**Table Of Content**

**Title**   **Page No**

**CHAPTER 1 INTRODUCTION…………………………………..............................2**

**1.1 Project Overview………………..........................................…….…….......2**

**1.2 Objectives……………………………………………..................................2**

**1.3 Scope..............................................................................................................2**

**1.5 Tools and Technology used …………….....................................................3**

**CHAPTER 2 LITERATURE Review………………………….....…..........................4**

**2.1 Background ………………………………..................................................4**

**2.2 State of Art ……………..…………………………….................................4**

**2.3 Related Work …...........................................................................................4**

**CHAPTER 3 Methodology………………………........................................................5**

**3.1 Data Collection ………………………….....................................................5**

**3.2 Data Preprocessing……………………………….......................................6**

**3.3 Feature Extraction………………………………........................................6**

**3.4 Model Selection.............................................................................................7**

**3.5 Model Training..............................................................................................9**

**3.6 Model Deployment........................................................................................11**

**CHAPTER 4 Results and Evaluation………………………........................................14**

**4.1 Evaluation Matrices……………………………………..............................14**

**4.2 Result Analysis………………………………...............................................15**

**4.3 Performance Evaluation…………………………........................................16**

**CHAPTER 5 Conclusion And Future Work……………………………….................17**

**5.1 Summary…………………………….............................................................17**

**5.2 Future Work……………………....................................................................17**

**CHAPTER 6 BIBLIOGRAPHY……………………………………………………......19**

**CHAPTER 1: INTRODUCTION**

The project aims to create a predictive maintenance model that can forecast when a machine is likely to fail based on sensors data. Predictive maintenance uses volt, pressure, rotation, and vibration data from sensors to assess and forecast failures before they happen. Predictive maintenance helps us to avoid reactive maintenance, without incurring costs associated with doing too much preventive maintenance. The industrial use case of this model is to decrease planned maintenance, decrease service cost and to increase the assets availability and lifetime.

**1.1 Project Overview:**

Predictive maintenance technique uses data analysis tools and techniques to detect the possible defects in equipment predict or detect the failures system uses the machine learning algorithms perform task. The goal of this project is to implement a predictive maintenance system for a company's industrial equipment.

**1.2 Objectives:**

* Collect and analyze data from industrial equipment sensors.
* Develop machine learning models to predict equipment failures based on sensor data.
* Integrate the predictive maintenance system with existing equipment monitoring systems.
* To increase the assets availability and lifetime.
* Evaluate the system's effectiveness in reducing downtime and maintenance costs.

**1.3 Scope:**

The scope of a predictive maintenance project typically involves using data analytics and machine learning algorithms to monitor and analyze equipment performance to predict when maintenance is needed. This type of project can be applied to a wide range of industries, such as manufacturing, transportation, energy, and healthcare. The project's specific scope will depend on the type of equipment being monitored, the data sources available, and the desired outcomes.

**1.4 Tools and Technology used:**

Python, along with several popular libraries such as Pandas, NumPy, Matplot lib, Scikit-learn (sklearn), Seaborn, and Flask, is widely used in predictive maintenance projects due to its flexibility, ease of use, and powerful data analysis capabilities.

Pandas: Pandas is a powerful library for data manipulation and analysis. It provides tools for data cleaning, transformation, and analysis, which are essential for preprocessing data in predictive maintenance projects.

NumPy: NumPy is a library for scientific computing with Python. It provides support for large, multi-dimensional arrays and matrices, as well as an enormous collection of mathematical functions that are used extensively in predictive maintenance projects.

Matplotlib: Matplotlib is a library for data visualization in Python. It provides tools for creating charts, graphs, and other visualizations used to analyze and present data in predictive maintenance projects.

Scikit-learn (sklearn): Scikit-learn is a popular library for machine learning in Python. It provides algorithms for classification, regression, clustering, and other tasks commonly used in predictive maintenance projects.

Seaborn: Seaborn is a Python library for statistical data visualization. It provides high-level interfaces for creating informative and attractive statistical graphics that are useful for visualizing relationships between variables in predictive maintenance projects.

Flask: Flask is a lightweight web application framework for Python. It is commonly used to build RESTful APIs (Application Programming Interface) used to expose predictive models to other applications or systems.

Together, these tools and technologies provide a powerful set of capabilities for collecting, processing, analyzing, and presenting data in predictive maintenance projects.

**Chapter 2: LITERATURE REVIEW**

**2.1 Background:**

The goal of predictive maintenance, a use of machine learning technology, is to anticipate machinery failures and take preventative maintenance measures. The first time NASA used machine learning to predict the failure of the Space Shuttle Main Engine was in the 1990s, so the field of predictive maintenance has a lengthy history. Predictive maintenance has since developed into a significant area of research and development with numerous uses in sectors like manufacturing, healthcare, and transportation.

**2.2 State Of Art:**

advanced machine learning algorithms, like deep learning and neural networks, are used to analyze huge datasets and find complex patterns that traditional statistical models are unable to. This is the state-of-the-art in predictive maintenance using ML technology. Unsupervised learning algorithms for anomaly detection, transfer learning for improved model generalization, and the incorporation of real-time sensor data for predictive maintenance are some recent trends in study and development.

