Configuration Manual

MSc Research Project
Data Analytics

Sakshi Sanjay Kale
Student ID: x22219340

School of Computing
National College of Ireland

Supervisor: Naushad Alam

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Sakshi Sanjay Kale ……………………………………………………………………………………………………

Student ID: …x22219340……………………………………………………………………………………………………

Programme: Data Analytics ……………………………………………………………… **Year:** ……2023-2024 ……………………..

Module: …………MSc Research Project

Lecturer: ……Naushad Alam ……………………………………………………………………………………………………

Submission Due Date: ……12/08/2024 ……………………………………………………………………………………………………

Project Title: Predictive Maintenance of Equipment using Machine Learning Algorithms

Word Count:……………………………………… **Page Count:** …………………………………………………

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own 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 other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……………………………………………………………………………………………………

Date: ……………………………………………………………………………………………………

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):	

Configuration Manual

Sakshi Kale
X22219340@student.ncirl.ie
Data Analytics
National College of Ireland

1 Overview

This is the configuration manual for the " Predictive Maintenance of Equipment" work. The environment, pre-requests, and code execution will all be covered in detail in this configuration manual.

2 Hardware / Software Requirements

2.1 Hardware Requirements

The hardware configuration of the system on which this research project is built and executed is as follows:

Operating System Name: Microsoft Windows 11 Pro

Operating System Version: 10.0.22631 N/A Build 22631

Caption: Intel64 Family 6 Model 140 Stepping 1

DeviceID : CPU0

MaxClockSpeed : 2419

Name: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz

Total Physical Memory: 16,210 MB

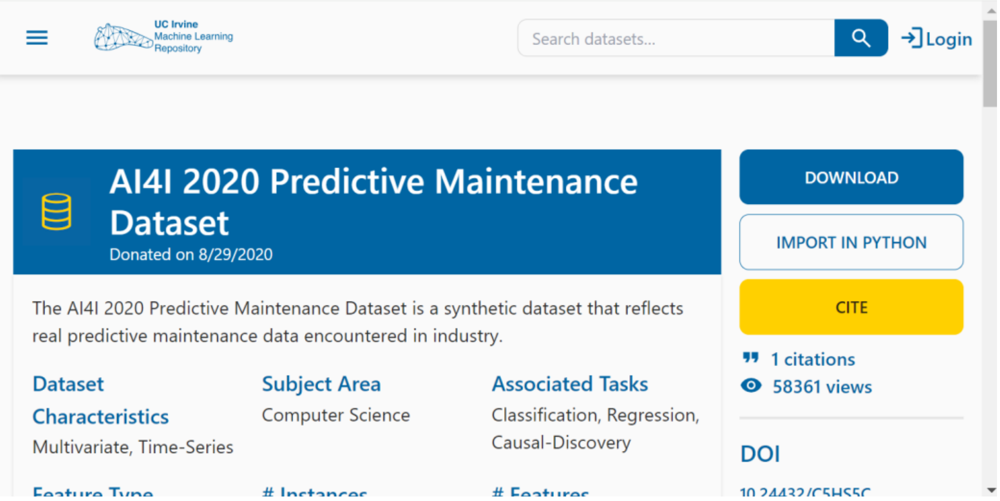
2.2 Software Requirements

Software required for build and execution:

- Development Environment: Jupyter Notebook
- Scripting Language: Python 3.11.8
- Other Tool: Word, Anaconda.

2 Data Selection

The dataset about equipment is obtained from the learning repository which is named as AI41 2020 Predictive Maintenance dataset. It is obtained from the following link: https://archive.ics.uci.edu/dataset/601/ai4i+2020+predictive+maintenance+dataset\



AI4I 2020 Predictive Maintenance Dataset

Donated on 8/29/2020

The AI4I 2020 Predictive Maintenance Dataset is a synthetic dataset that reflects real predictive maintenance data encountered in industry.

