

Text  Description automatically generated with low confidence

**Futuristic Analytics for Pump Health: Machine-Driven Forecasting of Remaining Useful Life**

SmartInternz

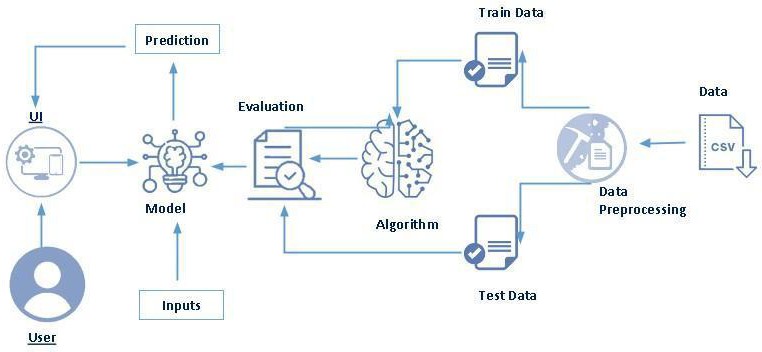
[www.smartinternz.com](http://www.smartinternz.com/)

**Futuristic Analytics for Pump Health: Machine-Driven Forecasting of Remaining Useful Life**

# Project Description:

In this project, This project aims to develop a Predictive Maintenance model leveraging raw sensor data to predict the remaining useful life of critical equipment. By predicting maintenance needs based on real-time sensor values, the model empowers industries, including manufacturing, energy, and transportation, to optimize maintenance schedules, minimize downtime, and prevent unexpected failures. The dataset used in this project will be collected from various sources such as online surveys, social media platforms, and other publicly available data sources. The data will be pre-processed and cleaned to ensure quality and eliminate noise or missing values. once the data is cleaned, it will be split into training and testing sets. Several machine learning models will be built and evaluated on the training data to determine the best-performing model. The models to be explored include linear regression, random forests, and SVM. After selecting the best-performing model, it will be used to predict the satisfaction level of the passengers in the testing set. The model's performance will be evaluated based on various metrics such as R\_Square, mse, mae, and mape.

# Technical Architecture:



# Pre requisites:

**To complete this project, you must required following software’s, concepts and packages**

* + **Anaconda navigator and pycharm:**
* Refer the link below to download anaconda navigator
* Link : <https://youtu.be/1ra4zH2G4o0>
* **Python packages:**
* Open anaconda prompt as administrator
* Type “pip install numpy” and click enter.
* Type “pip install pandas” and click enter.
* Type “pip install scikit-learn” and click enter.
* Type “pip install seaborn” and click enter.
* Type ”pip install matplotlib” and click enter.
* Type ”pip install scipy” and click enter.
* Type ”pip install pickle-mixin” and click enter.
* Type “pip install Flask” and click enter.

# Prior Knowledge:

You must have prior knowledge of the following topics to complete this project.

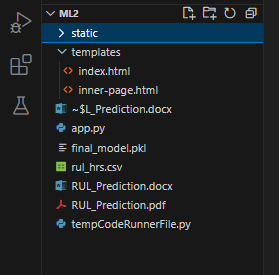
* ML Concepts
* Supervised learning: <https://www.javatpoint.com/supervised-machine-learning>
* Linear Regression: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-linear-regression/
* Decision tree Regression: https://medium.com/analytics-vidhya/regression-trees-decision-tree-for-regression-machine-learning
* Random forest Regression <https://towardsdatascience.com/random-forest-regression>
* SVM Regressor:
* Evaluation metrics: https://towardsdatascience.com/3-evaluation-metrics-for-regression-80cb34cee0e8
* Flask Basics: <https://www.youtube.com/watch?v=lj4I_CvBnt0>

**Project Flow:**

* User interacts with the UI to enter the input.
* Entered input is analyzed by the model which is integrated.
* Once the model analyses the input the prediction is showcased on the UI To accomplish this, we have to complete all the activities listed below,
* Define Problem / Problem Understanding
  + Specify the business problem
  + Business requirements
  + Literature Survey
  + Social or Business Impact.
* Data Collection & Preparation
  + Collect the dataset
  + Data Preparation
* Exploratory Data Analysis
  + Descriptive statistical
  + Visual Analysis
* Model Building
  + Training the model in multiple algorithms
  + Testing the model
* Performance Testing & Hyperparameter Tuning
  + Testing model with multiple evaluation metrics
  + Comparing model accuracy before & after applying hyperparameter tuning
* Model Deployment
  + Save the best model
  + Integrate with Web Framework
* Project Demonstration & Documentation
  + Record explanation Video for project end-to-end solution
  + Project Documentation- step-by-step project development procedure

# Project Structure:

Create the Project folder which contains files as shown below



* We are building a Flask application that needs HTML pages stored in the templates folder and a Python script app.py for scripting.
* model.pkl will be our saved model. Further, we will use this model for flask integration.
* The training folder contains a model training file.

# Milestone 1: Define Problem / Problem Understanding

## Activity 1: Specify the business problem

Refer Project Description

## Activity 2: Business requirements

## To ensure the success of this endeavor, we will conduct a thorough analysis of business requirements. This includes understanding the unique needs and challenges of industries that heavily rely on pumps. The project will aim to align with industry standards, comply with regulatory frameworks, and incorporate user-friendly interfaces for seamless integration into existing workflows. The ultimate objective is to deliver a solution that not only meets technical specifications but also adds significant value to businesses through enhanced pump maintenance strategies.

## Activity 3: Literature Survey (Student Will Write)

## In the literature survey phase, the project team will delve into existing research and publications related to predictive maintenance, machine learning applications in industrial settings, and specifically, pump health forecasting. By reviewing and synthesizing relevant literature, the team aims to identify state-of-the-art methodologies, emerging trends, and potential challenges in the domain. This comprehensive understanding will serve as the foundation for designing an advanced and effective predictive analytics solution for pump health.

## Activity 4: Social or Business Impact.

## Social Impact: The optimized operation and maintenance of pumps result in lower energy consumption, aligning with environmental conservation goals. This positively impacts the carbon footprint of industries and supports efforts toward greener practices The project contributes to increased safety in industries by minimizing the risk of unexpected pump failures. This is particularly crucial in sectors such as manufacturing and energy, where equipment malfunctions can have severe safety implications.

## Business Impact: The project significantly reduces operational costs by minimizing unplanned downtime and optimizing maintenance schedules. Businesses benefit from lower repair expenses, reduced overtime wages, and improved overall cost efficiency.

# Milestone 2: Data Collection & Preparation

ML depends heavily on data. It is the most crucial aspect that makes algorithm training possible. So, this section allows you to download the required dataset.

## Activity 1: Collect the dataset

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.

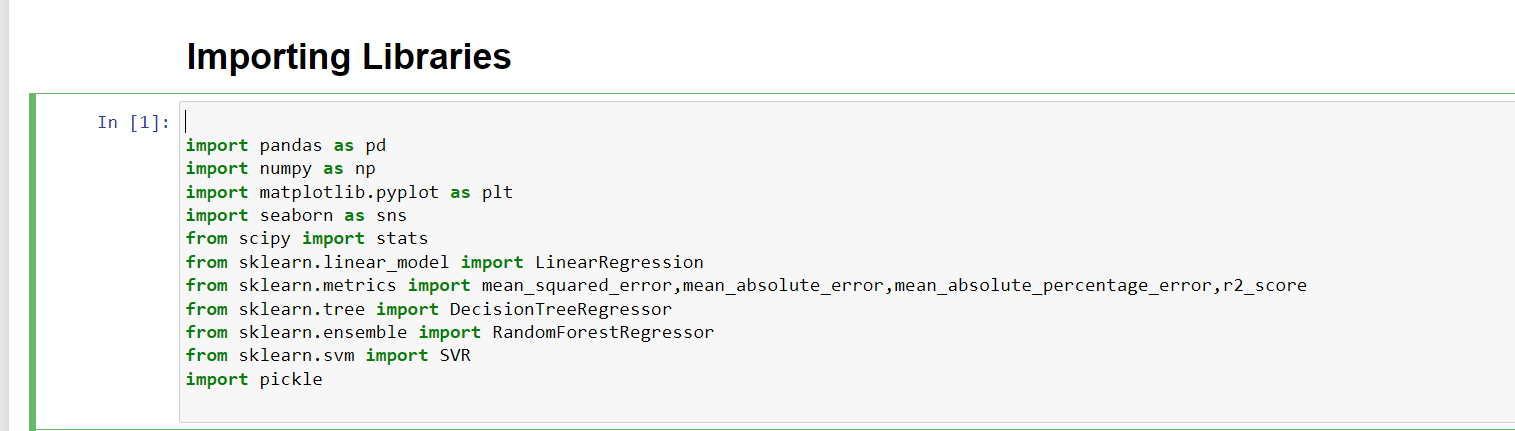
In this project, we have used .csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

Link: <https://www.kaggle.com/datasets/anseldsouza/water-pump-rul-predictive-maintenance/data>

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualization techniques and some analyzing techniques.