Our project aims to advance the cutting edge of predictive maintenance by developing a proof-of-concept system that uses machine learning algorithms to accurately predict equipment failures. We plan to use a combination of supervised and unsupervised learning techniques to analyze telemetry data and maintenance records, identify potential problems, and generate insights that can be used to optimize maintenance schedules and prevent equipment failures.

**2.3 Related Work:**

Numerous strategies and procedures have been put forth, and there has been a lot of research and development in the area of predictive maintenance. Li et al.'s (2020) proposal of a deep learning approach for the predictive maintenance of industrial equipment is one noteworthy piece of work. In the study, sensor data was used to train a deep neural network that could accurately anticipate device failures. Chen et al.'s (2021) proposal of a transfer learning approach for the predictive maintenance of wind turbines is another important piece of work. To analyze vibration signals and forecast devices, the study used a pre-trained deep neural network.

**CHAPTER 3: METHODOLOGY**

**3.1 Data Collection:**

The goal of this project is to build a predictive maintenance model for industrial equipment. To achieve this, we will be using a dataset from Kaggle that contains historical sensor data for various industrial machines. The dataset was collected over a period of time and includes a variety of variables that could impact the health of the equipment.

The dataset was obtained from Kaggle's Microsoft Azure Predictive Maintenance. It contains data from multiple sensors installed on 100 machines. The data is provided in CSV format and consists of five files:

Sources link:

<https://azuremlsampleexperiments.blob.core.windows.net/datasets/PdM_telemetry.csv><https://azuremlsampleexperiments.blob.core.windows.net/datasets/PdM_errors.csv><https://azuremlsampleexperiments.blob.core.windows.net/datasets/PdM_maint.csv><https://azuremlsampleexperiments.blob.core.windows.net/datasets/PdM_failures.csv><https://azuremlsampleexperiments.blob.core.windows.net/datasets/PdM_machines.csv>

**Telemetry Time Series Data (PdM\_telemetry.csv)**: It consists of hourly average of voltage, rotation, pressure, vibration collected from 100 machines for the year 2015.

**Error (PdM\_errors.csv)**: These are errors encountered by the machines while in operating condition. Since these errors don't shut down the machines, these are not considered as failures. The error date and times are rounded to the closest hour since the telemetry data is collected at an hourly rate.

**Maintenance (PdM\_maint.csv)**: If a component of a machine is replaced, that is captured as a record in this table. Components are replaced under two situations:

1. During the regularly scheduled visit, the technician replaced it (Proactive Maintenance)

2. A component breaks down and then the technician does unscheduled maintenance to replace the component (Reactive Maintenance). This is considered a failure and corresponding data is captured under Failures. Maintenance data has both 2014 and 2015 records. This data is rounded to the closest hour since the telemetry data is collected at an hourly rate.

**Failures (PdM\_failures.csv)**: Each record represents replacement of a component due to failure. This data is a subset of Maintenance data. This data is rounded to the closest hour since the telemetry data is collected at an hourly rate.

**Metadata of Machines (PdM\_Machines.csv)**: Model type & age of the Machines.

**3.2 Data Preprocessing:**

In our project, there are a total of five csv files that provide information on the machine and its failure, routine maintenance information, fault component, and the times that events occur. First, we load all the csv files and check for null values, but none of them have any.

After that, we split the datetime column into date and time and stored them in two separate columns. Then we divided the date column into three separate columns for date, month, and year, and finally separated the time column into just the hour because the data was collected on an hourly basis. After that we drop the original column of datetime.

We combine all five files into a single csv file that contains all the machine data. We use telemetry data as a base file for merging and combining error, maintenance, and failure data based on the year, month, and day as well as the machine id. We also combine machine data using the machine id.

