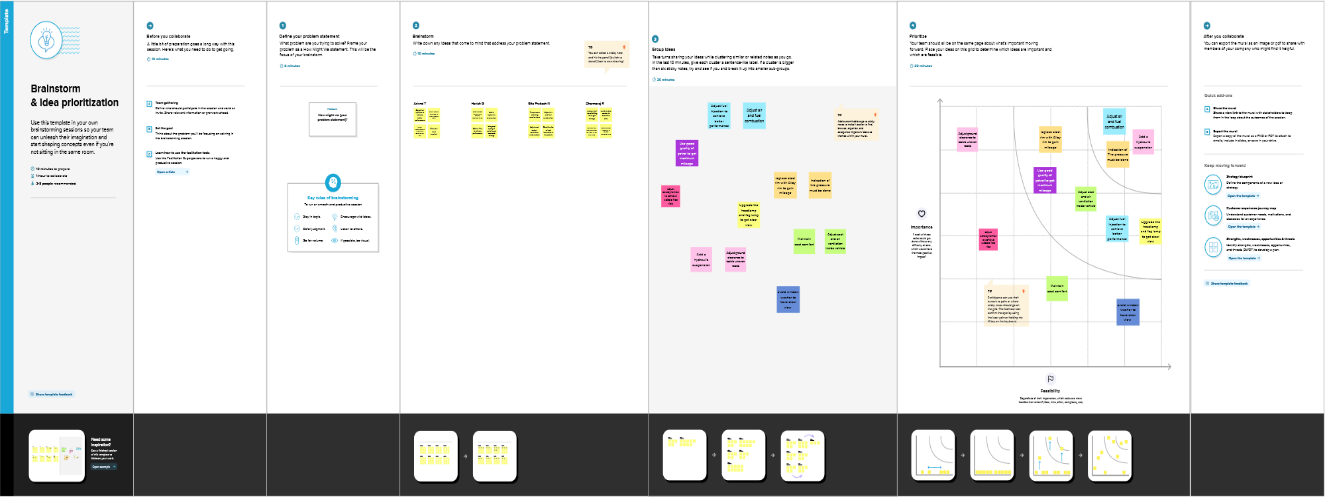
The performance analysis of the car is based on the engine type, no of engine cylinders, fuel type, horsepower, etc. These are the factors on which the health of the car can be predicted. It is an ongoing process of obtaining, researching, analysing, and recording health based on the above three factors. 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.

