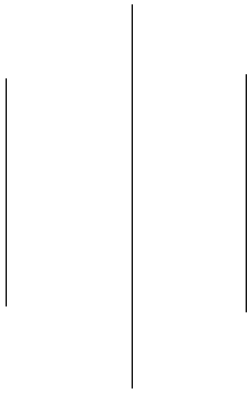
# Swarrnim Startup and Innovation University

## A Project Report On

# ENHANCING VEHICLE SAFETY:

# A MACHINE LEARNING APPROACH FOR PREDICTIVE MAINTENANCE




**Submitted By:**

**Team_2783**

**Rahul Kapar**

**Navneet Kamaliya**

**Aryan Gajjar**

**Suman Mandal**

**Madhav Nandan Singh**

**Mentored By:**

**Prof. Dulari Bhatt**

# Acknowledgement

We wish to express our sincere gratitude to our project guide **Prof. Dulari Ma'am** and all the faculty members for helping us through our project by giving us the necessary suggestions and advice along with their valuable coordination in completing this work.
We also thank our parents, friends and all the members of the family for their precious support and encouragement which they provided in the completion of our work. In addition to that, we would also like to mention the college personnel who gave me permission to use and experience the valuable resources required for the above, we once again thank the faculties and members of **Edunet Foundation, Code Unnati and SAP** for their valuable support in the completion of the project.

Thank you,

# ABSTRACT

This project report investigates the application of machine learning techniques to enhance vehicle safety through predictive maintenance. Vehicle safety is of paramount importance in modern transportation systems, with proactive maintenance playing a crucial role in preventing accidents and ensuring passenger well-being. This study aims to leverage machine learning algorithms to predict potential safety-related issues in vehicles before they manifest, thereby improving overall safety standards.

The project begins by collecting and preprocessing data from a Dataset embedded in vehicles, capturing key performance indicators, such as engine health, Lubricant oil pressure, Lubricant oil temperature, and other critical parameters. Feature engineering techniques are employed to extract relevant features from the raw sensor data, facilitating the development of robust predictive models.

# TABLE OF CONTENTS

# INTRODUCTION

Proactive precautions are required due to the growing incidence of incidents that are linked to part failures. This article describes our effort to create a machine-learning model that can forecast maintenance needs, thereby enhancing vehicle safety and lowering mechanically- related accident rates.

## OBJECTIVE

- **Create a Predictive Maintenance Model:**
  - Utilize historical maintenance data.
  - Identify patterns and trends in crucial components susceptible to malfunction.

- **Enable Proactive Interventions:**
  - Develop a model to foresee potential issues.
  - Lower the risk of accidents by intervening before critical failures occur.

# REQUIREMENT ANALYSIS

To build this project we need Anaconda Navigator, jupyter notebook, Kaggle.com web, laptop, and internet connection. The tools and technology that we'll use to build our machine-learning project are mentioned below:

## Tools and Technology used

➢ **Tools**

- Laptop
- Desktop
- Wi-Fi

➢ **Technology**

- Machine Learning
- Kaggle.com

# IMPORTANCE AND URGENCY

The development of a predictive model for automobile servicing carries immense importance and urgency in the realm of vehicle safety and maintenance. With road accidents often stemming from mechanical failures, the ability to forecast the need for servicing becomes paramount in preventing potential hazards. By accurately predicting maintenance requirements, this model aids in preemptively addressing issues before they escalate, thereby significantly reducing the risk of accidents caused by vehicle malfunctions. Moreover, in a world where road safety is of utmost concern, the urgency of such a model cannot be overstated. Every moment without proactive maintenance increases the likelihood of a potential safety hazard on the road. Therefore, the timely deployment of this model is crucial in safeguarding both drivers and pedestrians, promoting a safer and more secure transportation environment.