**3.3 Feature Extraction:**

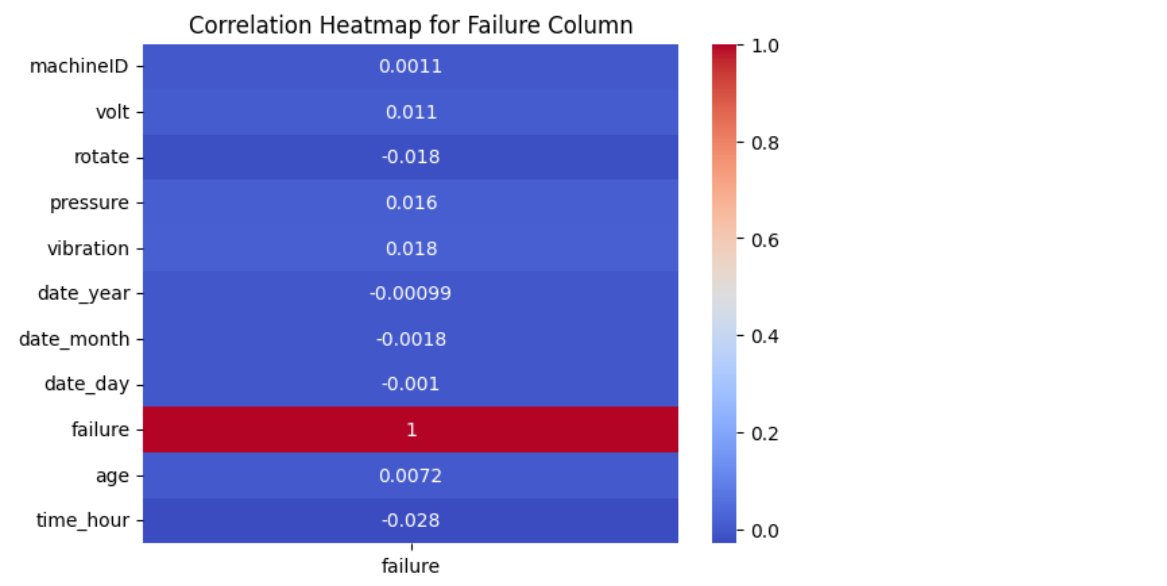
After merging, we checked for null values again, and as a result, 99% of the error type, component maintenance, and failure component values are now empty. Therefore, we choose to handle null values only for the failure component because the maintenance component's data stores the comp type that is replaced while performing regular maintenance, which has no impact on the machine's productivity, and error only contains data about errors that don't stop the machine from working. So, we dropped that two column.

We now have a very unbalanced set of data to deal with because there is only one target column with component failure values that are less than 1% failure and only four categories of component failure. Therefore, we treat all four types of component failure as failures and mark as 1 and treat null values as not failures mark as 0.

Since the data set has been cleaned up, we no longer have to deal with any null values.

In order to better fully understand, we also create a graph when checking the association of all the attributes with respect to the failure column during feature selection.





According to the results, there is roughly a minor connection between any of the attributes and failure rate. Rotation has a negative correlation with failure, while voltage, pressure, and vibration are positively correlated.

**3.4 Model Selection:**

For a predictive maintenance project with a binary classification problem, where the objective is to predict whether a machine is likely to fail or not, there are several machine learning models that can be used. These include:

1. Logistic Regression: Logistic Regression is a statistical learning algorithm that can be used for binary classification in predictive maintenance. In logistic regression, the algorithm models the probability of an event occurring using a logistic function, and then assigns the label based on a decision threshold. Logistic Regression is computationally efficient, interpretable, and can handle both numerical and categorical features. However, Logistic Regression assumes a linear relationship between the input features and the target variable and can be sensitive to outliers and multicollinearity.
2. Decision Trees: Decision Trees are a popular machine learning algorithm that can be used for binary classification in predictive maintenance. In decision trees, the algorithm recursively splits the data based on the features that best separate the two classes, creating a tree-like structure that can be easily interpreted. Decision trees are computationally efficient, can handle both numerical and categorical features, and can capture non-linear relationships between the input features and the target variable. However, decision trees are prone to overfitting, and small changes in the data can result in different trees being generated.
3. Random Forests: Random Forests is an ensemble learning algorithm that can be used for binary classification in predictive maintenance. In random forests, the algorithm builds multiple decision trees on randomly selected subsets of the data and features, and then combines the results to make a prediction. Random Forests can reduce overfitting, handle missing data, and capture non-linear relationships between the input features and the target variable. However, Random Forests can be computationally expensive, and can be less interpretable compared to single decision trees.
4. K-Nearest Neighbors (KNN): KNN is a non-parametric and lazy learning algorithm that can be used for binary classification in predictive maintenance. In KNN, the algorithm classifies a new data point by finding the k nearest neighbors in the training set based on a similarity metric such as Euclidean distance, and then assigns the label based on the majority vote of the k nearest neighbors. KNN is easy to implement and can be used for both small and large datasets. However, KNN can be sensitive to the choice of k and can be computationally expensive, especially for large datasets.
5. Support Vector Machines (SVM): SVM is a supervised learning algorithm that can be used for binary classification in predictive maintenance. In SVM, the algorithm tries to find the hyperplane that maximizes the margin between the two classes. SVM can work well with high-dimensional data and is less prone to overfitting compared to other classification models. SVM can also handle non-linear decision boundaries through the use of kernels such as polynomial and radial basis function (RBF) kernels. However, SVM can be sensitive to the choice of hyperparameters and can be computationally expensive for large datasets.

Random forest is a popular algorithm used for classification problems in machine learning. It is an ensemble learning method that combines multiple decision trees to make more accurate predictions. Random forest classifier has several advantages over other models, which could be the reason why it performed better in your predictive maintenance project. Here are some possible reasons:

Random forest can effectively handle non-linear relationships between the input features and the output variable. This is particularly useful for predictive maintenance projects where the relationship between equipment conditions and failure may not be a linear one.

Random forest is less sensitive to outliers than other models, such as logistic regression or SVM. This makes it more suitable for predictive maintenance projects where outliers in the data are common.

**3.5 Model Training:**

Training a model is important in a machine learning project because it is the process by which the model is taught to recognize patterns and make predictions based on input data. In other words, the training process allows the model to learn from past examples so that it can make accurate predictions on new data.