Dataset Characteristics
Multivariate, Time-Series

Subject Area
Computer Science

Associated Tasks
Classification, Regression, Causal-Discovery

1 citations
58361 views

DOI
10.24432/C5HS5C

Feature Type

# Instances

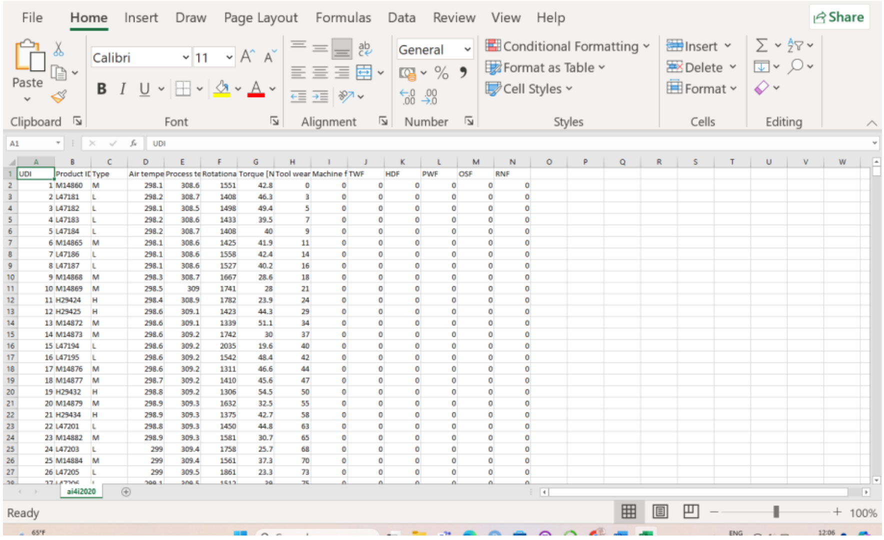
# Features

Figure 1: Dataset Website

3 Data transformation and Model Building

3.1 Dataset

The dataset below contains the following attributes like Air temperature, Process temperature, Rotational Speed, Torque, Tool Wear, Machine failure, TWF

