Exploring Machine Learning Algorithms for Predictive Maintenance in Manufacturing Industries
Configuration Manual

MSc Research Project
Data Analytics

Suprith Shiva Boraiah
StudentID:x22166785

School of Computing
National College of Ireland

Supervisor: Arghir Nicolae Moldovan

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Suprith Shiva Boraiah
Student ID:	x22166785
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Arghir Nicolae Moldovan
Submission Due Date:	14/12/2023
Project Title:	Exploring Machine Learning Algorithms for Predictive Maintenance in Manufacturing Industries Configuration Manual
Word Count:	1054
Page Count:	7

I hereby certify that the information contained in this (my submission) is information about research I conducted for this project. All information other than my contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use another author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Suprith Shiva Boraiah
Date:	14th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Exploring Machine Learning Algorithms for Predictive Maintenance in Manufacturing Industries
Configuration Manual

Suprith Shiva Boraiah
x22166785

1 Introduction

The setup guide offers comprehensive instructions for configuring systems or devices. Its main objective is to provide a detailed outline of procedures relevant to conducting a research study. Furthermore, it explicitly outlines the essential machine configuration needed for building and executing models. The steps outlined cover both the minimal setup required for project success and the installation of essential applications and packages. Serving as an essential guide, the manual clarifies the complexities of the setup process to ensure smooth navigation and implementation.

2 Project Files Detail

1. machine_failure_data (D1)

- File Name: D1
- Description: This dataset contains information related to machine failures.
- Link:(https://www.kaggle.com/datasets/binaicrai/machine-failure-data)

2. ai4i2020 (D2)

- File Name: D2
- Description: This dataset is related to predictive maintenance and is from the AI4I 2020 competition.
- Link:(https://www.kaggle.com/datasets/stephanmatzka/predictive-maintenance-dataset-ai4i-2020)

3. binary classification of machine failures dataset (D3)

- File Name: D3
- Description: This dataset is focused on binary classification of machine failures.
- Link:(https://www.kaggle.com/datasets/nileshthonte/binary-classification-of-machine-failures-dataset/discussion)

4. Explainable AI (XAI) Drilling Dataset

- File Name: D4
- Description: This dataset is designed for Explainable AI (XAI) and specifically involves drilling data.
- Link:(https://www.kaggle.com/datasets/raphaelwallsberger/xai-drilling-dataset)

Tool Used: Jupyter Notebook
Purpose: Jupyter Notebook is employed for both modeling and evaluating the machine learning models.
This interactive computing environment allows for easy iteration and visualization, making it a popular choice for data science and machine learning tasks.

3 System Specification

A system specification refers to a written document that outlines the technical specifications and requirements of a system. This document typically includes details about the system's components, functionality, design, and other technical features. In Figure 1, you can observe the system configuration employed for executing this project. Nevertheless, the recommended configuration is designed to guarantee the smooth execution of the code without encountering any issues.