# METHODOLOGY

In our project we used different algorithms and techniques to check the accuracy, report and the confusion matrix. The name of algorithms we had used with their respective accuracy, report and confusion matrix are give below:

## -Logistics Regression:

```
In [10]: accuracy = accuracy_score(Y_test, Y_pred)
         print("Accuracy:", accuracy)

         Accuracy: 0.6527896263436274

In [11]: print("\nClassification Report:")
         print(classification_report(Y_test, Y_pred))

         Classification Report:
                       precision    recall  f1-score   support

                    0       0.58      0.28      0.38      2214
                    1       0.67      0.88      0.76      3647

             accuracy                           0.65      5861
            macro avg       0.63      0.58      0.57      5861
         weighted avg       0.64      0.65      0.62      5861

In [12]: print("\nConfusion Matrix:")
         print(confusion_matrix(Y_test, Y_pred))

         Confusion Matrix:
         [[ 618 1596]
          [ 439 3208]]
```

## -XG Boost:

```
In [17]: accuracy = accuracy_score(Y_test, Y_pred)
         print("Accuracy:", accuracy)

         Accuracy: 0.6432349428425184

In [18]: print("\nClassification Report:")
         print(classification_report(Y_test, Y_pred))

         Classification Report:
                       precision    recall  f1-score   support

                    0       0.54      0.38      0.44      2214
                    1       0.68      0.81      0.74      3647

             accuracy                           0.64      5861
            macro avg       0.61      0.59      0.59      5861
         weighted avg       0.63      0.64      0.63      5861

In [19]: print("\nConfusion Matrix:")
         print(confusion_matrix(Y_test, Y_pred))

         Confusion Matrix:
         [[ 831 1383]
          [ 708 2939]]
```

## -Random Forest:

```
In [31]: accuracy_score(Y_test,test_preds)
Out[31]: 0.655860774611841

In [32]: print("\nClassification Report:")
         print(classification_report(Y_test, Y_pred))

Classification Report:
              precision    recall  f1-score   support

           0       0.57      0.32      0.41      2214
           1       0.67      0.86      0.75      3647

    accuracy                           0.65      5861
   macro avg       0.62      0.59      0.58      5861
weighted avg       0.64      0.65      0.62      5861

In [33]: print("\nConfusion Matrix:")
         print(confusion_matrix(Y_test, Y_pred))

Confusion Matrix:
[[ 708 1506]
 [ 526 3121]]
```

## -Support Vector Machine:

```
In [24]: accuracy = accuracy_score(Y_test, Y_pred)
         print("Accuracy:", accuracy)
Accuracy: 0.6533014843883297

In [25]: print("\nClassification Report:")
         print(classification_report(Y_test, Y_pred))

Classification Report:
              precision    recall  f1-score   support

           0       0.57      0.32      0.41      2214
           1       0.67      0.86      0.75      3647

    accuracy                           0.65      5861
   macro avg       0.62      0.59      0.58      5861
weighted avg       0.64      0.65      0.62      5861

In [26]: print("\nConfusion Matrix:")
         print(confusion_matrix(Y_test, Y_pred))

Confusion Matrix:
[[ 708 1506]
 [ 526 3121]]
```