# IDEATION & PROPOSED SOLUTION

* 1. **3.1 Empathy Map Canvas**



**3.2 Ideation & Brainstorming**

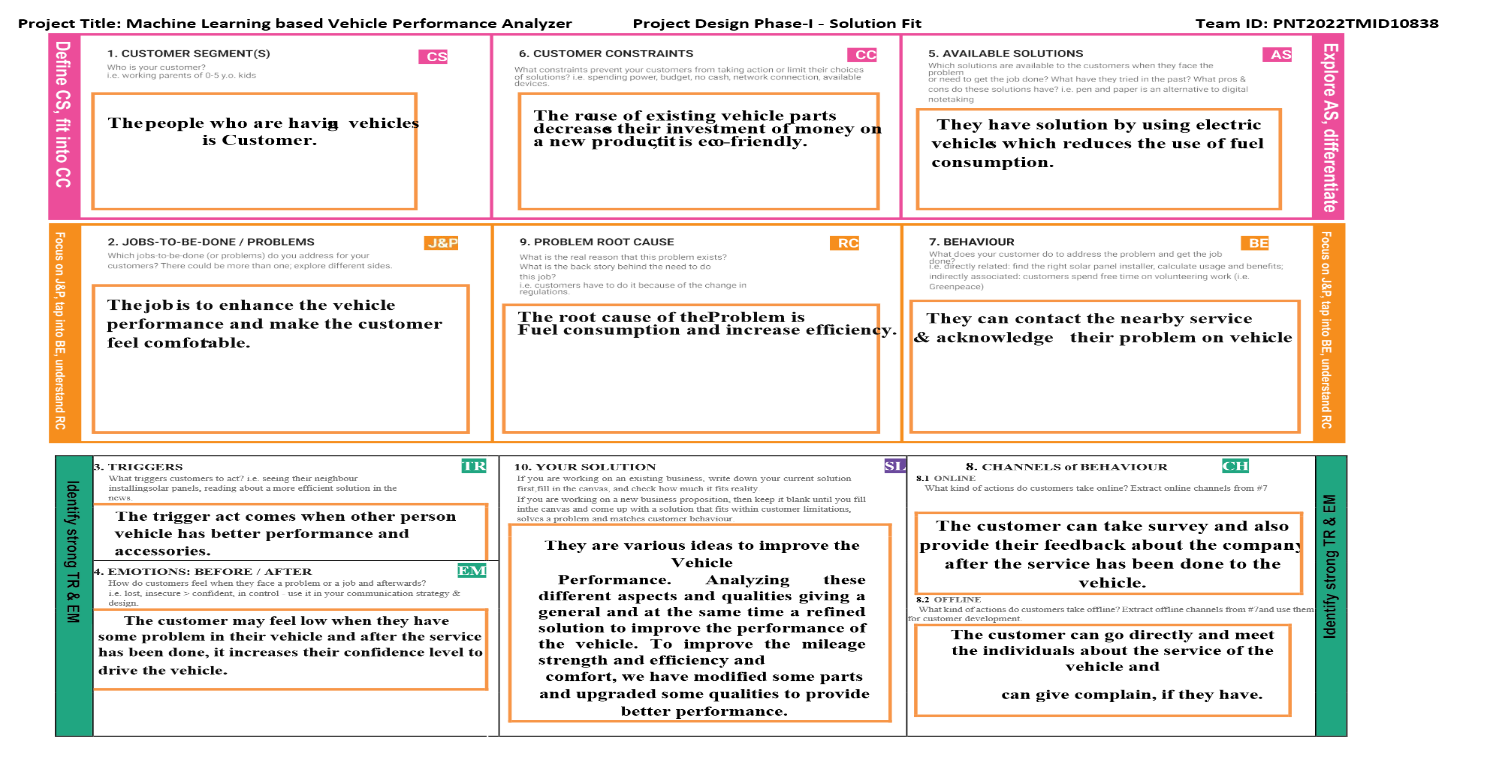


**3.3 Proposed Solution**

|  |  |  |
| --- | --- | --- |
| **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 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 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 of the 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 better refined solution. With less change to the vehicle we could achieve maximum  performance. |