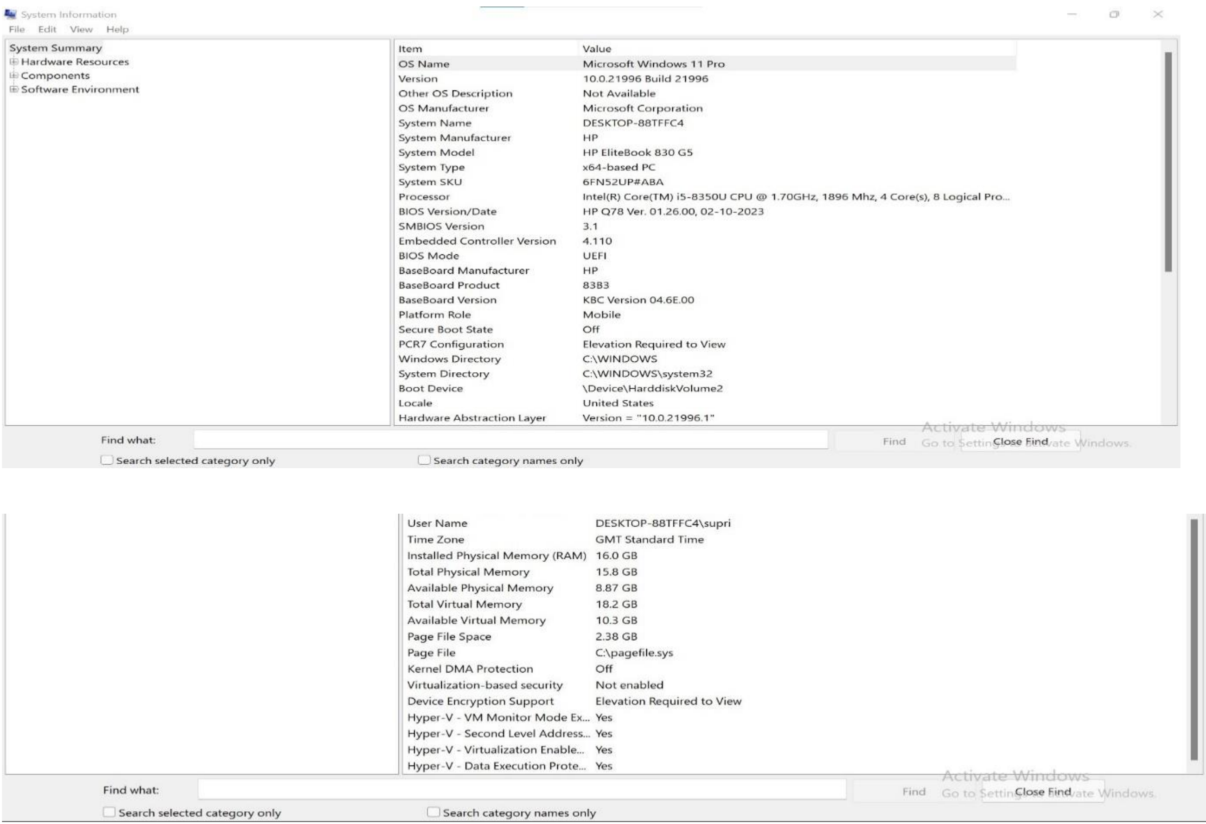

Figure 1: System Specification

4 Software Used

• Microsoft Excel: Utilized for the initial phase of exploration.

• Jupyter Notebook: Utilized specifically for the modeling and evaluation processes.

5 Download and Install

Before diving into Python coding, ensure that Python is installed on your system, preferably the latest version. In this instance, Python 3.10.2 for Windows 11 was downloaded and installed. Once Python is set up, you'll need a development environment to write, execute, and view code output. Jupyter Notebook stands out as a popular and user-friendly choice. It comes bundled with the Python distribution Anaconda, available for download based on your operating system.

Figure 2 illustrates Anaconda's dashboard, showcasing not only the Jupyter Notebook but also pre-installed packages. To initiate Python code development, the initial step is to launch Jupyter Notebook and create a new Python file within the platform. This establishes a conducive environment for coding and testing.

