PROJECT REPORT ON

CAR PERFORMANCE PREDICTION

USING

IBM WATSON MACHINE LEARNING

TABLE OF CONTENTS

1	INTRODUCTION		1
	1.1	OVERVIEW	1
	1.2	PURPOSE	1
2	LITERATURE SURVEY		2
	2.1	EXISTING PROBLEM	2
	2.2	PROPOSED SOLUTION	2
3	THEORITICAL ANALYSIS		3
	3.1	BLOCK DIAGRAM	3
	3.2	HARDWARE/SOFTWARE DESIGNING	3
4	EXPERIMENTAL INVESTIGATIONS		4
5	FLOWCHART		5
6	RESULT		6
7	ADVANTAGES & DISADVANTAGES		7
8	APPLICATIONS		8
9	CONCLUSION		9
10	FUTURE SCOPE		10
11	BIBLIOGRAPHY		11
	APPENDIX		12
	A	SOURCE CODE	12

1. Introduction

1.1 Overview

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 the engine 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 the 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 prediction engine and engine management system. This approach is the very important step towards understanding the vehicles performance.

1.2 Purpose

The automotive industry is extremely competitive. With increasing fuel prices and picky consumers, automobile makers are constantly optimizing their processes to increase fuel efficiency.

But what if you could have a reliable estimator for a car's mpg given some known specifications about the vehicle?

Then, you could beat a competitor to market by both having a more desirable vehicle that is also more efficient, reducing wasted R&D costs and gaining large chunks of the market.

2. Literature Survey

2.1 Existing Problem

According to others using this dataset, some of the mpg values for the cars are incorrect, meaning that some of our predictions will be off by a large amount, but we shouldn't always trust the listed mpg value.

It is a very time-consuming task to predict the performance of a car by actual testing. For one car it could take up to week of testing and paper work to do it properly.

Using a machine learning model, we can do the same in seconds.

2.2 Proposed Solution

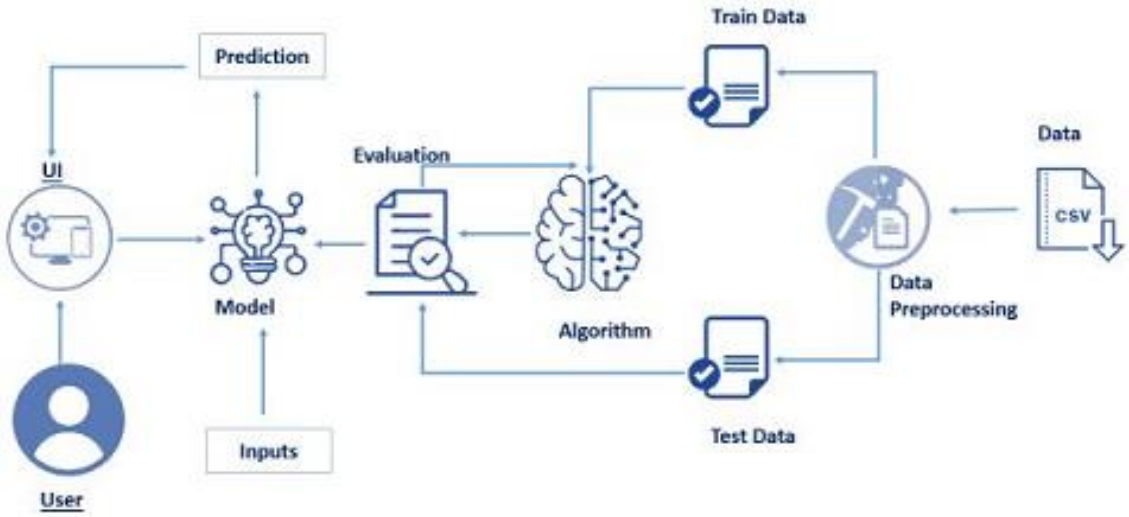
Using advanced machine learning model which is trained using the verified dataset of cars and its various attributer the model can be trained to predict the performance of car provided necessary values given.

The model can predict the performance with an accuracy of 93% given the fact that the result can be seen in seconds the model is reliable.