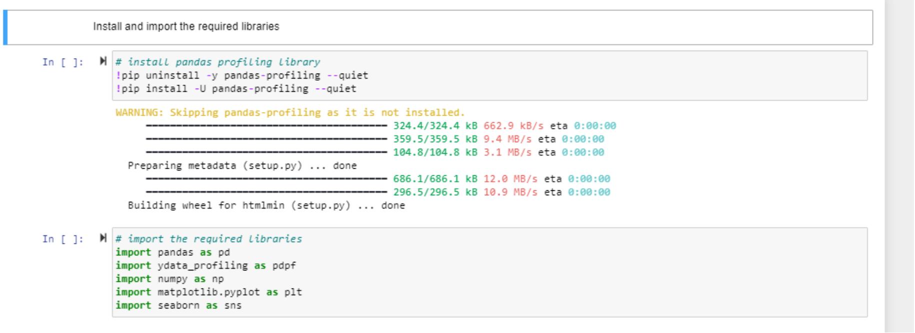


	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	UDI	Product ID	Type	Air tempe	Process te	Rotationa	Torque [N	Tool wear	Machine f	TWF	HDF	PWF	OSF	RNF
2	1	M14860	M	298.1	308.6	1551	42.8	0	0	0	0	0	0	0
3	2	L47181	L	298.2	308.7	1408	46.3	3	0	0	0	0	0	0
4	3	L47182	L	298.1	308.5	1498	49.4	5	0	0	0	0	0	0
5	4	L47183	L	298.2	308.6	1433	39.5	7	0	0	0	0	0	0
6	5	L47184	L	298.2	308.7	1408	40	9	0	0	0	0	0	0
7	6	M14865	M	298.1	308.6	1425	41.9	11	0	0	0	0	0	0
8	7	L47186	L	298.1	308.6	1558	42.4	14	0	0	0	0	0	0
9	8	L47187	L	298.1	308.6	1527	40.2	16	0	0	0	0	0	0
10	9	M14868	M	298.3	308.7	1667	28.6	18	0	0	0	0	0	0
11	10	M14869	M	298.5	309	1741	28	21	0	0	0	0	0	0
12	11	H29424	H	298.4	308.9	1782	23.9	24	0	0	0	0	0	0
13	12	H29425	H	298.6	309.1	1423	44.3	29	0	0	0	0	0	0
14	13	M14872	M	298.6	309.1	1339	51.1	34	0	0	0	0	0	0
15	14	M14873	M	298.6	309.2	1742	30	37	0	0	0	0	0	0
16	15	L47194	L	298.6	309.2	2035	19.6	40	0	0	0	0	0	0
17	16	L47195	L	298.6	309.2	1542	48.4	42	0	0	0	0	0	0
18	17	M14876	M	298.6	309.2	1311	46.6	44	0	0	0	0	0	0
19	18	M14877	M	298.7	309.2	1410	45.6	47	0	0	0	0	0	0
20	19	H29432	H	298.8	309.2	1306	54.5	50	0	0	0	0	0	0
21	20	M14879	M	298.9	309.3	1632	32.5	55	0	0	0	0	0	0
22	21	H29434	H	298.9	309.3	1375	42.7	58	0	0	0	0	0	0
23	22	L47201	L	298.8	309.3	1450	44.8	63	0	0	0	0	0	0
24	23	M14882	M	298.9	309.3	1581	30.7	65	0	0	0	0	0	0
25	24	L47203	L	299	309.4	1758	25.7	68	0	0	0	0	0	0
26	25	M14884	M	299	309.4	1561	37.3	70	0	0	0	0	0	0
27	26	L47205	L	299	309.5	1861	23.3	73	0	0	0	0	0	0
28	27	L47206	L	299.1	309.5	1512	39	75	0	0	0	0	0	0

Figure 2: Dataset

Package installations and library importing

The required libraries are installed such as pandas as pd, ydata_profiling as pdpf, numpy as np, matplotlib.pyplot as plt, seaborn as sns



Install and import the required libraries

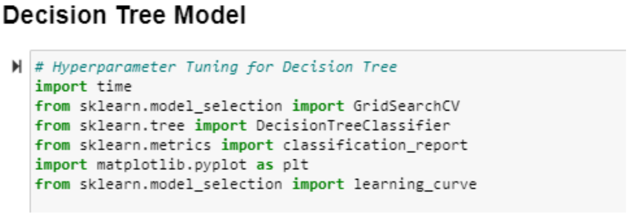
```
In [ ]:  # install pandas profiling library
         !pip uninstall -y pandas-profiling --quiet
         !pip install -U pandas-profiling --quiet

         WARNING: Skipping pandas-profiling as it is not installed.
              ---------------------------------------- 324.4/324.4 kB 662.9 kB/s eta 0:00:00
              ---------------------------------------- 359.5/359.5 kB 9.4 MB/s eta 0:00:00
              ---------------------------------------- 104.8/104.8 kB 3.1 MB/s eta 0:00:00
           Preparing metadata (setup.py) ... done
              ---------------------------------------- 686.1/686.1 kB 12.0 MB/s eta 0:00:00
              ---------------------------------------- 296.5/296.5 kB 10.9 MB/s eta 0:00:00
           Building wheel for htmlmin (setup.py) ... done

In [ ]:  # import the required libraries
         import pandas as pd
         import ydata_profiling as pdpf
         import numpy as np
         import matplotlib.pyplot as plt
         import seaborn as sns
```

Figure 3 : Package Installation

The below figure is for libraries for Decision Tree Models such as GridSearchCV, DecisionTreeClassifier, classification_report, etc.