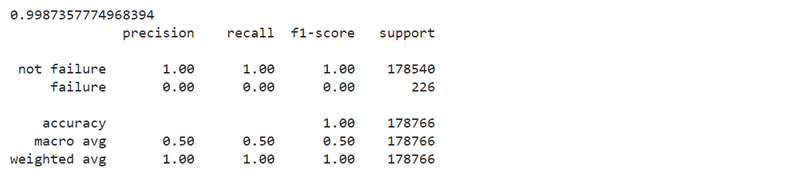
The training process involves providing the model with a large amount of labeled data, which means data that is already categorized or labeled with the correct output. The model is then trained to recognize patterns in the input data that correspond to the correct output labels. This is done through a process of adjusting the model's parameters until it produces the most accurate predictions possible.

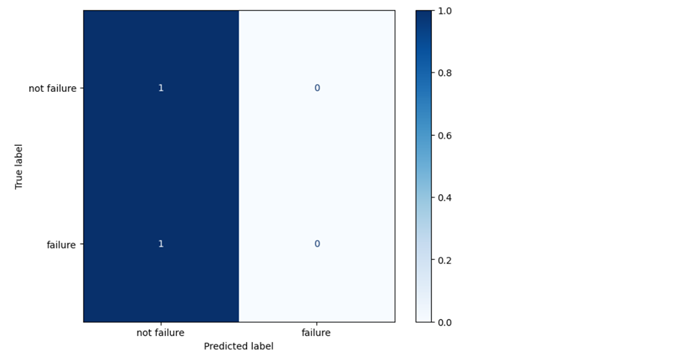
The importance of training a model lies in its ability to generalize to new, unseen data. Once a model has been trained, it can be used to make predictions on new data that it has never seen before. The more accurately the model has been trained, the better it will perform on new data. However, overfitting can also occur if a model is too heavily trained on a specific dataset, which can result in poor performance on new data.

In summary, training a model is a crucial step in machine learning because it enables the model to learn from labeled data and make accurate predictions on new, unseen data. The quality of the training data, the model architecture, and the training process itself are all important factors that can impact the performance of the trained model.

In order to apply the model, we first divided the data into the train and test parts. In order to have an impact on the target variable, we normalized the data first such that all of the attributes follow a normal distribution.

After applying the random forest model, we find that the accuracy, precision, recall, and f1-score are almost 99% accurate for predicting the chances of not failure happening, whereas the model performs poorly and receives lower recall and f1-score when predicting failure.

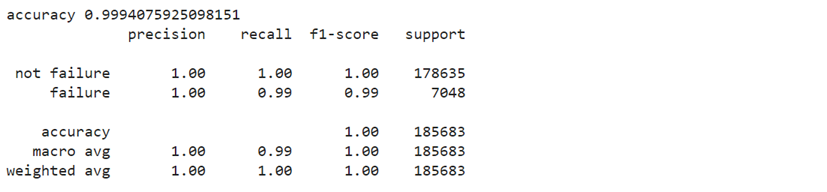


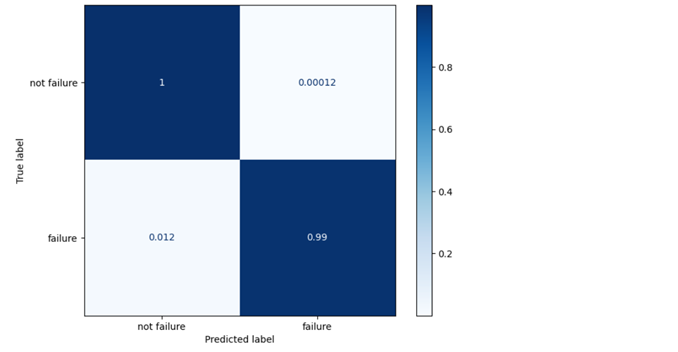


In a predictive maintenance project, it is common to encounter an imbalance in the data, where one class of outcomes may be overrepresented compared to another. This can lead to biased results and poor predictive performance as we can see in our model.

To address this issue, one technique that can be used is SMOTE (Synthetic Minority Over-sampling Technique). This is an oversampling method that generates synthetic samples of the minority class by interpolating between existing minority class samples. This helps to balance the classes in the dataset and improve the performance of machine learning algorithms trained on the sampled data.

By using SMOTE, we achieved more representative dataset that can lead to better predictive performance and enable more accurate predictions of component.