Anyone with prior knowledge of using a web browser can operate the application easily. Generally, a model is only as good as the data passed into it, and the data preprocessing we do ensure that the model has as accurate a dataset as possible.

3. Theoretical Analysis

3.1 Block Diagram



3.2 Hardware and Software Designing

- **IBM Watson Studio**

  Watson Studio accelerates the machine and deep learning workflows required to infuse AI into your business to drive innovation. It provides a suite of tools for data scientists, application developers and subject matter experts, allowing them to collaboratively connect to data, wrangle that data and use it to build, train and deploy models at scale. Successful AI projects require a combination of algorithms + data + team, and a very powerful compute infrastructure.

- IBM Watson Machine Learning
- IBM Cloud Object Storage

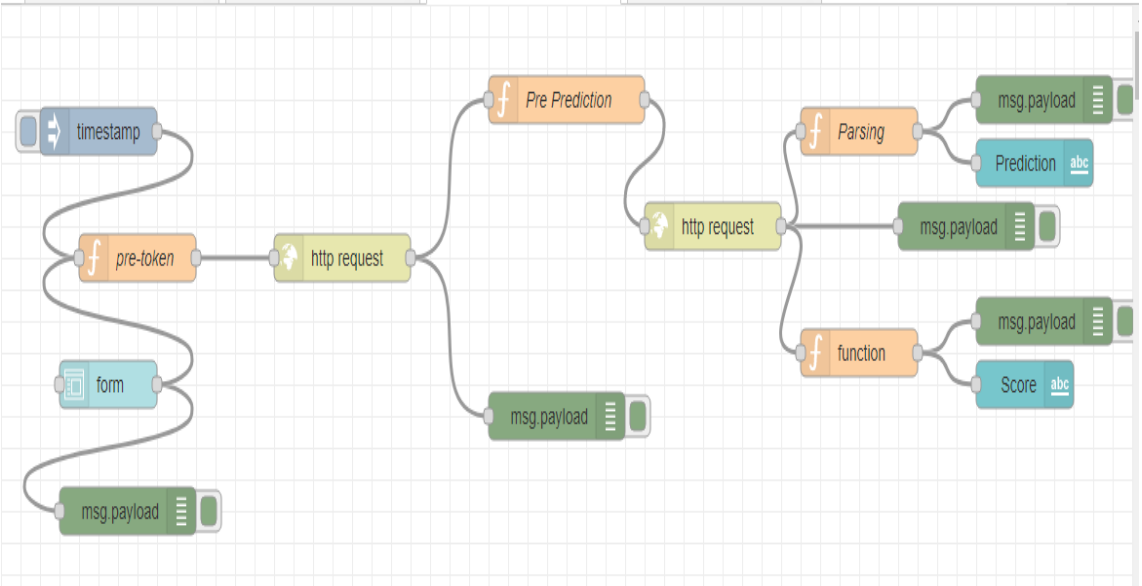
4. Experimental Investigation

In this section, we will be creating and training our model for predicting what a car's mpg will be. Since there are multiple algorithms, we can use to build our model, we will compare the accuracy scores after testing and pick the most accurate algorithm.

From this list, we are using DecisionTree, RandomForest, and KNeighborsRegressor to perform our predictions. We then see which algorithm produces the highest accuracy and select it as our algorithm of choice for future use.