## -Neural Network:

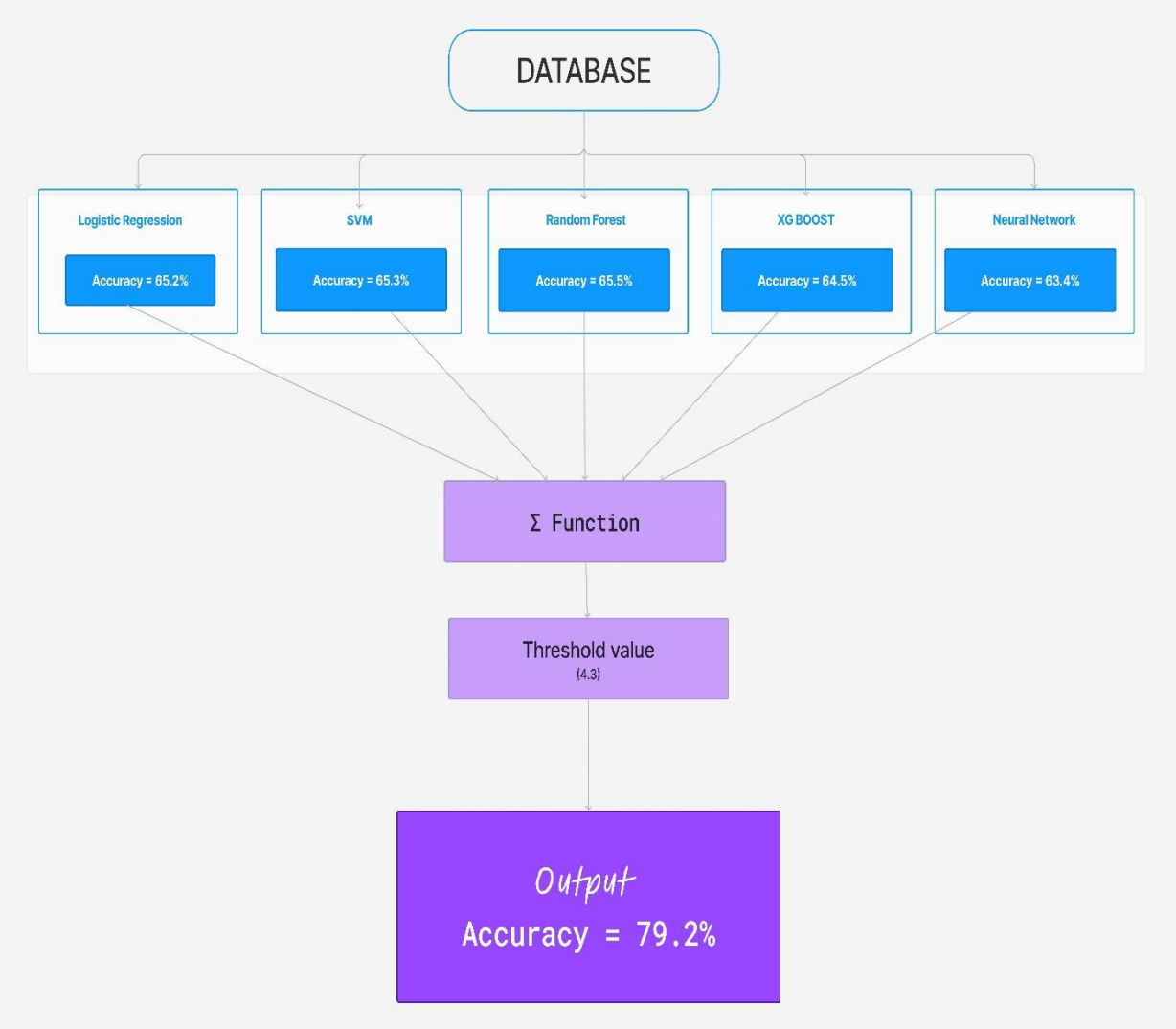
```
In [37]: model_nn.fit(X, Y, epochs=11, batch_size=5)

Epoch 1/11
3907/3907 ━━━━━━━━━━━━━━━━━━━━ 10s 2ms/step - accuracy: 0.6055 - loss: 1.5500
Epoch 2/11
3907/3907 ━━━━━━━━━━━━━━━━━━━━ 10s 2ms/step - accuracy: 0.6270 - loss: 0.6607
Epoch 3/11
3907/3907 ━━━━━━━━━━━━━━━━━━━━ 7s 2ms/step - accuracy: 0.6266 - loss: 0.6610
Epoch 4/11
3907/3907 ━━━━━━━━━━━━━━━━━━━━ 8s 2ms/step - accuracy: 0.6286 - loss: 0.6606
Epoch 5/11
3907/3907 ━━━━━━━━━━━━━━━━━━━━ 7s 2ms/step - accuracy: 0.6250 - loss: 0.6614
Epoch 6/11
3907/3907 ━━━━━━━━━━━━━━━━━━━━ 7s 2ms/step - accuracy: 0.6306 - loss: 0.6585
Epoch 7/11
3907/3907 ━━━━━━━━━━━━━━━━━━━━ 7s 2ms/step - accuracy: 0.6372 - loss: 0.6548
Epoch 8/11
3907/3907 ━━━━━━━━━━━━━━━━━━━━ 7s 2ms/step - accuracy: 0.6369 - loss: 0.6550
Epoch 9/11
3907/3907 ━━━━━━━━━━━━━━━━━━━━ 7s 2ms/step - accuracy: 0.6316 - loss: 0.6580
Epoch 10/11
3907/3907 ━━━━━━━━━━━━━━━━━━━━ 7s 2ms/step - accuracy: 0.6304 - loss: 0.6600
Epoch 11/11
3907/3907 ━━━━━━━━━━━━━━━━━━━━ 7s 2ms/step - accuracy: 0.6345 - loss: 0.6553

Out[37]: <keras.src.callbacks.history.History at 0x1c6813830d0>
```