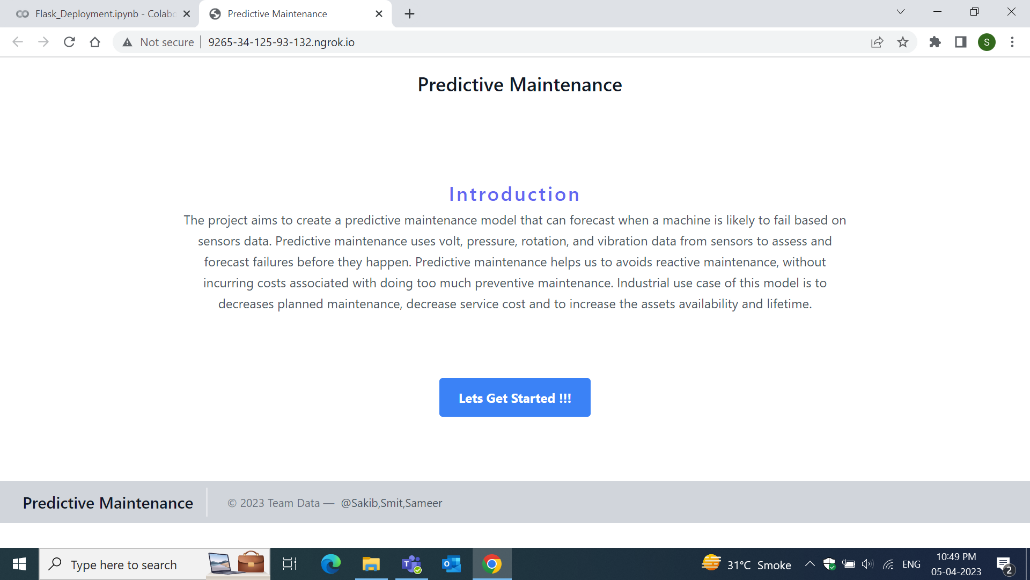
**3.6 Model Deployment:**

Steps to deploy project using Flask and Ngrok:

1. Build and train the ML model using Python and ML libraries like Scikit-learn.
2. Use Flask to create a RESTful API that can receive data and return predictions.
3. Install Ngrok and use it to create a secure tunnel to the locally running Flask app. This allows you to expose the Flask app to the internet and receive requests from external clients.

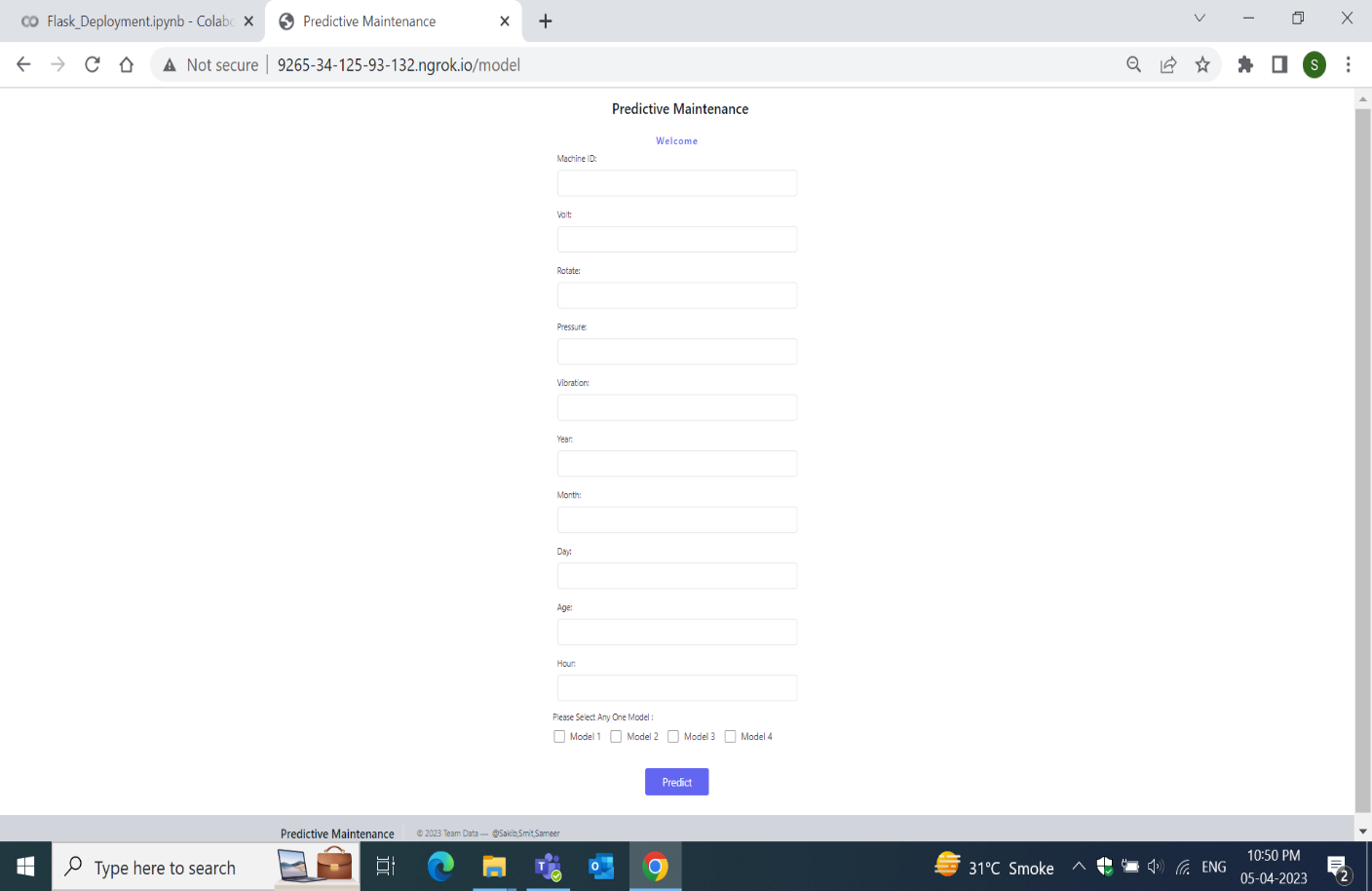
###### **Home page:**

This is the home page of predicative maintenance project with contains the introduction about the project and aims of the project.