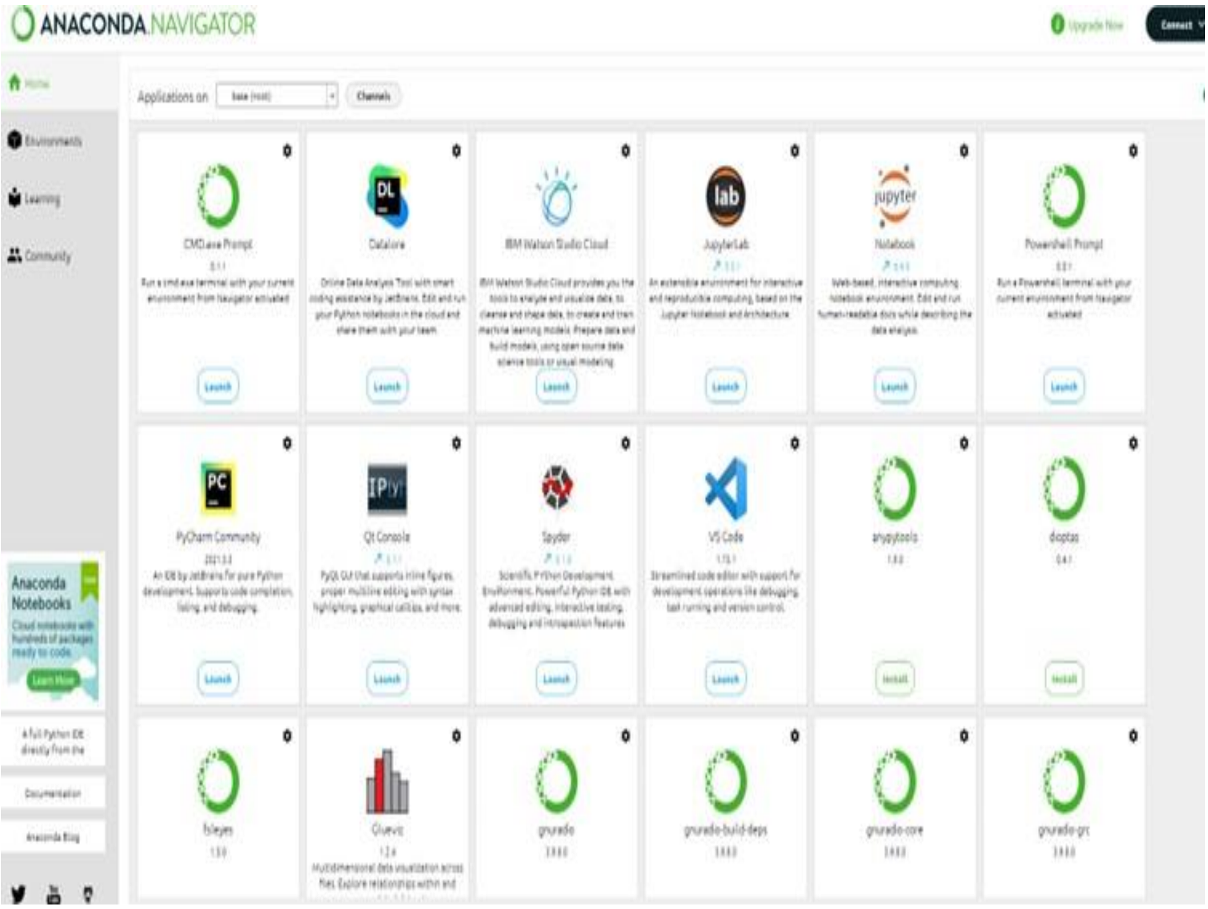

Figure 2: System Specification

6 Project Development

Upon completing the outlined steps, access the Jupyter Notebook. Initiate this process by selecting the "new" button situated at the upper section of the file open segment. Utilize the file reference specified in the code section to load the programmed file. Subsequently, you have the flexibility to execute all cells simultaneously or opt for individual cell execution. If a particular package requires installation, execute the command "pip install package-name" to address the requirement.

6.1 Importing Library

Figure 3 showcases the packages employed in this project. It's worth noting that the cloud platform comes pre-equipped with several essential libraries, all ready for use.

```
In [1]: #installing all the required packages for the project
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix, roc_curve, auc
from sklearn.model_selection import train_test_split, GridSearchCV
import matplotlib.pyplot as plt
from sklearn.preprocessing import label_binarize
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import roc_auc_score
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix, roc_curve, auc
from sklearn.preprocessing import label_binarize
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report, roc_curve, auc
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import GradientBoostingClassifier
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, roc_auc_score, auc

In [ ]:
```

Figure 3: Packages used in the Project.