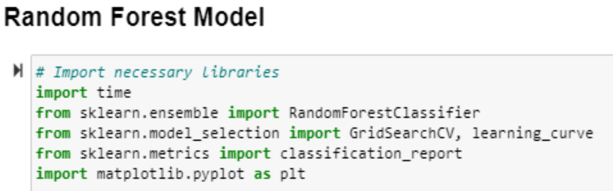


Decision Tree Model

```
# Hyperparameter Tuning for Decision Tree
import time
from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
from sklearn.model_selection import learning_curve
```

Figure 4: Package Installation for Decision Tree Model

The below figure is for libraries for Random Forest Models such as RandomForestClassifier, GridSearchCV, classification_report, learning_curve.

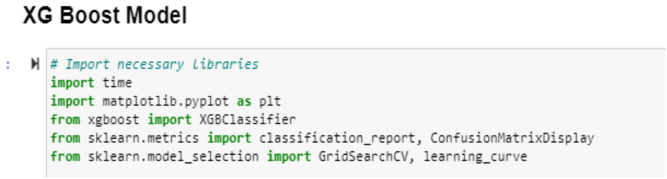


Random Forest Model

```
# Import necessary libraries
import time
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV, learning_curve
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
```

Figure 5: Package Installation for Random Forest Model

The below figure is for libraries for Random Forest Models such as XGBClassifier, ConfusionMatrixDisplay, classification_report, learning_curve.



XG Boost Model

```
# Import necessary libraries
import time
import matplotlib.pyplot as plt
from xgboost import XGBClassifier
from sklearn.metrics import classification_report, ConfusionMatrixDisplay
from sklearn.model_selection import GridSearchCV, learning_curve
```

Figure 6: Package Installation for XG Boost Model

4 Model Implementation

4.1 Decision Tree Model

The phase of model building began with specific actions that involved the making and training of a Decision Tree model. First, it was necessary to create an instance of the DecisionTreeClassifier, which is a popular decision tree-implementing tool in the Python programming language.

4.2 Random Forest Model

The model-building process continued with the development of a Random Forest model. Initially, the RandomForestClassifier of the sci-kit-learn library was initialized. It is well known that this classifier is resilient and capable of handling large more complex datasets.

4.3 XGBoost Model

The effectiveness and capability of this XGBoost model are outstanding, particularly for a huge amount of data. Therefore, as the initial step we called the XGBClassifier that is a powerful tool specially developed for implementing gradient boosting algorithms from the xgboost package.

4.3.1 Hyperparameter Tuning Using Grid Search

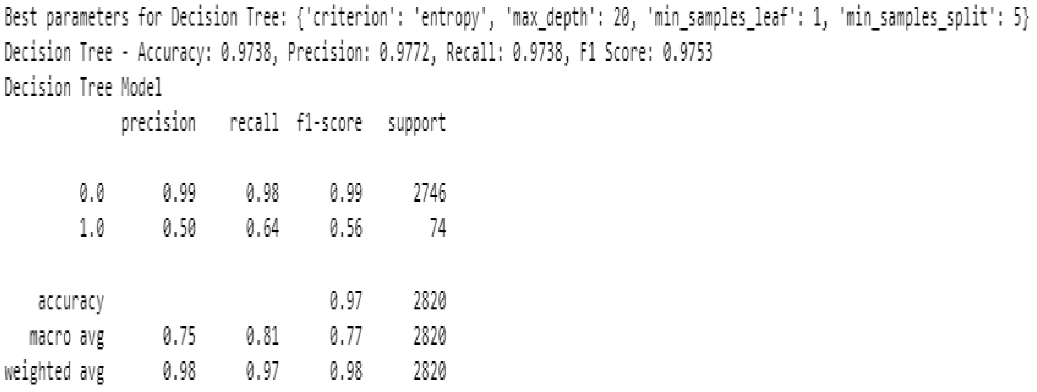
When it comes to hyperparameter tuning of the Decision Tree, Random Forest, and XGBoost models, a grid search with cross-validation. This procedure consists of many crucial stages. First of all, the set of all possible combinations of hyperparameters is generated, wherein the values of the key parameters are stated. For instance, adjustments of such factors as max_depth, min_samples_split, and n_estimators are considered to determine the impact they have on the model.