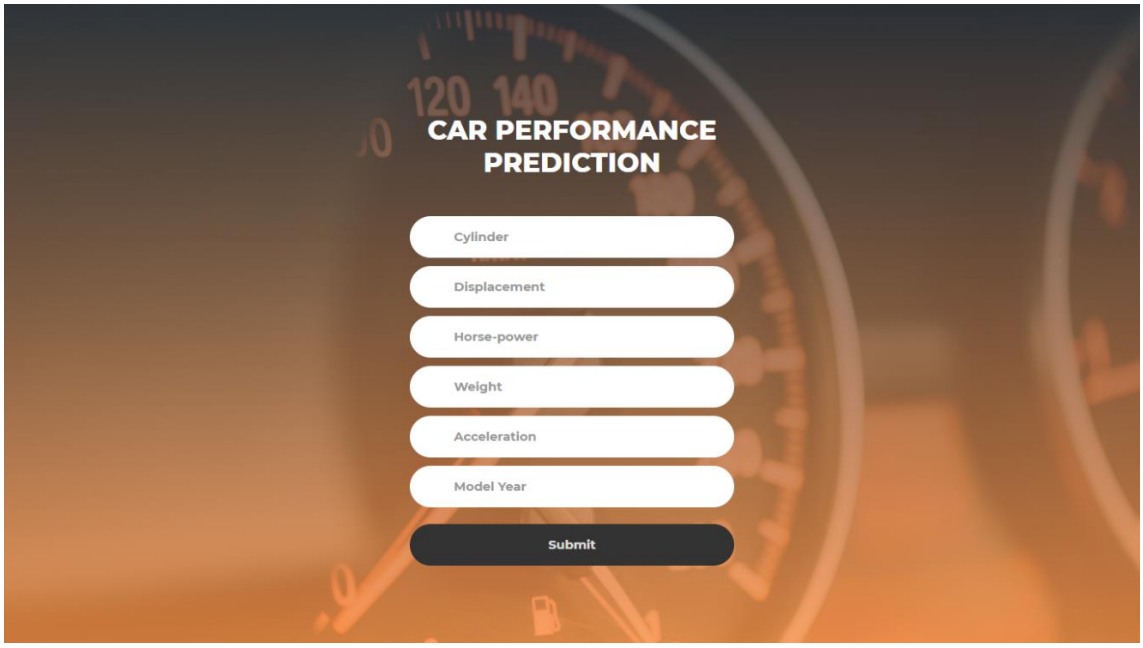
On the results of the following algorithms, we have done the conclusion the Random Forest model is the most accurate out of all the models which we have tested.

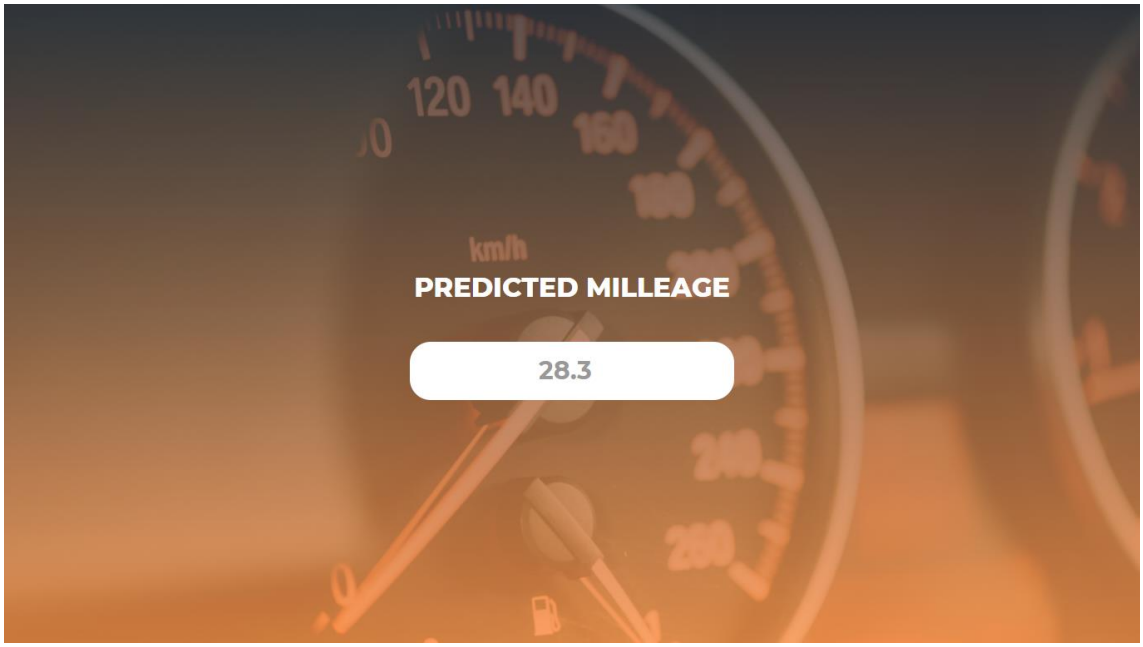
5. Flowchart


timestamp
pre-token
http request
Pre Prediction
form
msg.payload
msg.payload
http request
Parsing
msg.payload
Prediction
msg.payload
function
msg.payload
Score