6.2 Importing Files

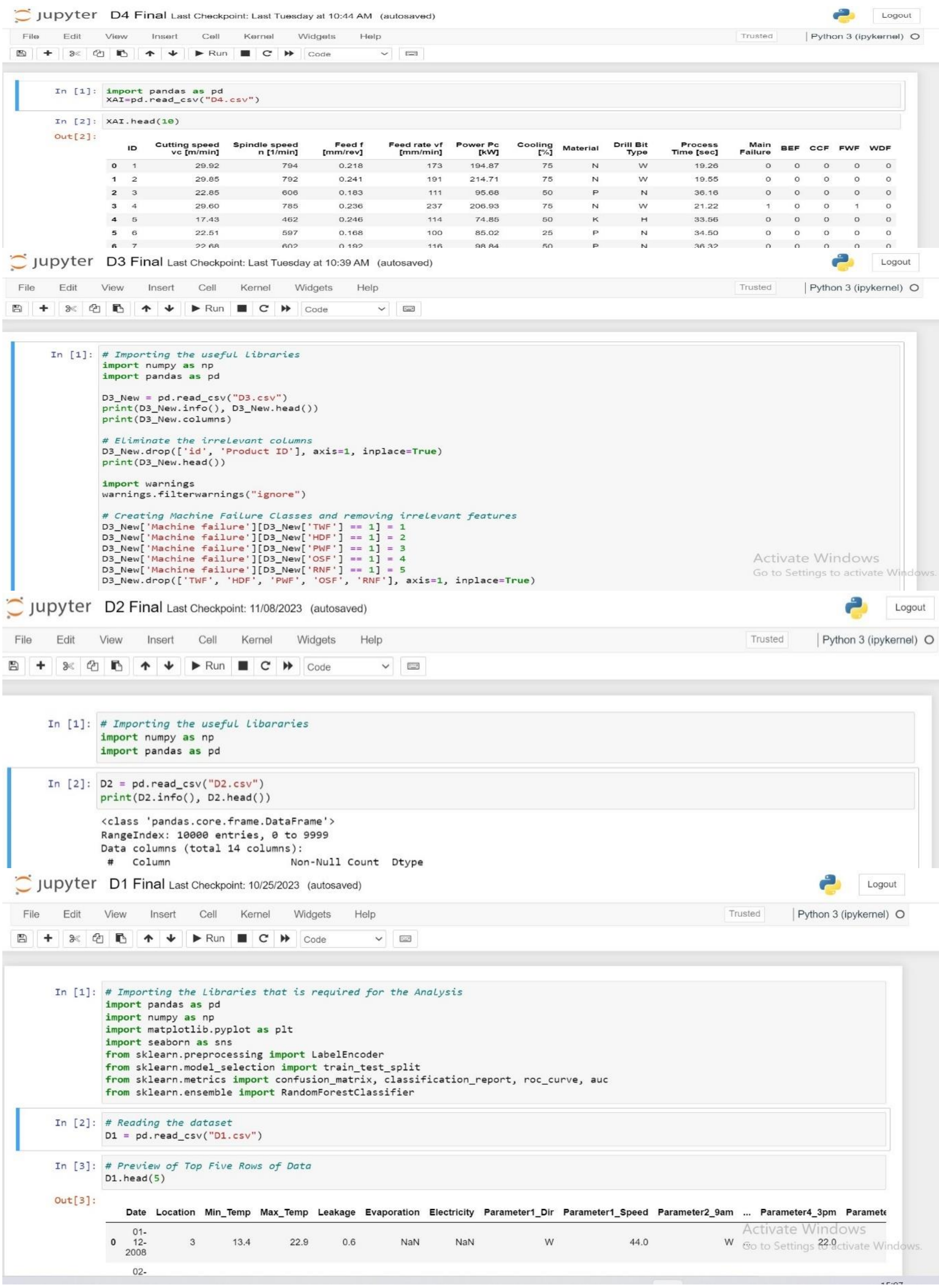

Figure 4: Procedure to Fetch the File in Jupyter Notebook

To begin, ensure that you initiate the process by uploading the CSV file within your Jupyter Notebook environment. Utilize the Jupyter interface for this task, a standard method widely employed for file uploads.

Subsequently, meticulously verify the file location to ensure the CSV file resides in the identical directory as your Jupyter Notebook. For streamlined accessibility, it is imperative to specify the correct path if the file is stored in a different location. This meticulous attention to file placement ensures seamless integration and retrieval within the project framework.

Subsequently, employ the Pandas library within your Jupyter Notebook cell to execute the operation of reading the CSV file into a Data Frame. This involves utilizing the prescribed code that is tailored for this purpose.

Following the execution of the code, proceed to run the entire Jupyter Notebook cell. This can be accomplished by either selecting the cell and clicking the "Run" button or utilizing the keyboard shortcut, which is typically Shift + Enter.

To validate the successful loading of data, inspect the initial rows of the Data Frame. Print these rows to the console or display them in the notebook to confirm that the data has been imported and integrated into the Data Frame as intended. This verification step ensures the accuracy and completeness of the data-loading process.

6.3 Processing

1. Missing Values:

- Imputed using KNN imputer.

2. Null Values Handling:

- Checked with IsNull () and sum () utilities.
- Duplicate check using the duplicated () function.

3. Outliers Treatment:

- Identified extreme outliers using box graphs.
- Removed outliers with minimal data loss.

4. Normalization:

- Utilized Box-Cox and Max Scalar techniques.
- Checked normalization graphs.

5. Feature Selection:

- Four steps: identification of constant features, Pearson correlation analysis, statistical analysis, and recursive feature deletion.

6. Libraries Used:

- Refer to Figure 3 for the libraries used for the above tasks.