**Note:** There are a number of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

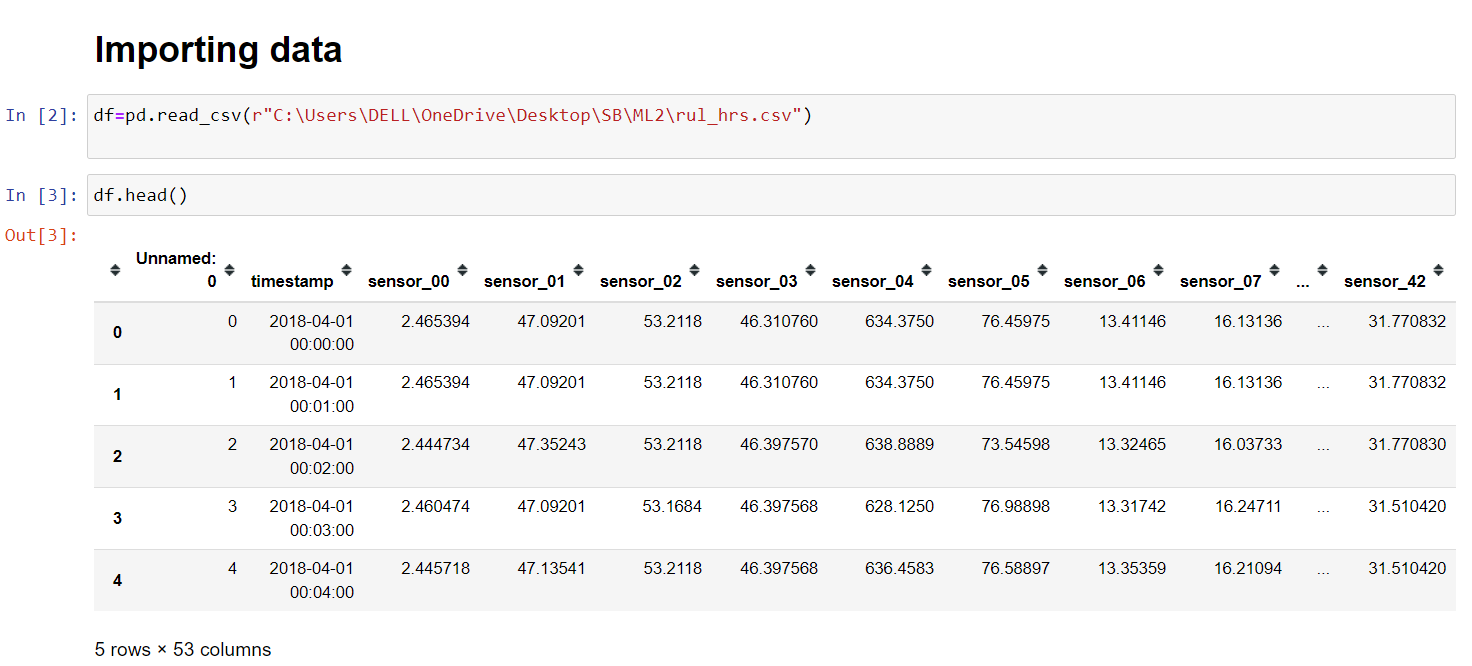
## Activity 1.1: Importing the libraries

Import the necessary libraries as shown in the image.

## Activity 1.2: Read the Dataset

Our dataset format might be in .csv, excel files, etc. We can read the dataset with the help of pandas.

In pandas, we have a function called read\_csv() to read the dataset. As a parameter, we have to give the directory of the csv file.



## Activity 2: Data Preparation

As we have understood how the data is, let's pre-process the collected data.

The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

* Data Cleaning
* Handling missing values
* Handling Outliers
* Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

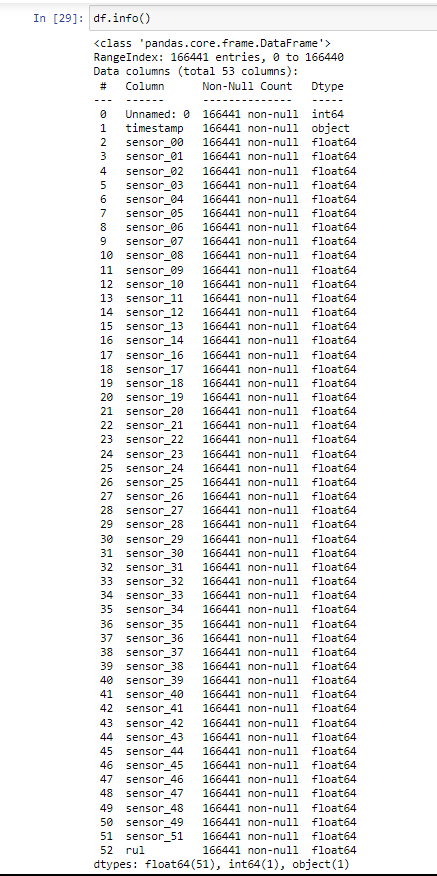
## Activity 2.1 Data Cleaning.

## 

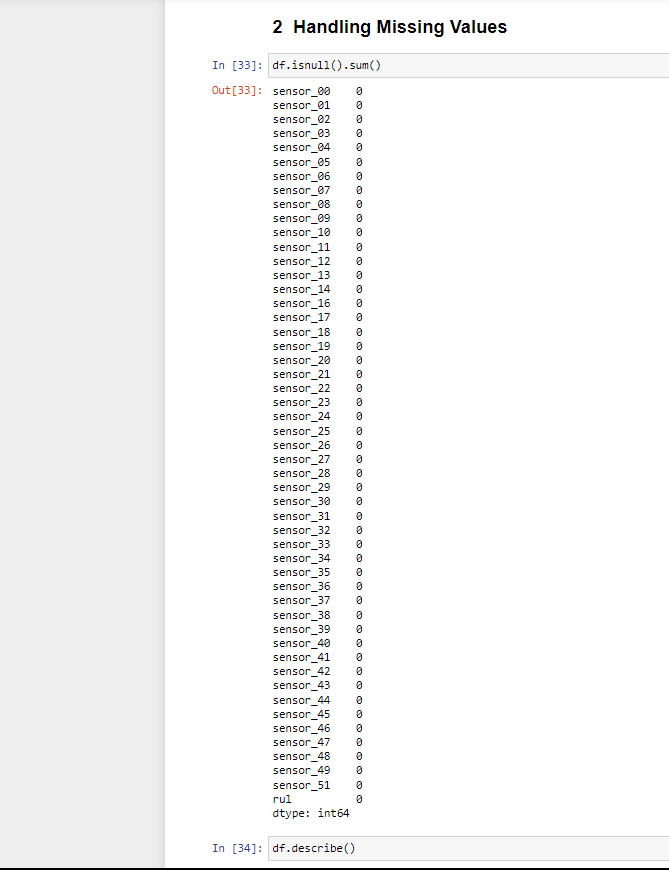
## Activity 2.2: Handling missing values

* Let’s find the shape of our dataset first. To find the shape of our data, the df.shape method.

To find the data type, df.info() function is used.



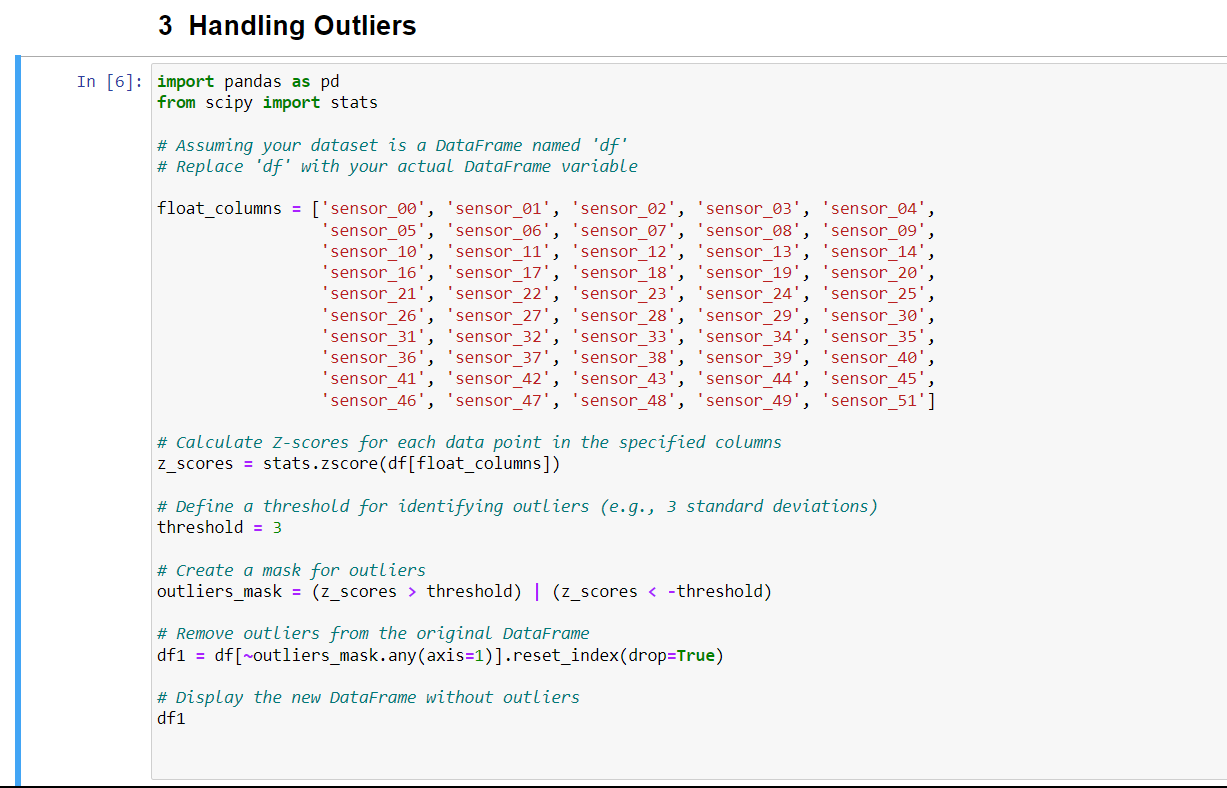
* + For checking the null values, df.isnull() function is used. To sum those null values we use .sum() function. From the below image we found that there are no null values present in our dataset. So we can skip handling the missing values step.