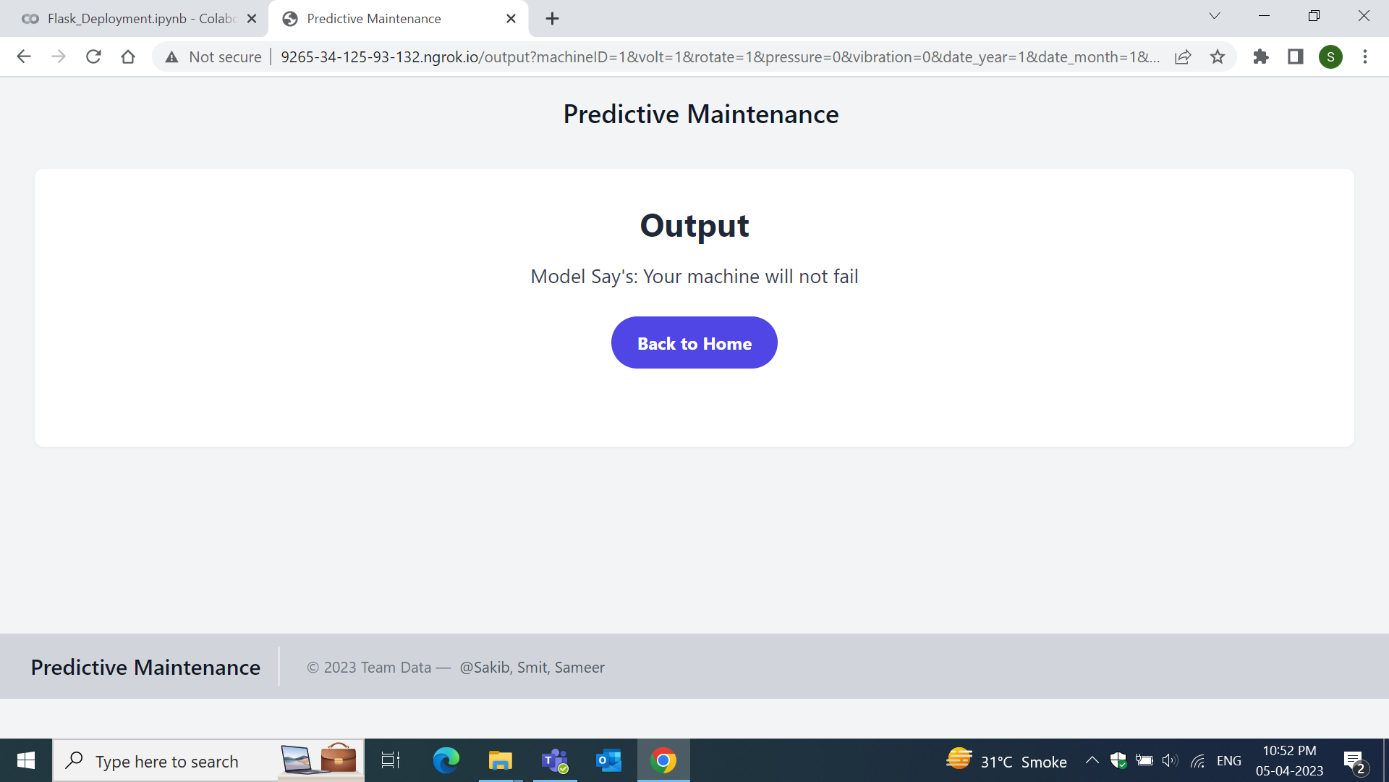
###### **Model page:**

Here customer have to input data into the input fileds and then user have to select model type of the machine and then have to press the predict button.