These are the methods we used in our project and got the accuracy of 62 to 66 percentage. But additionally in our project we used one more layer of summation function and put the threshold to 4.3 and shockingly got the accuracy of 79.2 percentage. It increases by almost 15 percentage which makes our model more effective in identifying the need of maintenance or not. This is the shocking result we had got after making the modifications in traditional techniques.

```
In [17]: print("\nAccuracy:")
         print(accuracy)

Accuracy:
0.7921603042270001

In [18]: print("\nClassification Report:")
         print(classification_report(Y_train, Y_pred))

Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.75      0.72      5004
           1       0.85      0.82      0.83      8670

    accuracy                           0.79     13674
   macro avg       0.78      0.78      0.78     13674
weighted avg       0.80      0.79      0.79     13674

In [19]: print("\nConfusion Matrix:")
         print(confusion_matrix(Y_train, Y_pred))

Confusion Matrix:
[[3740 1264]
 [1578 7092]]
```

# SYSTEM DESIGN

The architecture of the program is given below:




Initially, we have a dataset which is later divided into train set and test set. The train set is passed to the different algorithms mentioned in the above diagram and got the accuracy on test set as mentioned earlier. These algorithms give the output in either 0 or 1 except Neural Network where 0 means servicing is needed and 1 means not.

Neural network give output in range of 0 to 1. The output is stored in a csv file.

The outputs from these algorithms are added together by our summation function which we are doing manually in MS Excel for now. Later we will develop a program for the automation. The output generated from the summation function is passed through a threshold function, where we find the value of 4.3 and 4.4 is best suited for our dataset. The threshold function what does is, if the output generated but summation function is smaller equal to 4.3 then make a output of 0 and 1 if greater. All these is done in the same excel file where summation function works. After threshold function we get a new column of 0s and 1s which is compared to the train set. And in our case we got the accuracy of 79.2 percentage.

# IMPACT ON VEHICLE SAFETY

Implementing a predictive model for automobile servicing can have a significant impact on vehicle safety. By accurately predicting the need for servicing, this model helps identify potential issues before they escalate into safety hazards. Preventive maintenance can address problems such as brake failures, engine malfunctions, or tire wear, which are critical factors affecting road safety. With an accuracy of 79%, the model provides reliable insights, enabling timely interventions and reducing the likelihood of accidents due to mechanical failures. Moreover, by promoting proactive maintenance schedules, the model contributes to prolonging the lifespan of vehicles, ensuring they remain in optimal condition on the road. Overall, this advancement in AI/ML engineering not only enhances vehicle performance and reliability but also plays a vital role in enhancing road safety for drivers and passengers alike.

# CONCLUSION AND FUTURE ENHANCEMENT

In conclusion, the development and implementation of a predictive model for automobile servicing represent a significant stride forward in enhancing vehicle safety. With an accuracy rate of 79%, this model proves to be a valuable tool in identifying and addressing potential mechanical issues before they pose safety risks on the road. By enabling proactive maintenance measures, the model not only improves vehicle reliability and performance but also contributes to reducing the incidence of accidents caused by preventable mechanical failures. Moving forward, continued advancements in AI and machine learning in the automotive industry hold promise for further improving road safety and ensuring a safer driving experience for all.

# REFRENCE


1. https://chat.openai.com/
2. https://github.com/Code-Unnati/Advance-Course
3. https://www.kaggle.com/