6. Result

The final result of the project is the prediction of the car performance prediction. The output would be in this format:





7. Advantages and Disadvantages

Advantages

The main advantage the proposed model is it that it can predict whether certain attributes like number of cylinders, weight, acceleration, etc. have any effect on the performance of a vehicle before ever building one.

Therefore, manufactures can have extra opinion on how to create a perfect machine.

It can also help regular people to predict the performance of their vehicle by inputting the features of their car.

Disadvantages:

- Need more datasets, to increase the accuracy of the algorithms.
- The accuracy of the application depends on the dataset used to train the model
- The proposed application is Web-based, hence cannot be used in Mobile devices.
- The result of the application depends upon the accuracy of the algorithms

8. APPLICATION

By Manufacturers

Using this proposed application manufacturers can predict whether or not to use something. It can work as a simulation of how the car performs if it is tuned like this.

Using this data, they can choose whether or not to include a certain feature or not. They can fine tune the vehicle by reducing weight we can increase the performance of a car etc.

Race car Professionals

In car Racing everything is about performance by using this model they can predict the performance and fine tune the can to perform better in the race. There are a number of factors which can contribute to better performance.

9. CONCLUSION

During this notebook, we built a model that could reliably predict a car's mpg given some information about the car within 2.5 mpg of the actual value. This model could be trained with newer car data and be used to predict competitor's future mpg ratings for upcoming cars, allowing companies to potentially resources currently used on R&D today on making more efficient, more popular vehicles that outshine competitors.

While our model may be inaccurate in some cases, we talked about how our dataset can contain inaccurate values for the mpg, and oftentimes, our predictions are more accurate than the values in the dataset.

For newer cars, the collected data is significantly more reliable, so our model will be able to perform better with a different, more accurate dataset.

10. FUTURE SCOPE

The purpose of this model is to help those in the motor vehicle engineering department. We all know that every industry never stays the same so they innovate to help humans move forward as a species. With every innovation new feature will be added in the form of hardwire or software with these changes it can be a bit hard for the model to predict the performance so in the future we hope to include a better prediction model using a larger dataset of values with a lot of features from the popular car manufacturers which will help in prediction the performance more accurately

We can also redesign the web application to follows the latest trends and also so support different languages in the future.

11. BIBLIOGRAPHY

www.wikipedia.org

www.google.com

www.github.org

www.geekforgeeks.org

www.stackoverflow.com

APPENDIX

A. SOURCE CODE

```
from flask import Flask,redirect,url_for,render_template,request
import requests

API_KEY = "XvwSj9qdy7B_gr4BNjwR4p7-hlUMoZSDWjwHjgqhYZ8s"
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}

app=Flask(__name__)

@app.route('/')
def base():
    return render_template("base.html")

@app.route('/predict',methods =['POST','GET'])
def predict():
    if request.method == 'POST':

        cyl = request.form['cyl']
        dis = request.form['dis']
        hp = request.form['hp']
        w = request.form['w']
        a = request.form['a']
        my = request.form['my']

        arr=[[cyl,dis,hp,w,a,my]]
```

```
        payload_scoring = {"input_data": [{"fields":
['cylinders','displacement','horsepower','weight','acceleration','model year'],
"values": arr}]}
        response_scoring = requests.post('https://eu-
gb.ml.cloud.ibm.com/ml/v4/deployments/4551b24d-4ed8-457f-b3bc-
62dafaefc7b7/predictions?version=2022-03-05', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})

        op = response_scoring.json()
        pred = op['predictions'][0]['values'][0][0]
        pred = float("{:.2f}".format(pred))

        return render_template("predict.html",data=pred)

if __name__ == '__main__':
    app.run(debug=False)
```