Since no missing values in our dataset, we need not deal with that.

## Activity 2.4: Handling Outliers

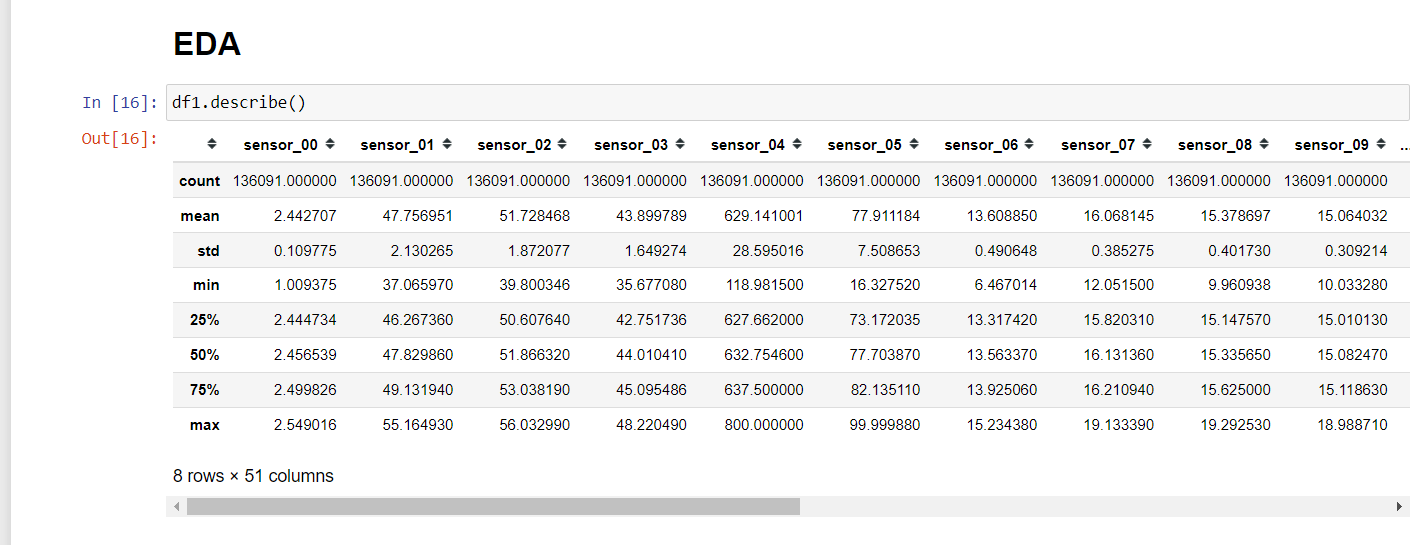
In our dataset, all the variables are numeric so we will use z-score methos to remove outliers. The data point which beyond the 3-sigma band of the normal distribution will be considered an outlier and we will remove this as our dataset quite large that will not affect on further results.



# Milestone 3: Exploratory Data Analysis

## Activity 1: Descriptive statistics

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.

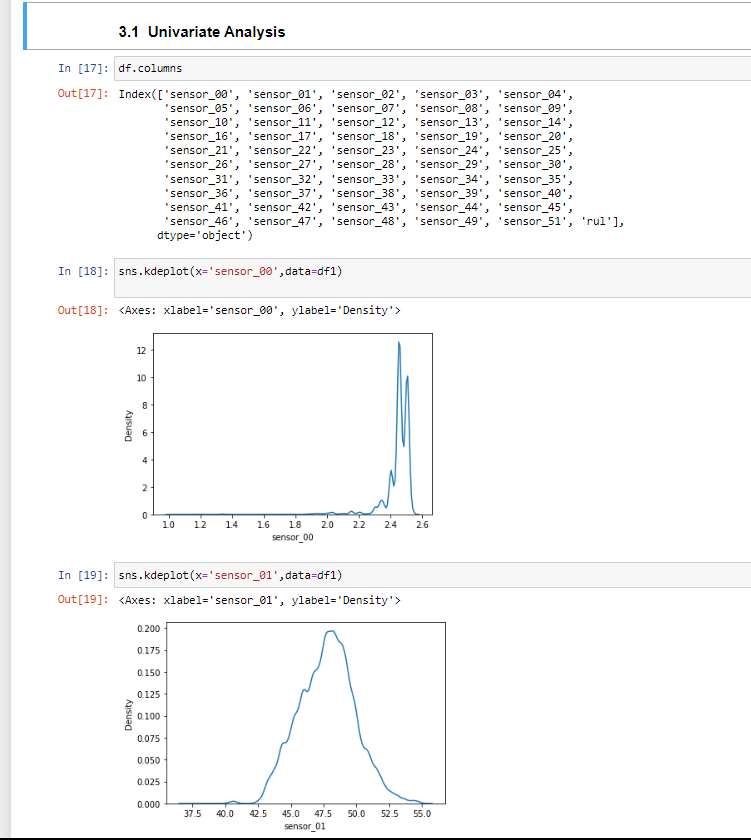


## Activity 2: Visual analysis

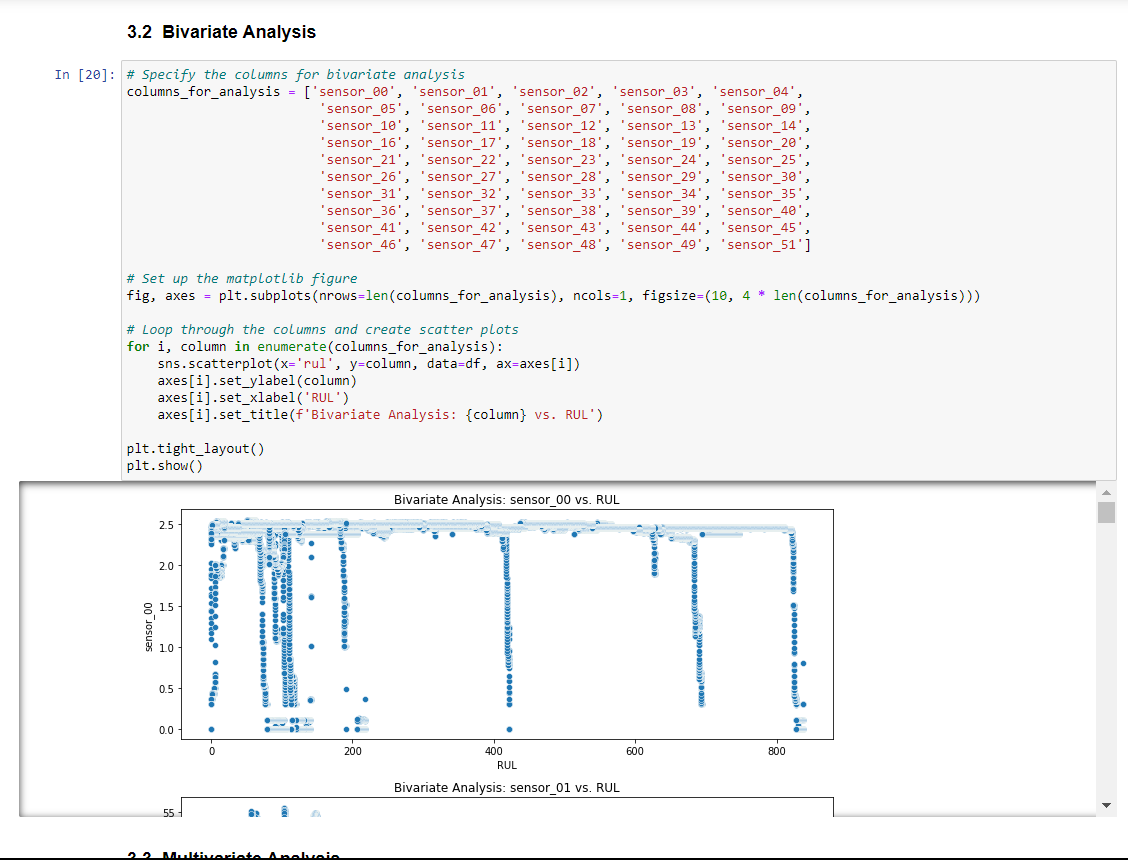
Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

## Activity 2.1: Univariate analysis

In simple words, univariate analysis is understanding the data with a single feature

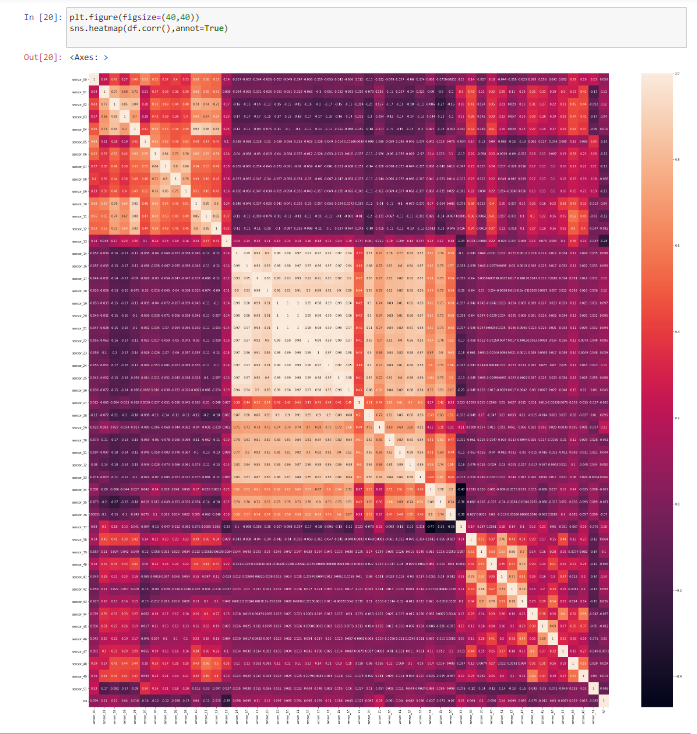
Seaborn package provides a wonderful function kdeplot (distribution plot), where we can find the distribution of the feature. To make multiple graphs in a single plot, we use subplot. 