**3.4 Problem Solution fit**



# REQUIREMENT ANALYSIS

* 1. **Functional 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. |

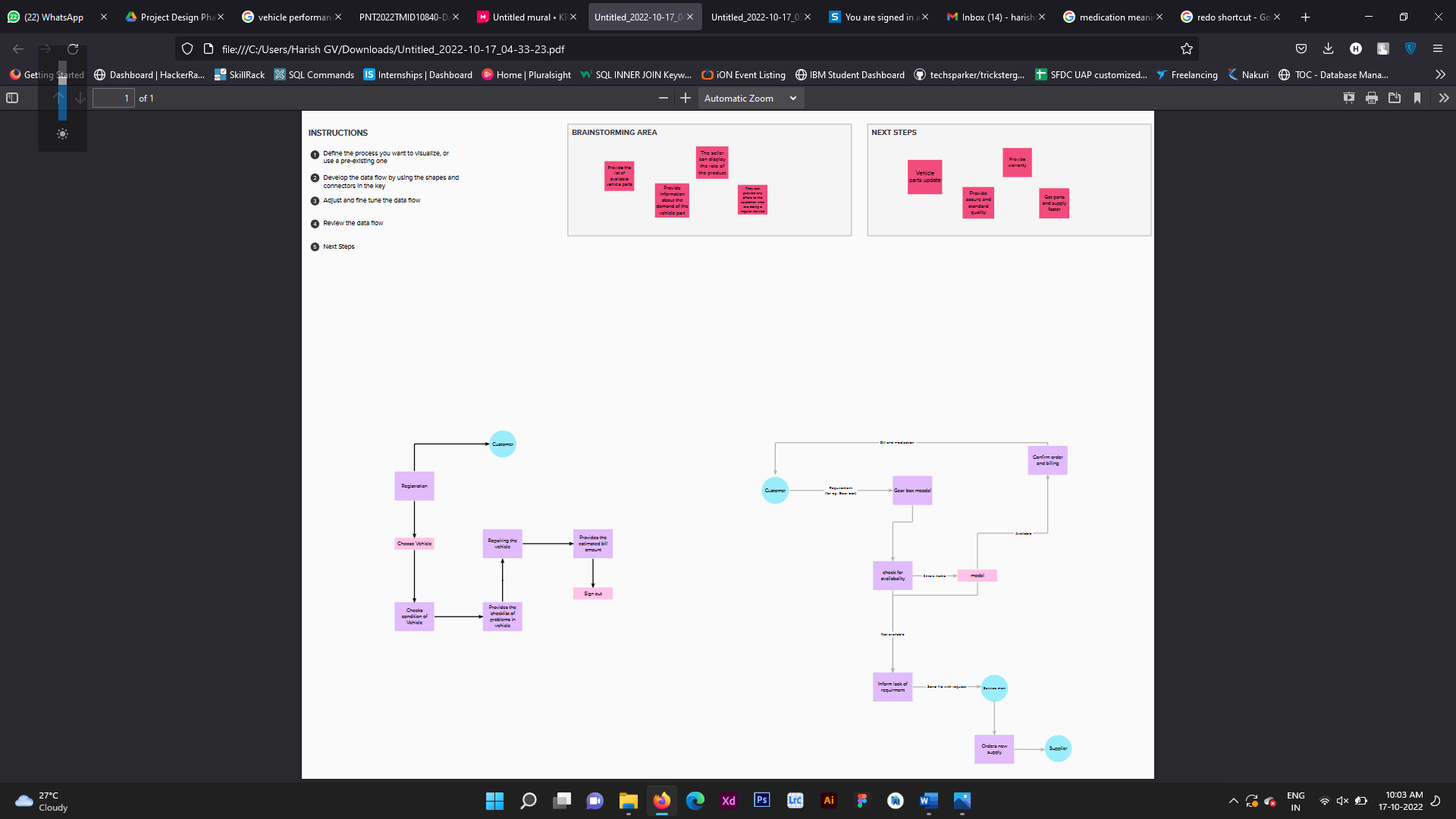
**4.2 Non-Functional requirements**

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

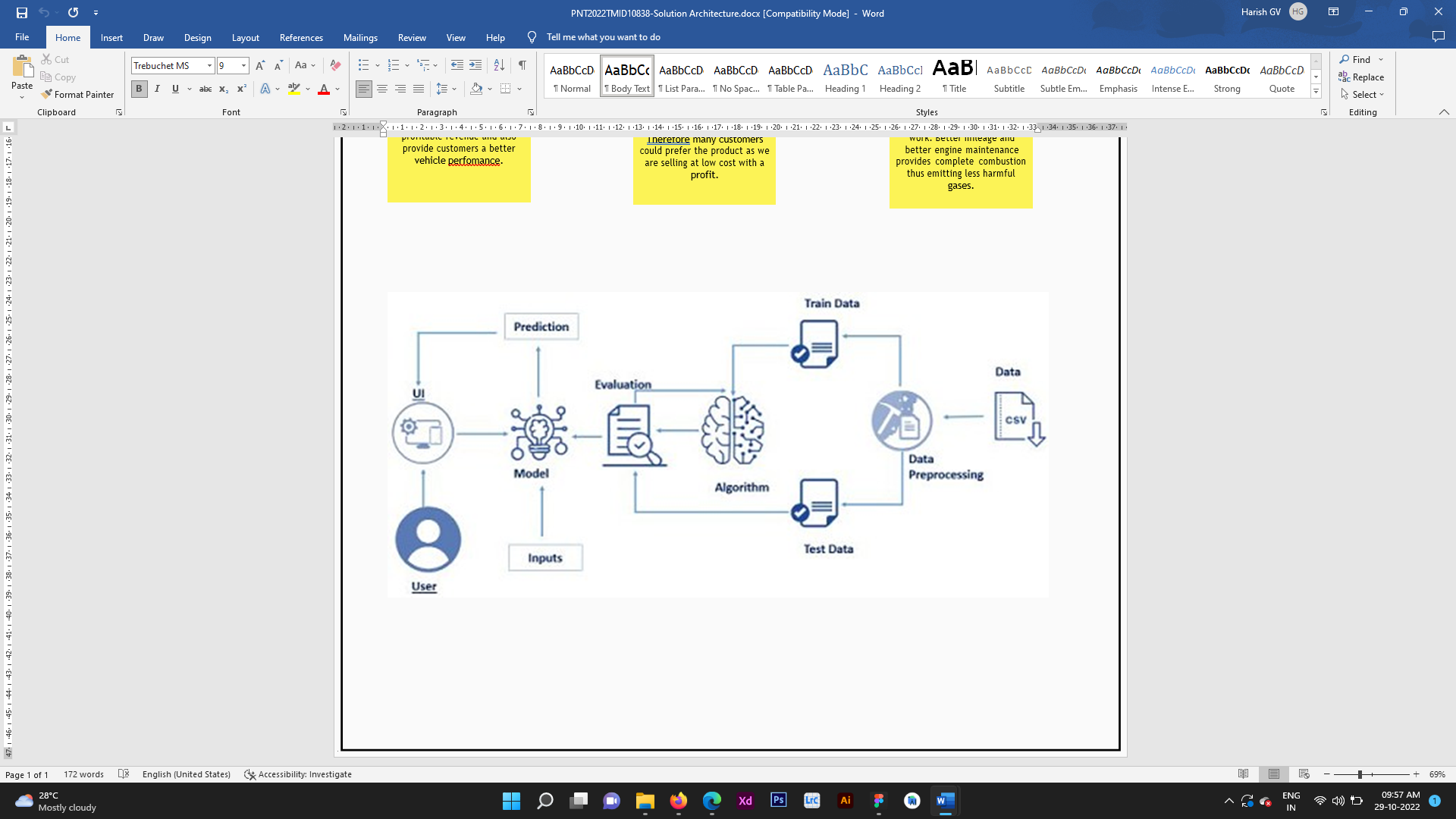
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | The analyzer allows the user to improve performance based on the results provided. It is  easy to use with just the data required. |
| NFR-2 | **Security** | The security is improved 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 like good 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 could achieve maximum performance. |

# PROJECT DESIGN

* 1. **5.1 Data Flow Diagrams**



**5.2 Solution & Technical Architecture:**



**Table-5.2.1: Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Component** | **Description** | **Technology** |
|  | User Interface | With the help of web UI, user has better experience  And can access the website user-friendly. | HTML, CSS, JavaScript, React JS. |
|  | Application Logic-1 | Customer can login with username and password. | Java / Python |
|  | Application Logic-2 | Customer can give their vehicle faults. | IBM Watson STT service |
|  | Application Logic-3 | Customer can check their vehicle performance and can check the vehicle after the service. | IBM Watson Assistant |
|  | Database | Data Type, Configurations etc. | MySQL, NoSQL, etc. |
|  | Cloud Database | Database Service on Cloud | IBM DB2, IBM Cloudant etc. |
|  | File Storage | File storage requirements | IBM Block Storage or Other Storage Service or Local Filesystem |
|  | External API-1 | Purpose of External API used in the application | IBM Weather API, etc. |
|  | External API-2 | Purpose of External API used in the application | Aadhar API, etc. |
|  | Machine Learning Model | Purpose of Machine Learning Model | Object Recognition Model, etc. |
|  | 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** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | List the open-source frameworks used | Technology of Opensource framework |
|  | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | e.g. SHA-256, Encryptions, IAM Controls, OWASP etc. |
|  | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | Technology used |
|  | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Technology used |
|  | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc. | Technology used |

* 1. **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 usage of 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 |