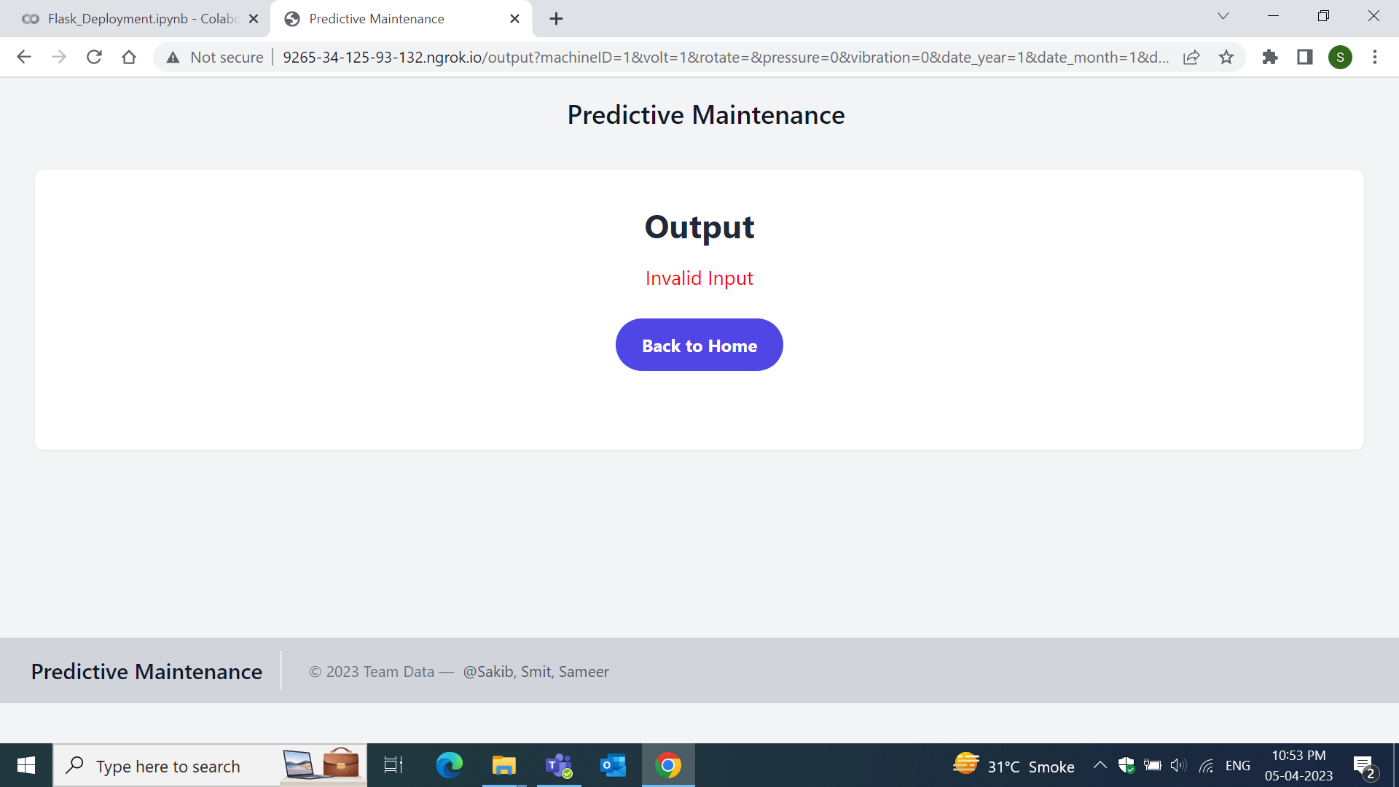


###### **Output page:**

It show output like this when it will not matched the condition to fail.



It will show the output like this when user leave any input as blank.



**CHAPTER 4 : Results and Evaluation**

**4.1 Evaluation Metrics:**

A machine learning model's performance is evaluated using evaluation metrics. These metrics provide insight into the model's performance, accuracy, and ability to accurately anticipate outcomes based on the input data.

Confusion matrix:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Actual values | | |
| Predicted values |  | Positive | Negative |
| Positive | TP | FP |
| Negative | FN | TN |

TP (True positive): This parameter indicated that after modeling how many predicated positive values are actual positive

FP (False positive): This parameter indicated that after modeling how many predicated positive values are actual negative values

TN (True negative): This parameter indicated that after modeling how many predicated negative values are actual negative

FP (False negative): This parameter indicated that after modeling how many predicated negative values are actual positive values

Accuracy: This represents the number of correctly classified data over total number of data. Accuracy is not much important metrics when dataset is unbalanced

Accuracy = TP+TN / TP+TN+FP+FN

Precision: It represents correctly predicted positive values over total predicted positive values

Precision= TP / TP+FP

Recall: It represents correctly predicted positive values over total actual positive values

Recall= TP / TP+FN

F1 score: It is harmonic mean of precision and recall

Based on Accuracy, precision, recall, f1 score we decide which model is performing better.

**4.2 Result Analysis:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 score |
| KNN | 0.997 | 0.927 | 0.990 | 0.957 |
| SVM | 0.981 | 1.00 | 0.508 | 0.674 |
| Random forest | 0.999 | 0.997 | 0.984 | 0.990 |
| Decision Tree classifier | 0.997 | 0.951 | 0.974 | 0.962 |
| Logistic regression | 0.981 | 0.992 | 0.511 | 0.675 |

The Random Forest model outperformed the other models in terms of accuracy, precision, recall, and F1 score, according to the results of the machine learning models. The accuracy of the model is 0.999, indicating that 99.9% of the data were properly categorized. The precision score is 0.997, meaning that 99.7% of all positive values that were predicted were in fact positive. With a recall score of 0.984, the model successfully detected 98.4% of the real positive values. Precision and recall's harmonic mean, or F1 score, is 0.990.