**Activity 2.2: Bivariate Analysis**

To find the relation between two features we use bivariate analysis. Here we are visualizing the relationship between RUL values and Sensors.

## Activity 2.3: Multivariate analysis

In simple words, multivariate analysis is to find the relation between multiple features. Here we have used a heatmap for continuous variables.



**Splitting data into train and test/ (data Preparation Encoding to category features)**

First we encoding the features so that we can use them for Machine Learning Now let’s split the Dataset into train and test sets. First, split the dataset into x and y and then split the data set.

Here x and y variables are created. On the x variable, df is passed by dropping the target variable. And on y target variable is passed. For splitting training and testing data we are using the train\_test\_split() function from sklearn. As parameters, we are passing x, y, test\_size, random\_state.



# Milestone 4: Model Building

## Activity 1: Training the model in multiple algorithms

Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms. For this project, we are applying Regression algorithms. The best model is saved based on its performance.

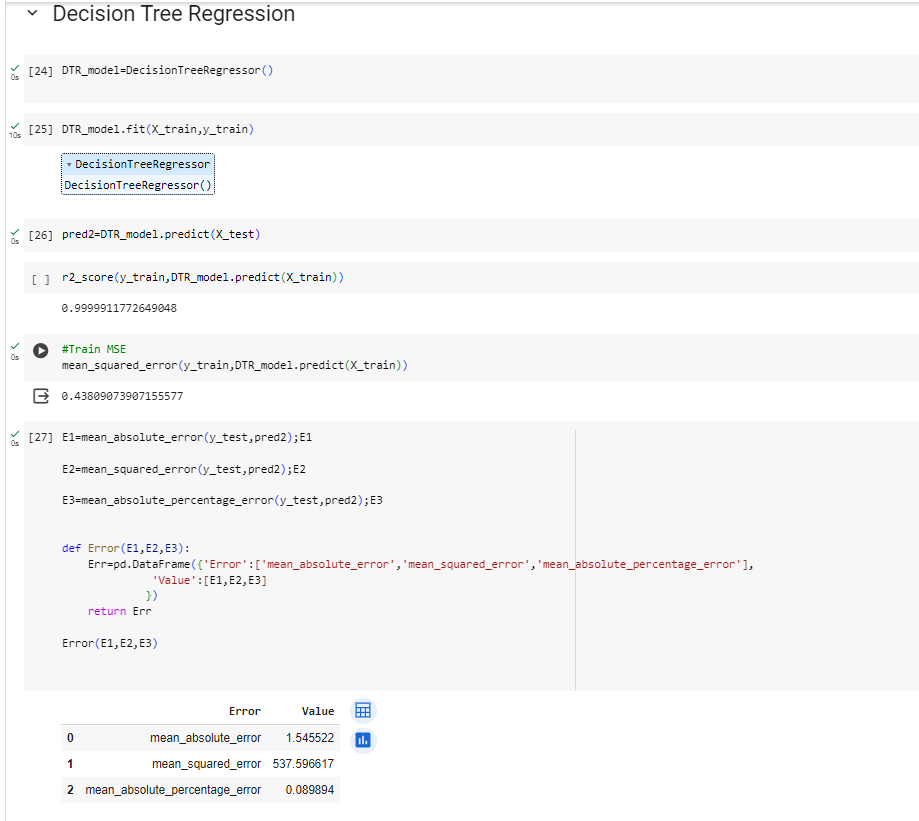
## Activity 1.1: Linear Regression

A function named LinearRegression() is created and train and test data are passed as the parameters. Inside the function,LinearRegression algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with the predict() function and saved in a new variable. For evaluating the model, MSE, MAE, MAPE R\_Square.

## 

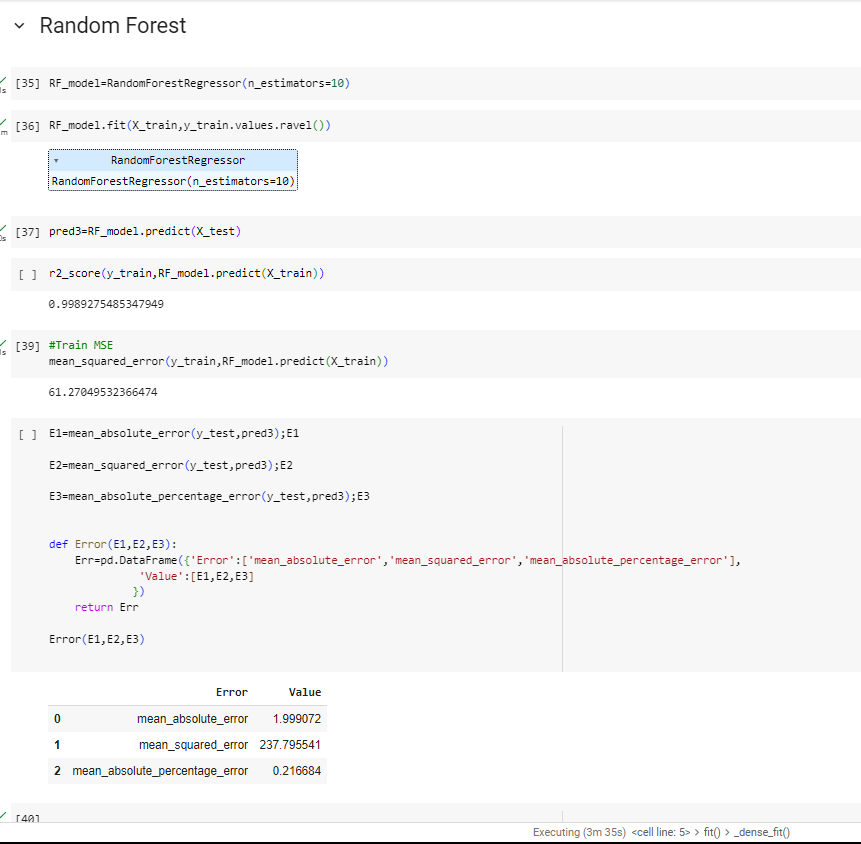
## Activity 1.2: Decision tree model

A function named decisionTree is created and train and test data are passed as the parameters. Inside the function, the Linear\_Regression, algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with the .predict() function and saved in a new variable. For evaluating the model, MSE, MAE, MAPE, and R\_Square.



## Activity 1.3: Random forest model

A function named randomForest is created and train and test data are passed as the parameters. Inside the function, RandomForestRegressor algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with the predict() function and saved in a new variable. For evaluating the model, MSE, MAE, MAPE R\_Square.

–––

## Activity 1.4: SVR (Regression)

A function named SVR is created and train and test data are passed as the parameters. Inside the function, the SVR algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with the .predict() function and saved in a new variable. For evaluating the model, MSE, MAE, MAPE R\_Square.



# Milestone 5: Performance Testing & Hyperparameter Tuning

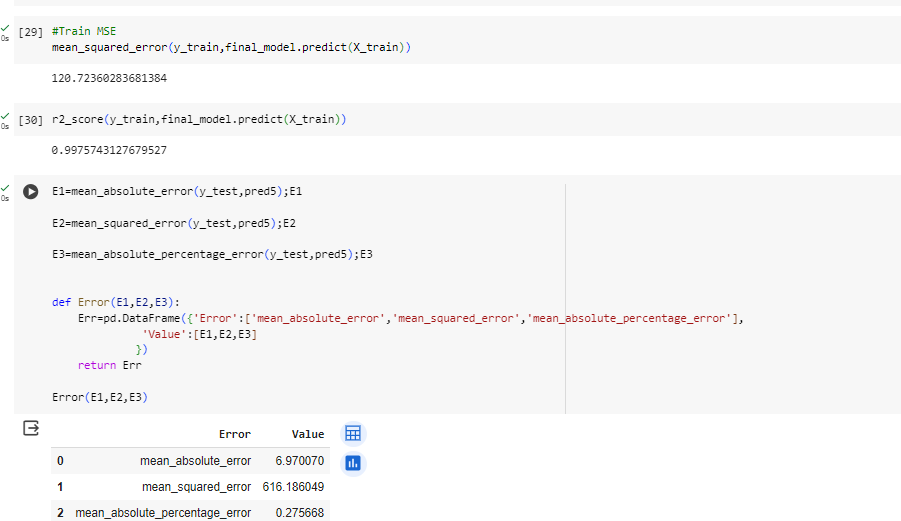
## Activity 1: Testing model with multiple evaluation metrics

Multiple evaluation metrics means evaluating the model's performance on a test set using different performance measures. This can provide a more comprehensive understanding of the model's strengths and weaknesses. We are using evaluation metrics for Regression tasks including MSE, MAE, MAPE, R\_Square