# PROJECT PLANNING & SCHEDULING

* 1. **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 |

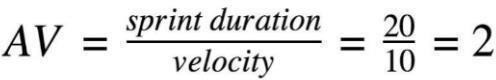
|  |  |  |  |
| --- | --- | --- | --- |
| Ideation Phase | M-07 | Prepare empathy map, take literature survey and Ideation. | Yes |
| Project Design Phase-I | M-08 | Proposed Solution, Problem solution fit, Solution architecture. | Yes |
| Project Design Phase-II | M-09 | Preparation of the technological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

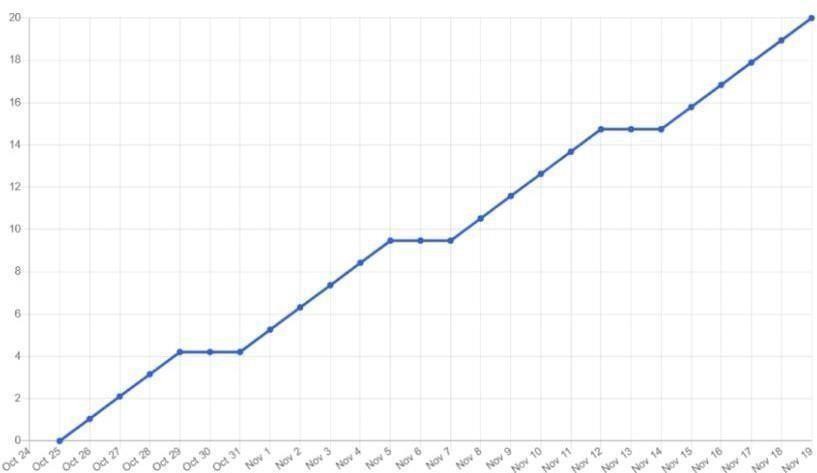
**6.2 Sprint Delivery Schedule:**

| **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 | 1. ov 2022 |

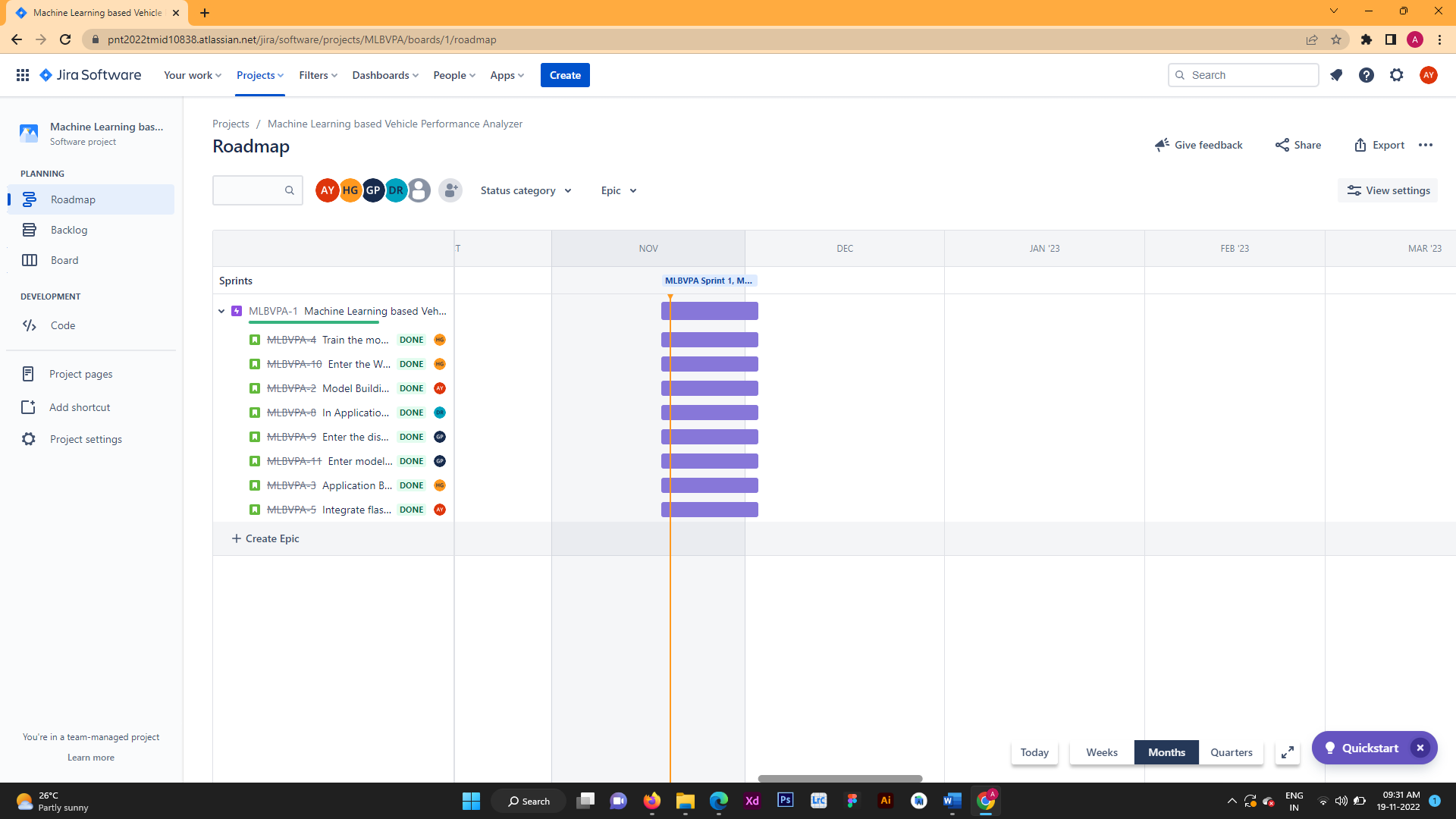
**6.3 Reports from JIRA:**

**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 charts can be applied to any project containing measurable progress over time.



****

**STATUS:**



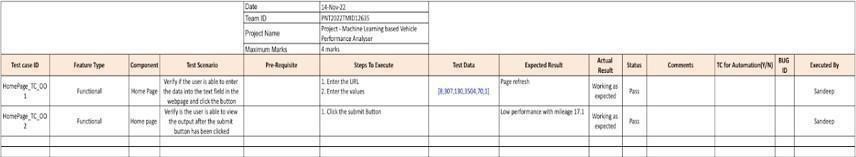
# CODING & SOLUTIONING:

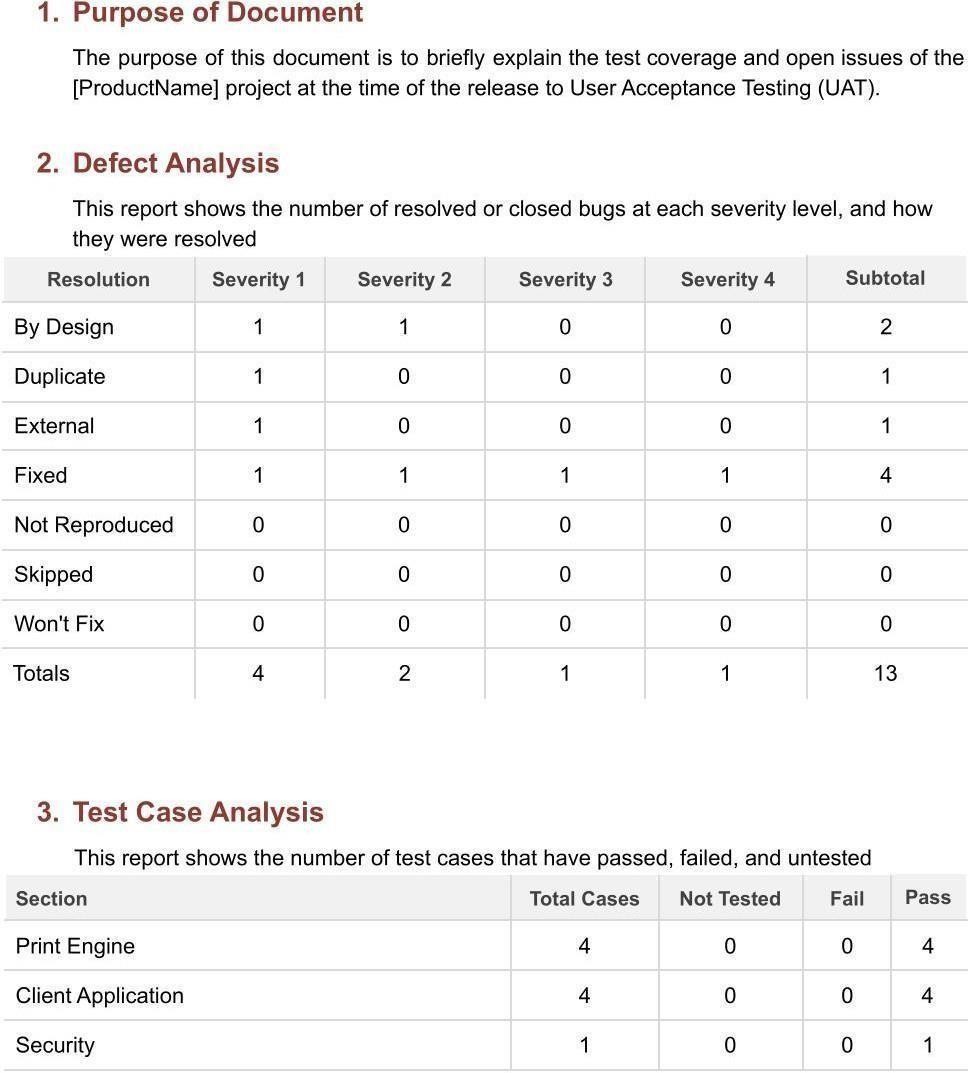
# 7.1 Features:

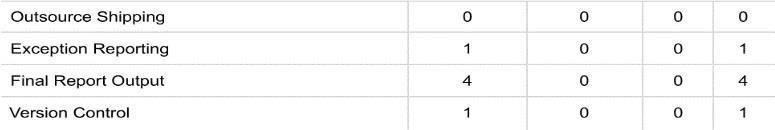
|  |  |  |
| --- | --- | --- |
| **FR**  **No.** | **Feature** | **Description** |
| FR-1 | Enter the input | Get input through the form |
| FR-2 | User Essential | Predict the performance of the vehicle |
| FR-3 | Data preprocessing | Sample dataset for training purpose |
| FR-4 | User input Evaluation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

# DFD:

1. **Testing:**
   1. **TEST CASES:**

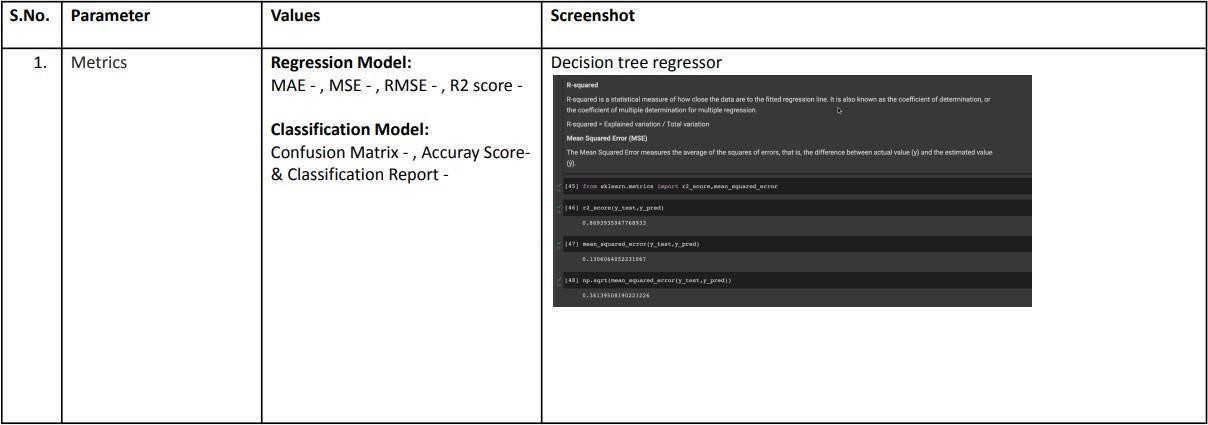


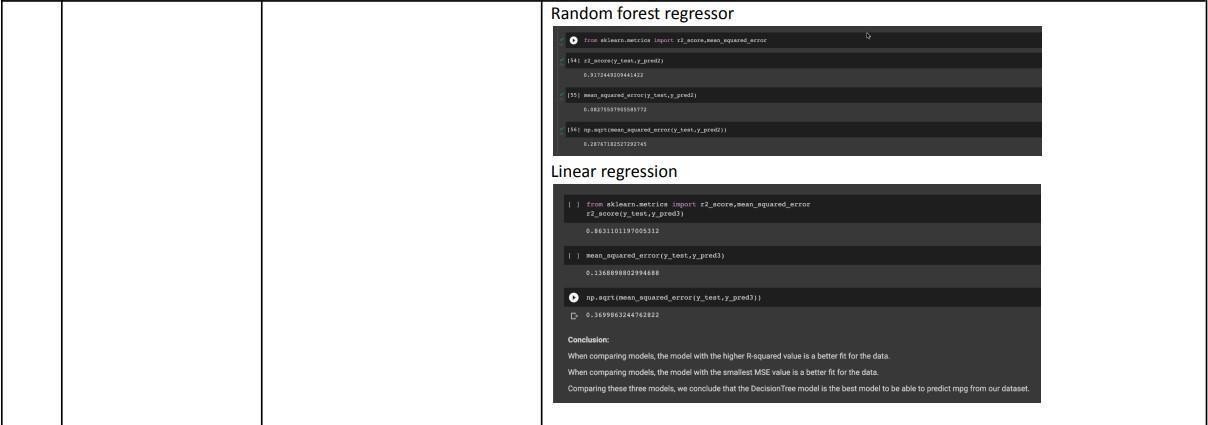
* 1. **User Acceptance Testing:**

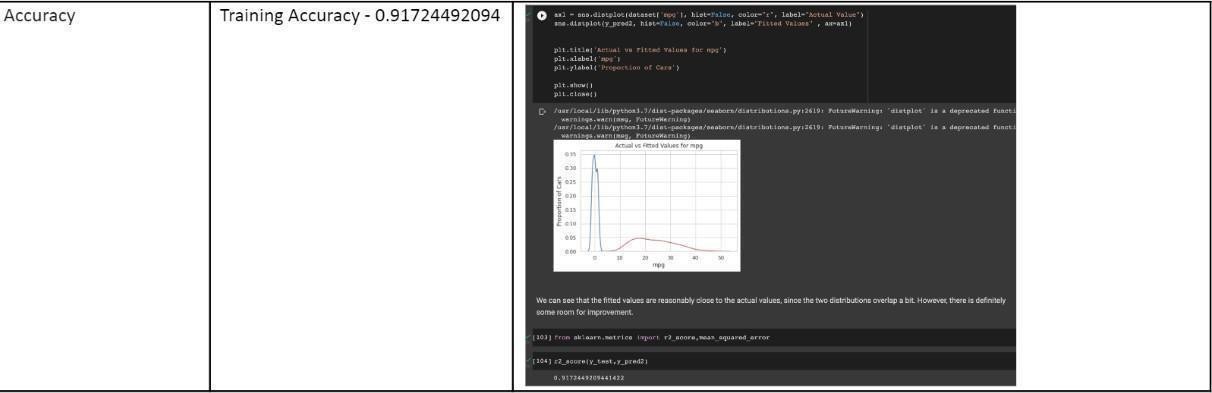


1. **RESULTS:**

**9.1 PERFORMANCE METRICS:**







1. **PROS AND CONS:**

**10.1 PROS:**

* + - Using the Random Forest Algorithm in the model helps to perform both classification as well 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.
  1. **CONS:**
     + The main limitation of using random forest algorithm in the model is that a large number of 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.

1. **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, fuel type, 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.

1. **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.

1. **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\_Qv7Fzslp7NDy9yWfHWExaSG',

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

* 1. **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","ModelYear","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())

* 1. **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>

* 1. **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","ModelYear","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>