The accuracy of the KNN and Decision Tree Classifier models, which were both successful, was 0.996 and 0.997, respectively. In terms of precision, recall, and F1 score, they underperformed the Random Forest model.

The weakest performers were the SVM and Logistic Regression models, with F1 scores of only 0.674 and 0.673, respectively. Although the models' excellent accuracy, their low precision and recall scores show that they are poor at correctly identifying positive values.

**4.3 Performance Evaluation:**

The Random Forest model's F1 score, which shows how well it properly identifies positive values, together with its high accuracy, precision, recall, and F1 score, are its strongest points. Its flaw is that, compared to some of the other models, such logistic regression, it could be more complicated and computationally expensive. Its drawbacks include the requirement for a sizable dataset and the possibility of overfitting if the number of trees is excessive.

High accuracy ratings were achieved by the KNN and Decision Tree Classifier models, which also performed well. With larger datasets, they might not perform as well as the Random Forest model, and they might also be more prone to overfitting.

With poor precision and recall score, the Logistic Regression and SVM models performed the poorest. For very unbalanced datasets, such as those with a dearth of positive examples, these models might not be appropriate. In contrast to some of the other models, they might be more computationally effective and simpler to understand.

Although the Random Forest model for preventive maintenance seems to be very accurate, other models might be more appropriate for particular issues.

**CHAPTER 5 : Conclusion and future work**

**5.1 Summary**

To find patterns, trends, and potential variables leading to machine failures, we used the Python programming language and different data analysis and visualization tools to analyze and visualize data linked to mistakes, failures, and machines.

The developed code retrieves data from multiple CSV files using Google Colab, a popular Python development environment, and addresses the issue of missing values in the data using various techniques such as dropping rows, filling with default values, filling with statistical measures, or using machine learning techniques. This ensures that the data analysis outputs are accurate and reliable.

Using the code, the project team performed exploratory data analysis (EDA), time series analysis, and machine failure analysis, yielding important insights into machine performance and maintenance routines. The data analysis and visualization findings were used to inform decision-making in the PdM project, such as optimizing maintenance schedules, identifying potential areas for improvement in maintenance strategies, and developing preventive maintenance strategies to minimize machine failures and downtime.

The project report includes a comprehensive description of the methodology employed, data analysis and visualization tools used, and outcomes gained. The study also addresses the findings' significance for the PdM project and makes recommendations for further research and development. Overall, the research has advanced predictive maintenance techniques in industry and provided useful insights for improving the efficiency and efficacy of machine maintenance practices.

**5.2 Future Work:**

The potential future work and research directions for improving the project could include the following:

* Incorporation of Advanced Machine Learning Techniques: The project could explore the use of advanced machine learning techniques such as deep learning algorithms, reinforcement learning, or ensemble methods to improve the accuracy and predictive capabilities of the PdM solution. These advanced techniques may uncover hidden patterns and relationships in the data that traditional machine learning techniques might miss, leading to better predictions and more effective maintenance strategies.
* Integration of Real-time Data: The project could consider incorporating real-time data streams from sensors or other sources to enable real-time monitoring and prediction of machine failures. This would allow for proactive and timely maintenance interventions, minimizing downtime and improving overall machine performance.
* Integration of Internet of Things (IoT) Technologies: The project could explore the use of IoT technologies to collect data from machines in real-time, enabling remote monitoring and control of machines, and facilitating predictive maintenance. This could involve incorporating IoT devices, sensors, and connectivity solutions to capture and transmit machine data, leading to more accurate and timely maintenance predictions.
* Incorporation of Domain-specific Knowledge: The project could consider incorporating domain-specific knowledge or expertise from industry experts or domain specialists to enhance the accuracy and relevance of the predictive maintenance solution. This could involve integrating expert rules, heuristics, or domain-specific features into the machine learning algorithms to improve their performance and make the solution more customized for the specific industry or application.
* Expansion of Datasets: The project could explore the use of larger and more diverse datasets to train and validate the machine learning models. This could involve collecting additional data from multiple sources, including different types of machines, different operational conditions, or different maintenance strategies. A larger and more diverse dataset would enable the development of more robust and generalizable predictive maintenance models.
* In conclusion, the future work for improving the project could involve incorporating advanced machine learning techniques, integrating real-time data and IoT technologies, incorporating domain-specific knowledge, and expanding the datasets used for training and validation. These potential research directions could enhance the accuracy, timeliness, and relevance of the predictive maintenance solution and contribute to the advancement of the field of predictive maintenance in the specific application or industry.

**CHAPTER 6: Bibliography:**

1. The Predictive Maintenance Blog: https:/[/www.predictivemaint](http://www.predictivemaintenanceblog.com/)e[nanceblog.com/](http://www.predictivemaintenanceblog.com/)

The Predictive Maintenance Blog offers a wealth of information on predictive maintenance, including best practices, case studies, and industry news. The blog is updated regularly and covers topics such as data analytics, machine learning, and artificial intelligence.

1. IoT Analytics: <https://iot-analytics.com/>

IoT Analytics is a research and consulting firm that specializes in the Internet of Things (IoT) and predictive maintenance. The website offers a wealth of information on IoT trends, market research, and case studies.