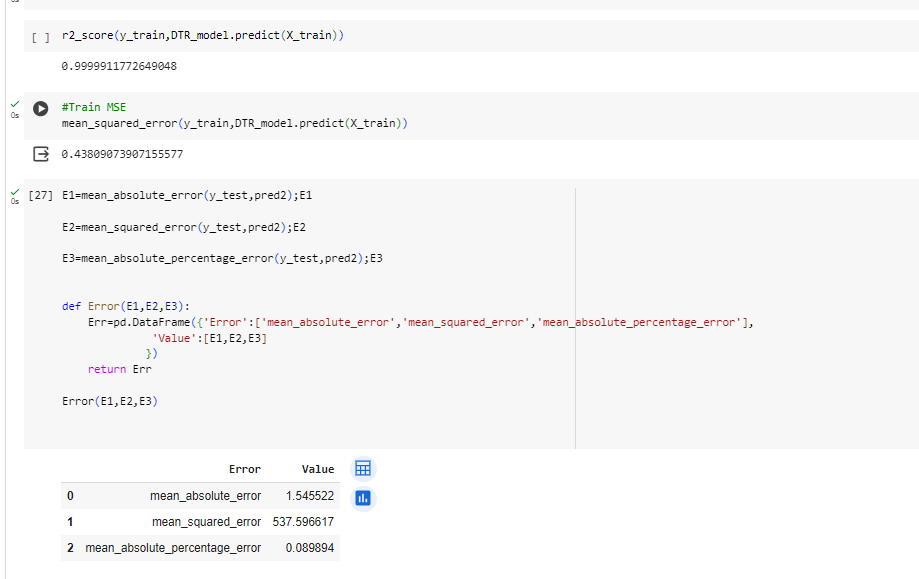
## Activity 1.1: Compare the model

From the above model, the random forest Regressor is performing well. From the below image, we can see the accuracy of the model here random forest is selected and evaluated with cross-validation. Additionally, we can tune the model with hyperparameter tuning techniques for the best model Random Forest.

**Linear Regression:**

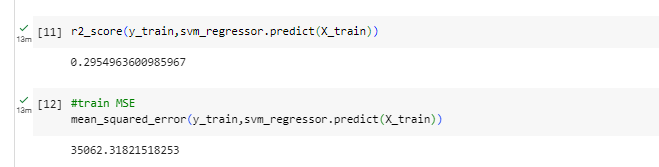
****

**Decision Tree:**

****

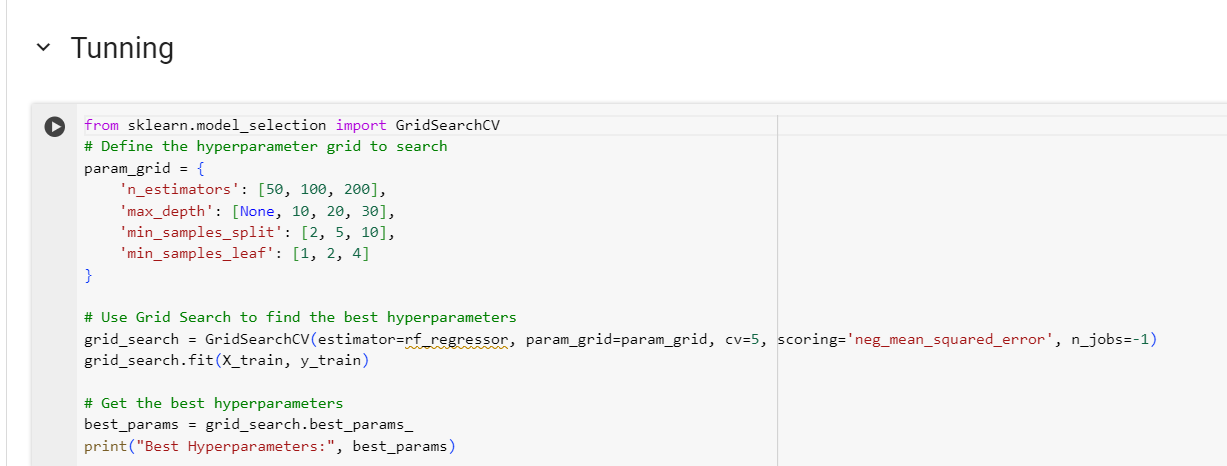
**Random Forest:**

****

**Support Vector Regressor:  
**

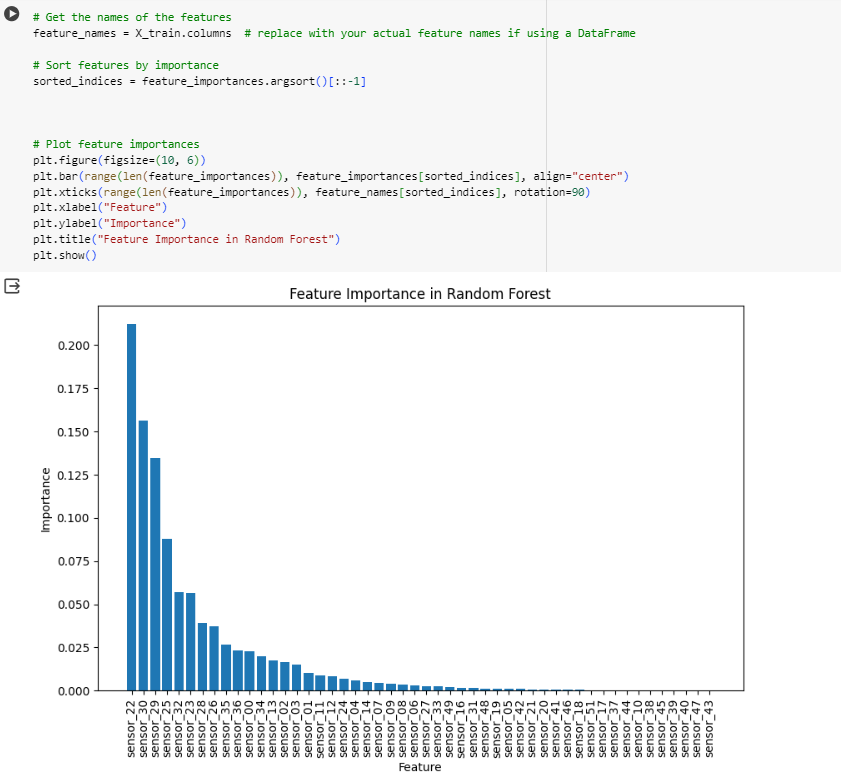
****

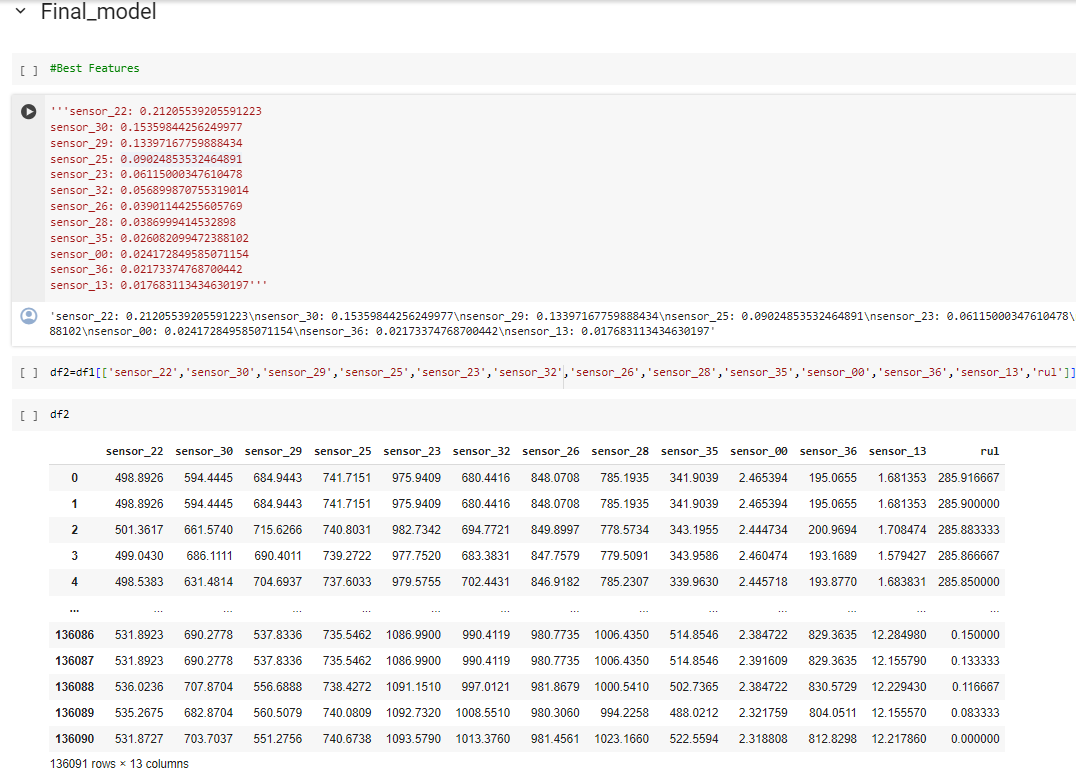
## Activity 1.2 Hyperparameter Tunning



From the all models, the random forest is performing well

## We can not take all features for the deployment so we select the best features by Random feature selection technique



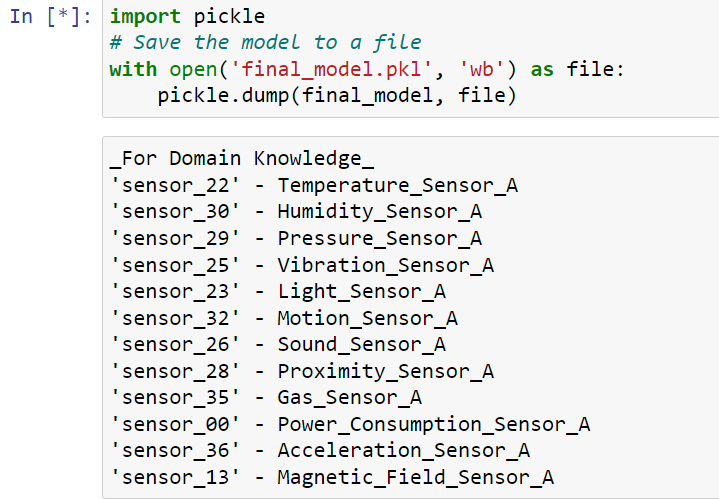


# 

# Milestone 6: Model Deployment

## Activity 1: Save the best model

Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance and saving its weights and configuration. This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future.