6.4 Modeling

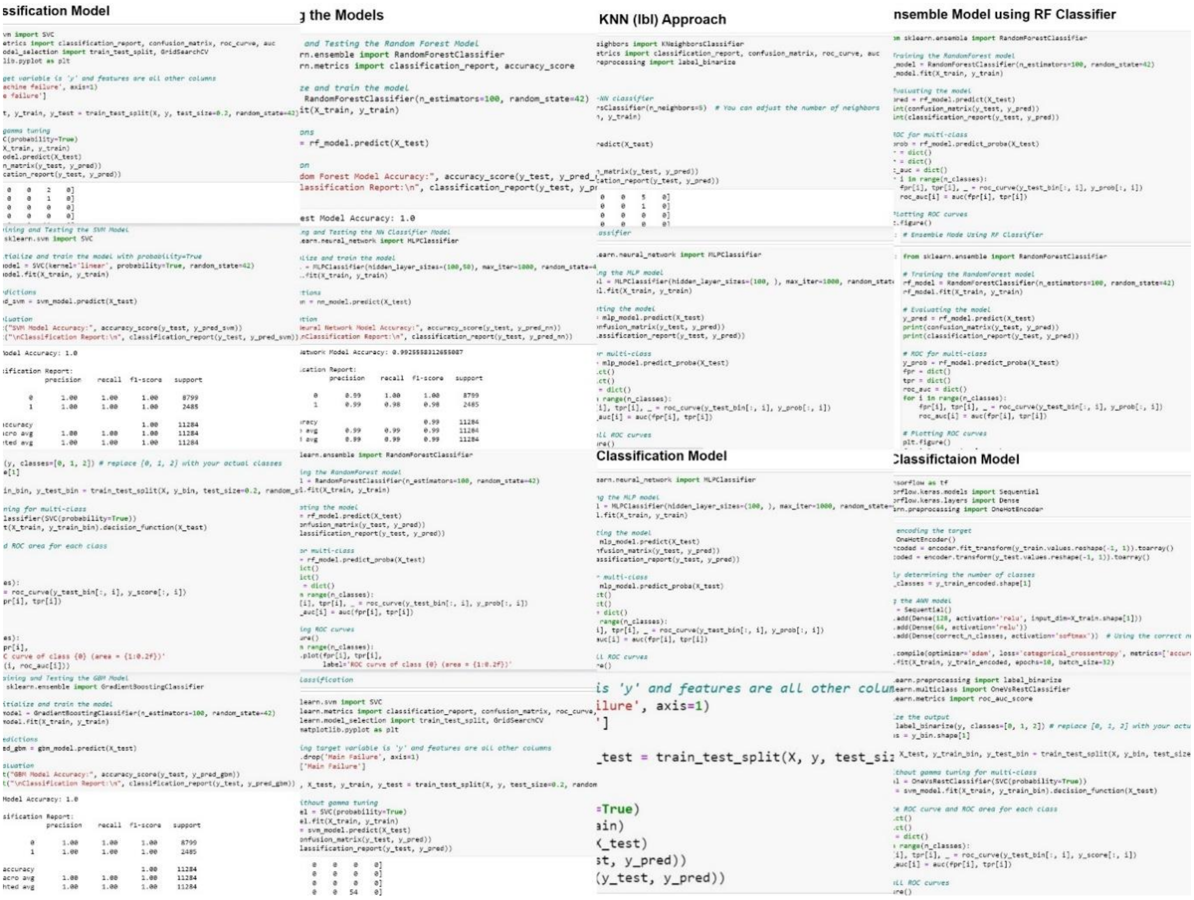

Figure 5: Code Snippet of Models

The research approach is based on the systematic gathering of data from an assortment of sensors and operational records, which reflects the complex interplay between machine performance and the surrounding environment. The implementation of preprocessing steps, including normalization, outlier removal, and missing value management, serves to guarantee the quality of the data. The rationale for choosing machine learning algorithms stems from their varied learning approaches and prospective applicability to distinct failure patterns. A selection of evaluation metrics is made, comprising accuracy, precision, recall, and F1-score, to offer a comprehensive evaluation of the predictive capability of each model. Priority is given to feature engineering to extract significant predictors from the unprocessed data. To enhance the dependability of the predictions, methods for mitigating model uncertainty, including confidence intervals and probabilistic thresholds, are implemented.

References

Fernandes, M., Canito, A., Bolón-Canedo, V., Conceição, L., Praça, I. and Marreiros, G., 2019. Data analysis and feature selection for predictive maintenance: A case-study in the metallurgic industry. International journal of information management, 46, pp.252-262.