4.4 Feature Transformation

.If the variance of predictors is too high, the method of normalization can be applied using PowerTransformer in sci-kit-learn. Essentially, this form of modification is especially beneficial when it comes to processes like linear models which are used where measures from testing are normally superficial. It includes making the variance stable and transforming features into normality lies in a better and more reliable way of making the prediction thus enhances the capability of the models of making the prediction.

5 Model Evaluation

The figure below shows the evaluation and results for the Decision Tree Model which includes the precision, recall, f1-score, accuracy, macro avg, weighted avg. Random Forest, XGBoost, and Decision Tree provide an awesome general accuracy with Decision Tree having 97% general accuracy and Random Forest and XGBoost having 98% general accuracy.

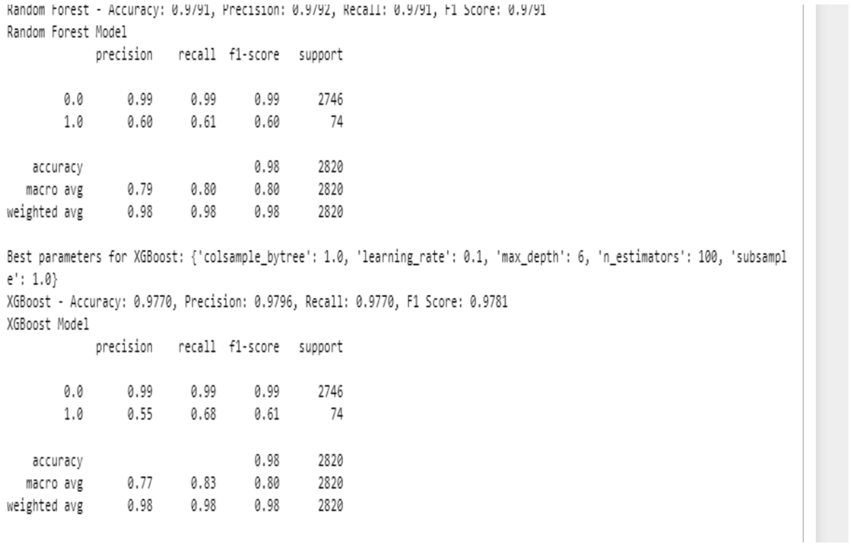


```
Best parameters for Decision Tree: {'criterion': 'entropy', 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 5}
Decision Tree - Accuracy: 0.9738, Precision: 0.9772, Recall: 0.9738, F1 Score: 0.9753
Decision Tree Model
              precision    recall  f1-score   support

         0.0       0.99      0.98      0.99      2746
         1.0       0.50      0.64      0.56        74

    accuracy                           0.97      2820
   macro avg       0.75      0.81      0.77      2820
weighted avg       0.98      0.97      0.98      2820
```

Figure 7: Evaluation for Decision Tree Model



```
Random Forest - Accuracy: 0.9791, Precision: 0.9792, Recall: 0.9791, F1 Score: 0.9791
Random Forest Model
              precision    recall  f1-score   support

         0.0       0.99      0.99      0.99      2746
         1.0       0.60      0.61      0.60        74

    accuracy                           0.98      2820
   macro avg       0.79      0.80      0.80      2820
weighted avg       0.98      0.98      0.98      2820

Best parameters for XGBoost: {'colsample_bytree': 1.0, 'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 100, 'subsampl
e': 1.0}
XGBoost - Accuracy: 0.9770, Precision: 0.9796, Recall: 0.9770, F1 Score: 0.9781
XGBoost Model
              precision    recall  f1-score   support

         0.0       0.99      0.99      0.99      2746
         1.0       0.55      0.68      0.61        74

    accuracy                           0.98      2820
   macro avg       0.77      0.83      0.80      2820
weighted avg       0.98      0.98      0.98      2820
```

Figure 8 : Evaluation for Random Forest Model and XGBoost Model