## Activity 2: Integrate with Web Framework

In this section, we will be building a web application that is integrated with the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

* Building HTML Pages
* Building server-side script
* Run the web application

## Activity 2.1: Building Html Pages:

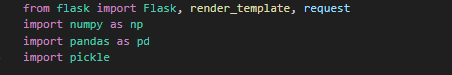
For this project create HTML files namely

* index.html
* inner-page.html

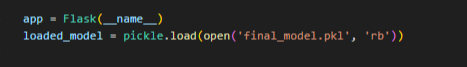
and save them in the templates folder. Refer to this [link](https://drive.google.com/drive/folders/1K-C2uvRstV8x6bwsOY_BWBvXdzjbbfPF?usp=share_link) for templates.

## Activity 2.2: Build Python code:

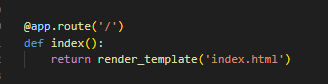
Import the libraries



Load the saved model. Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module ( name ) as argument.



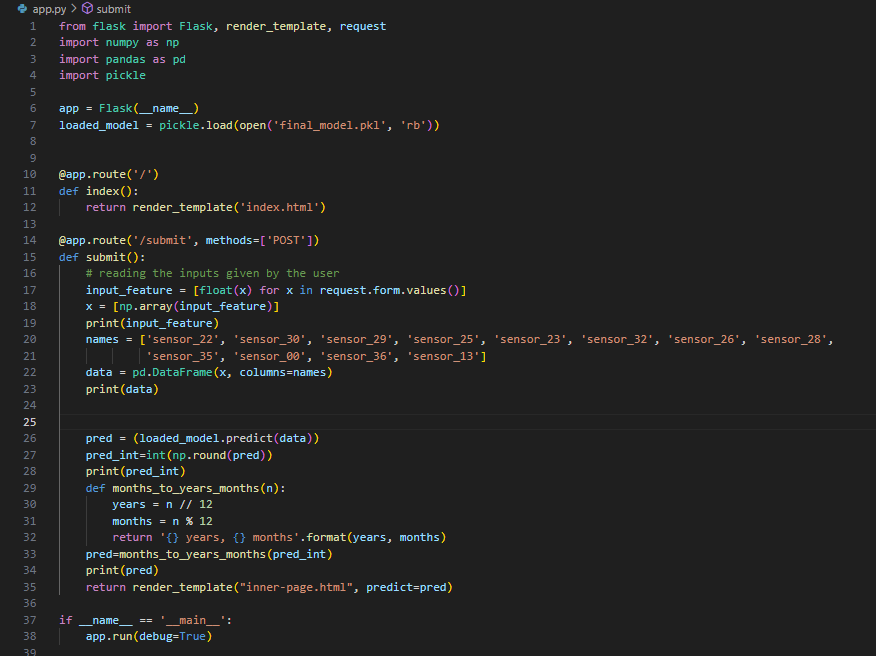
Render HTML page:



Here we will be using a declared constructor to route to the HTML page which we have created earlier.

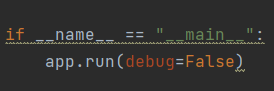
In the above example, ‘/’ URL is bound with the home.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

Retrieves the value from UI:



Here we are routing our app to predict() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will be rendered to the text that we have mentioned in the submit.html page earlier.

Main Function:



## Activity 2.3: Run the web application

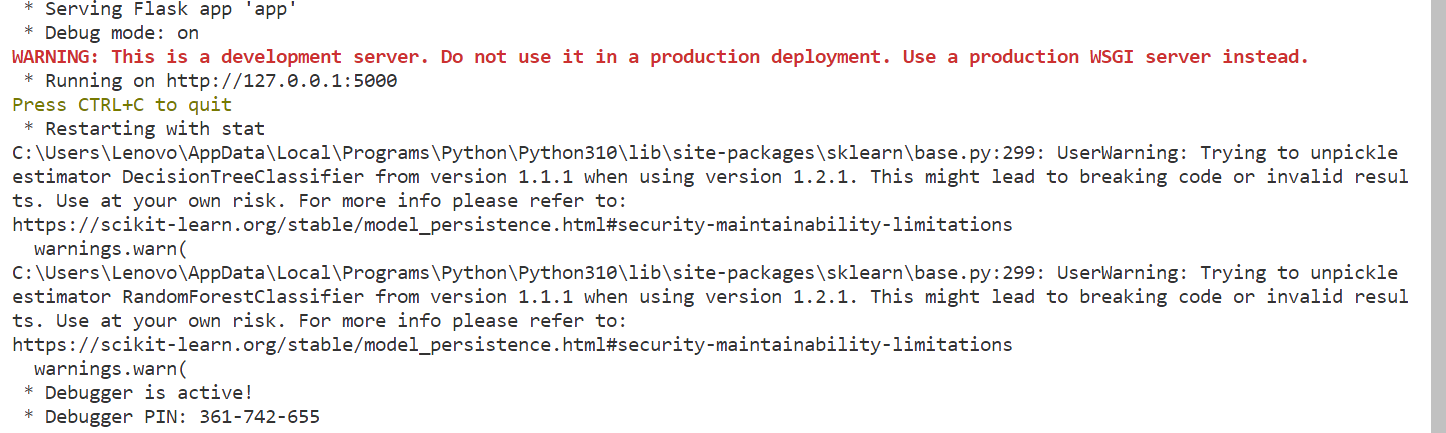
Open anaconda prompt from the start menu

Navigate to the folder where your python script is.

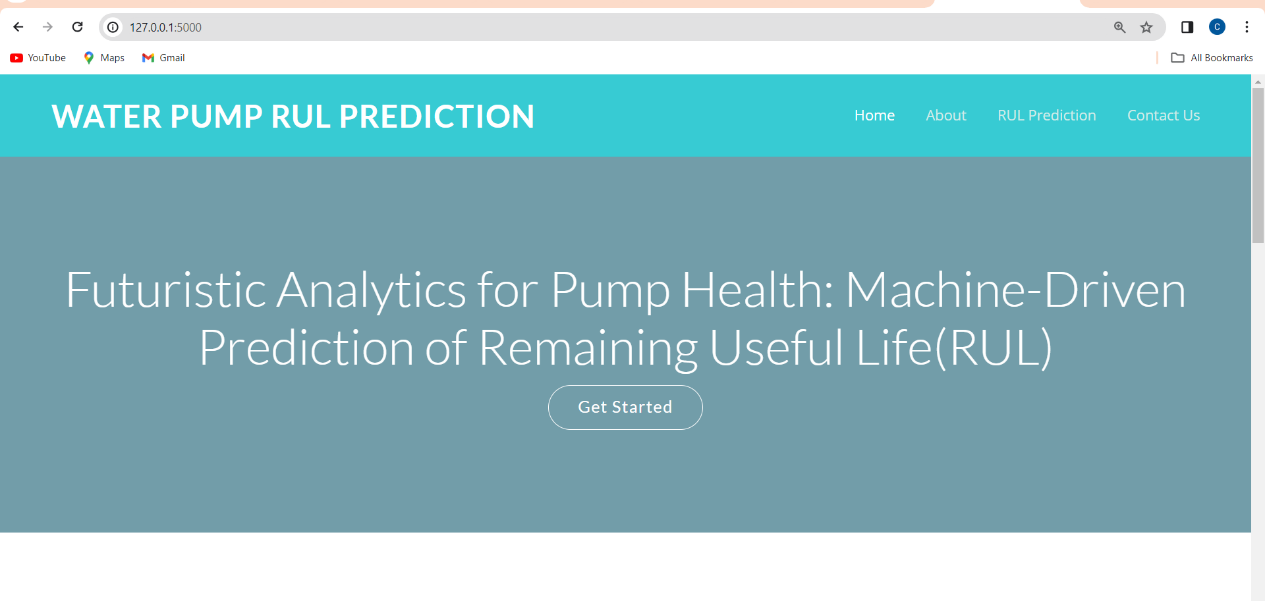
Now type “python app.py” command

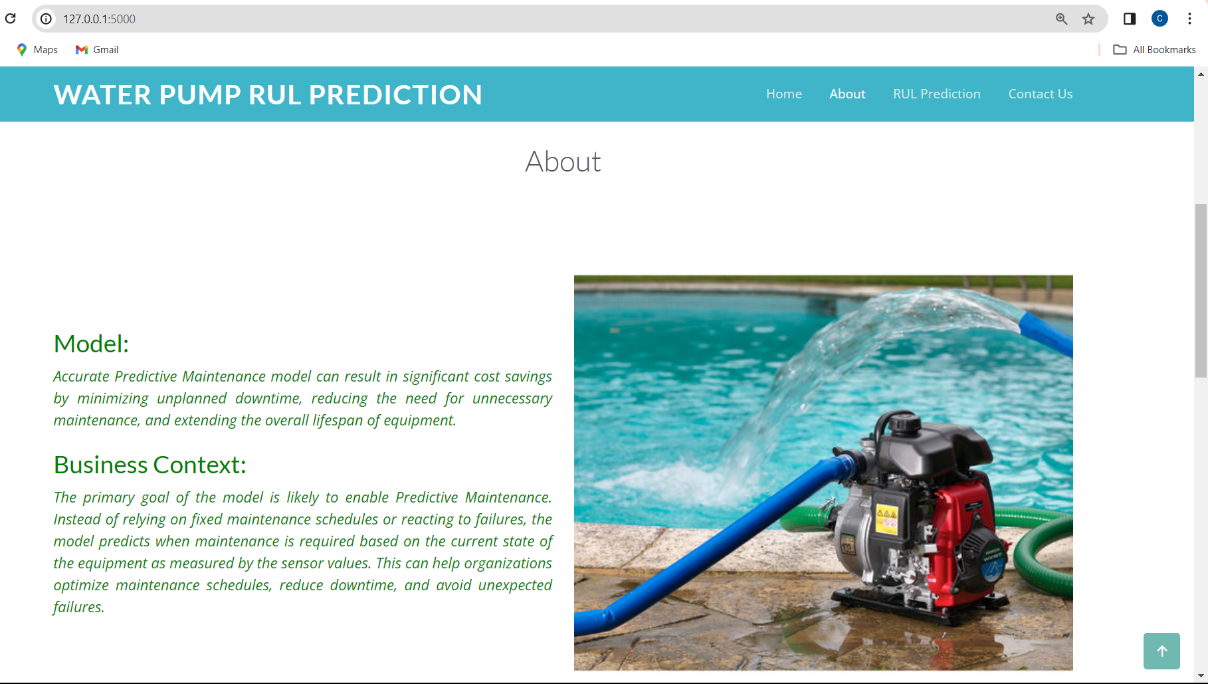
Navigate to the localhost where you can view your web page.

Click on the predict button from the top left corner, enter the inputs, click on the submit button, and see the result/prediction on the web.



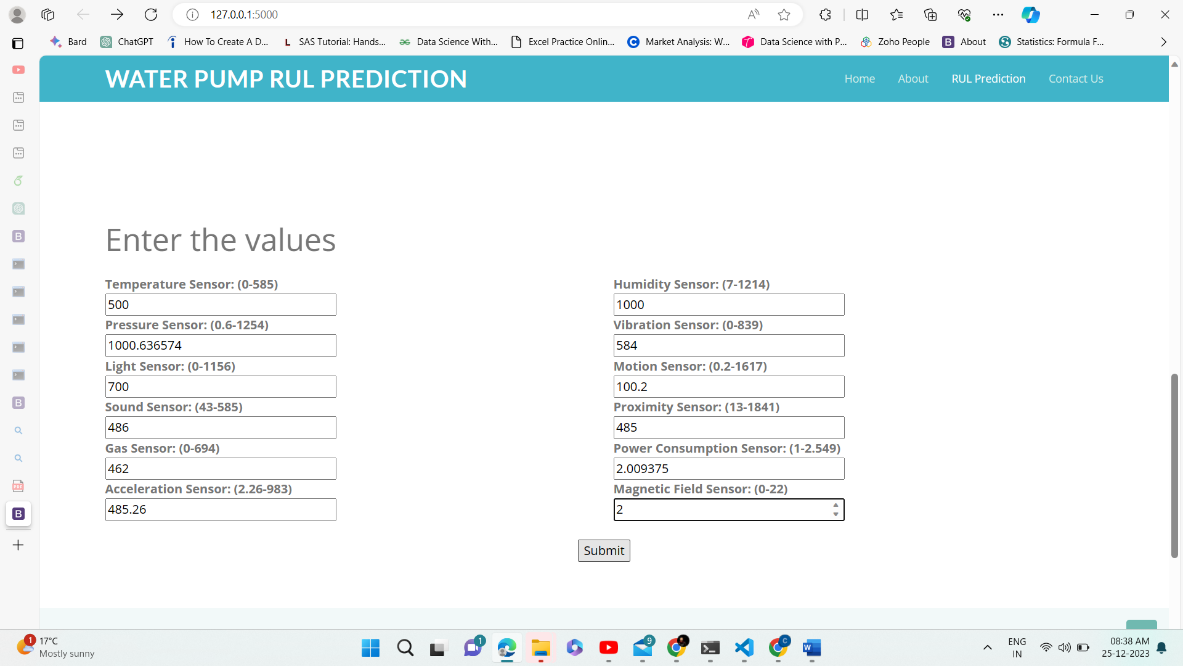
Now, Go to the web browser and write the localhost url (http://127.0.0.1:5000) to get the below result





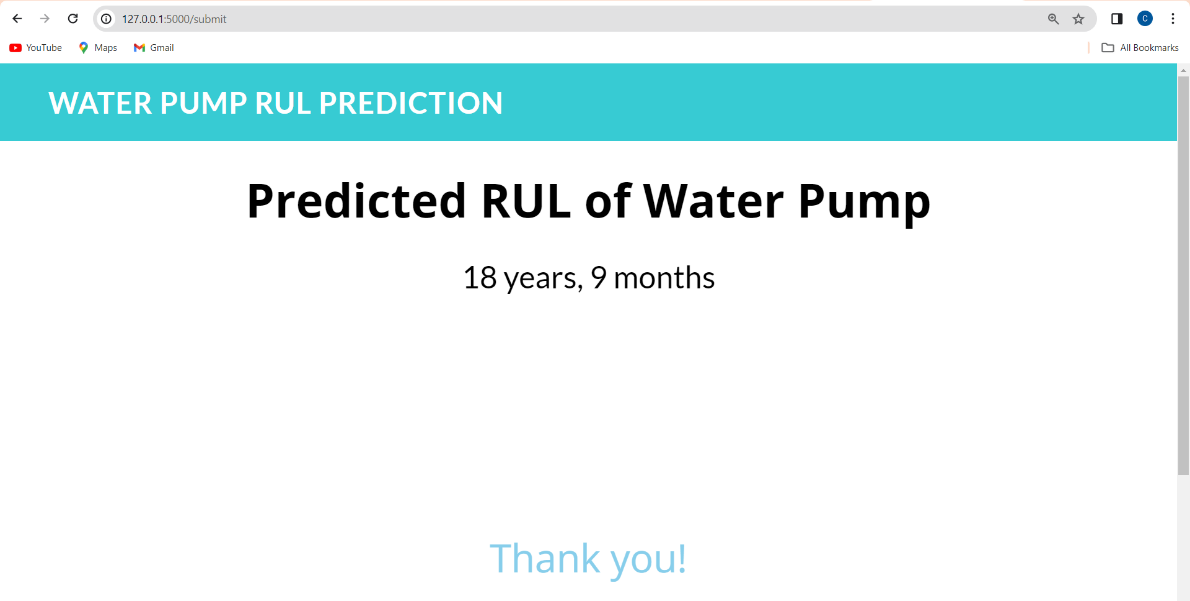
Now when you click on the ‘RUL Prediction’ button from the top right corner you will get redirected to the prediction page

**Input 1:**

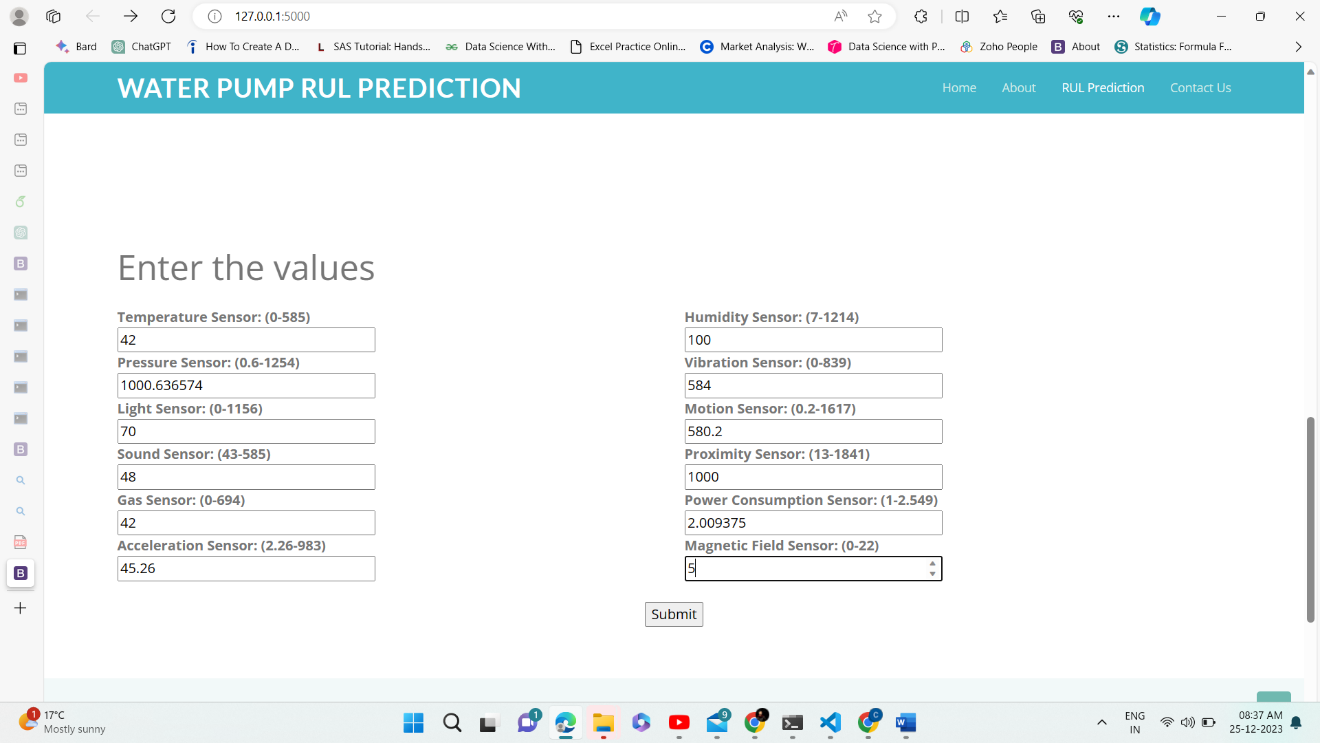


Lets look at what our prediction file looks like when you click the ‘RUL prediction’ button in the right corner below you will get redirected to the Enter values page After inputting the values and clicking on the submit button you will get output. We will see 3 Inputs and Outputs.

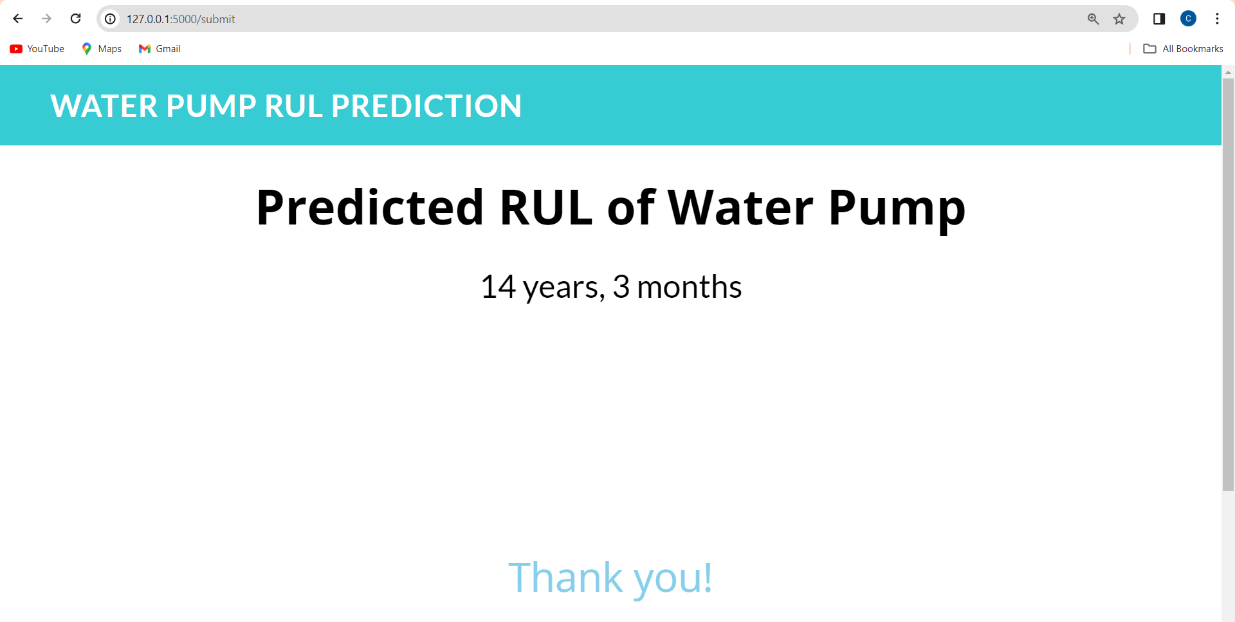
**Output 1:**

****

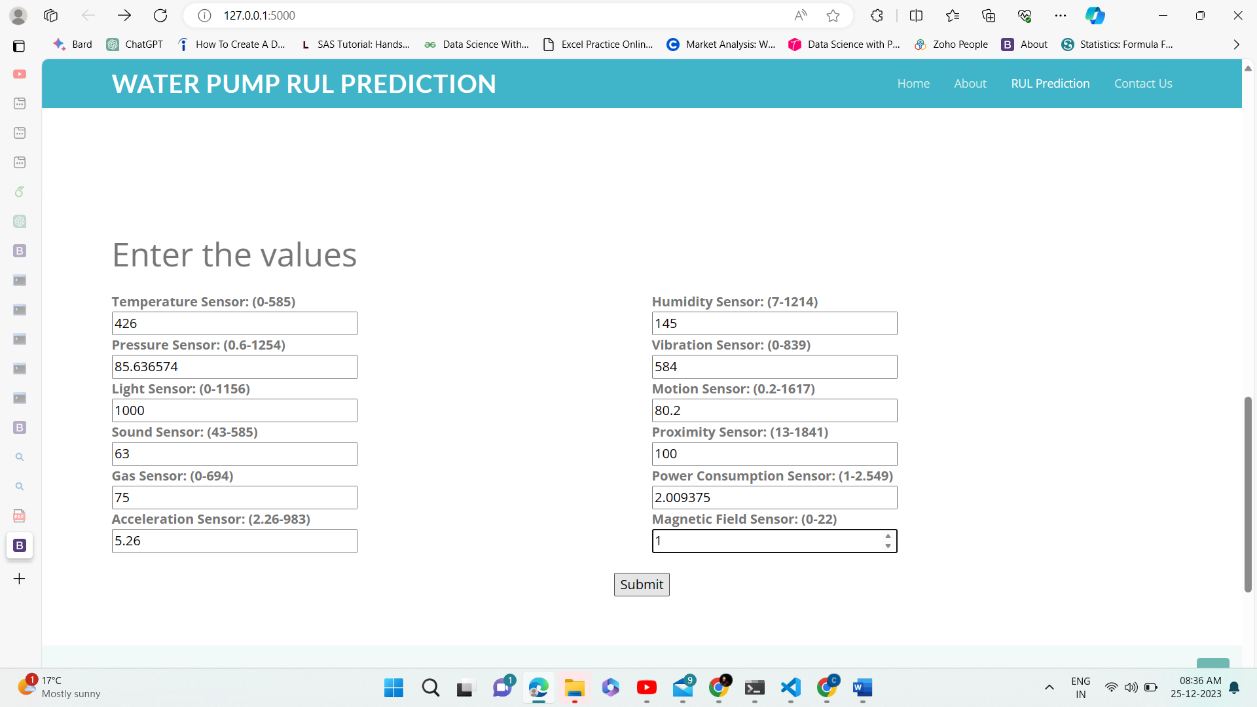
**Input 2:**

****

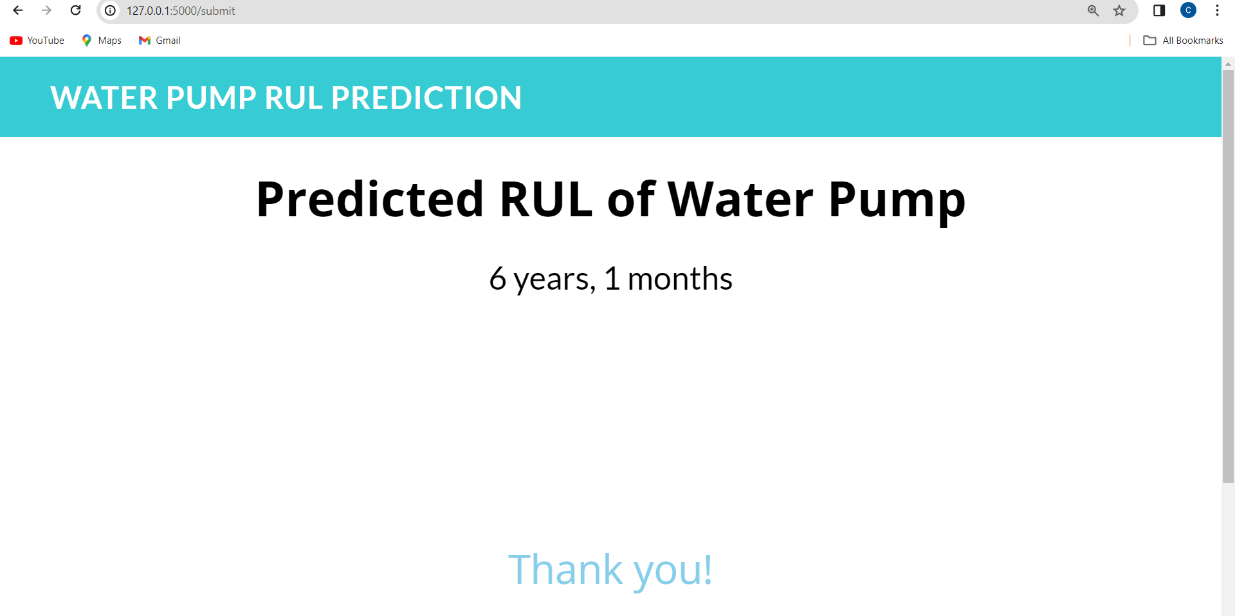
**Output 2:**



**Input 3:**



**Output 3:**

****

**Milestone 7: Project Demonstration & Documentation**

Below mentioned deliverables are to be submitted along with other deliverables

## Activity 1:- Record an explanation Video for the project end to end solution

**Activity 2:- Project Documentation-Step by step project development procedure**

Create a document as per the template provided