

## 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

		Irelan
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	

National

College of

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.

<u>ALL</u> 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: (<u>https://www.kaggle.com/datasets/binaicrai/machine-failure-data</u>)

### 2. ai4i2020 (D2)

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

### 3. binary classification of machine failures dataset (D3)

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

#### 4. Explainable AI (XAI) Drilling Dataset

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

#### 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.

	Item	Value
dware Resources	OS Name	Microsoft Windows 11 Pro
nponents	Version	10.0.21996 Build 21996
Software Environment	Other OS Description	Not Available
	OS Manufacturer	Microsoft Corporation
	System Name	DESKTOP-88TFFC4
	System Manufacturer	HP
	System Model	HP EliteBook 830 G5
	System Type	x64-based PC
	System SKU	6FN52UP#ABA
	Processor	Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz, 1896 Mhz, 4 Core(s), 8 Logical Pro
	BIOS Version/Date	HP Q78 Ver. 01.26.00, 02-10-2023
	SMBIOS Version	3.1
	Embedded Controller Version	4.110
	BIOS Mode	VEFI
	BaseBoard Manufacturer	HP
	BaseBoard Product	8383
	BaseBoard Version	KBC Version 04.6E.00
	Platform Role	Mobile
	Secure Boot State	Off
	PCR7 Configuration	Elevation Required to View
	Windows Directory	C.\WINDOWS
	System Directory	C:\WINDOWS\system32
	Boot Device	\Device\HarddiskVolume2
	Locale	United States
		1
	Hardware Abstraction Layer	Version = "10.0.21996.1" Activate Windows
Find what:	Hardware Abstraction Layer	Version = "10.0.21996.1" Activate Windows Find Go to Settin Glose Find, are Windows.
Find what:	Hardware Abstraction Layer	Version = "10.0.21996.1"         Activate Windows           Find         Go to \$ettin\$loss Find_ate Windows.           y         Find         Find_ate Windows.
Find what:	Hardware Abstraction Layer	Version = "10.0.21996.1"         Activate Windows           Find         Go to \$ettin <b>\$lose Find</b> .ate Windows.           y         Find         Find
Find what:	Hardware Abstraction Layer	Version = "10.0.21996.1" Activate Windows Find Go to \$ettin <b>Stose Find</b> are Windows. Y
Find what:	Hardware Abstraction Layer	Version = "10.0.21996.1" Activate Windows Find Go to \$ettin <b>Slose Find</b> rate Windows. y DESKTOP-88TFFC4supri
Find what:	Hardware Abstraction Layer	Version = "10.0.21996.1" Activate Windows Find Go to \$ettin \$loss Find.ate Windows y DESKTOP-88TFFC4/supri GMT \$landard Time
Find what:	Hardware Abstraction Layer     Search category names only     User Name     Time Zone     Installed Physical Memory (PAN	Version = "10.0.21996.1" Activate Windows Find Go to \$ettinglose Find ate Windows. y DESKTOP-88TFFC4supri GMT Standard Time A 150 GB
Find what:	User Name Time Zone Installed Physical Memory (RAM	Version = "10.0.21996.1" Activate Windows Find Go to \$ettin\$lose Find.ate Windows y DESKTOP-88TFFC4/supri GMT Standard Time 4) 160 G8
Find what:	User Name Time Zone Installed Physical Memory (RAM Total Physical Memory	Version = "10.0.21996.1" Activate Windows Find Go to \$ettin§tose Find.ate Windows. y DESKTOP-88TFFC4\supri GMT Standard Time 4) 150 GB 158 GB 283 GB
Find what:	User Name Time Zone Installed Physical Memory (RAM Total Physical Memory Available Physical Memory	Version = "10.0.21996.1" Activate Windows Find Go to \$ettin\$loss find.ate Windows y DESKTOP-88TFFC4supri GMT Standard Time 4) 16.0 G8 15.8 G8 8.87 C8 9 o e e
Find what:	Hardware Abstraction Layer  Hardware Abstraction Layer  User Name  Time Zone Installed Physical Memory (RAM Total Physical Memory Available Physical Memory Total Virtual Memory	Version = "10.0.21996.1" Activate Windows Find Go to \$ettin\$close Find.ate Windows y DESKTOP-88TFFC4\supri GMT Standard Time 4) 16.0 GB 18.8 GB 18.8 GB 18.2 GB
Find what:	User Name Time Zone Installed Physical Memory (RAM Total Physical Memory Available Physical Memory Total Virtual Memory Available Virtual Memory	Version = "10.0.21996.1" Activate Windows Find Go to \$ettingtose Find.ate Windows. y DESKTOP-88TFFC4supri GMT Standard Time 4) 16.0 G8 15.8 G8 8.87 G8 18.2 G8 10.3 G8 10.3 G8
Find what:	Hardware Abstraction Layer  User Name Time Zone Installed Physical Memory (RAN Total Physical Memory Available Physical Memory Available Virtual Memory Available Virtual Memory Page File Space	Version = "10.0.21996.1" Activate Windows y DESKTOP-88TFFC4/supri GMT Standard Time 4) 16.0 GB 15.8 GB 8.87 GB 12.2 GB 10.3 GB 2.38 GB
Find what:	User Name Time Zone Installed Physical Memory (RAM Total Physical Memory Available Physical Memory Available Virtual Memory Available Virtual Memory Page File Space Page File	Version = "10.0.21996.1" Activate Windows Find Go to \$ettin\$tose Find.ate Windows. y DESKTOP-88TFFC4\supri GMT Standard Time 4) 16.0 GB 15.8 GB 15.8 GB 15.8 GB 15.8 GB 2.38 GB 2.38 GB 2.38 GB C(pagefile.sys
Find what:	User Name Time Zone Installed Physical Memory (RAM Total Physical Memory Available Physical Memory Total Virtual Memory Available Physical Memory Page File Space Page File Kernel DMA Protection	Version = "10.0.21996.1" Activate Windows y DESKTOP-88TFFC4/supri GMT Standard Time 4) 16.0 G8 15.8 G8 16.3 G8 17.8 G8 18.2 G8 10.3 G8 2.38 G8 C/pagefile.sys Off
Find what:	Hardware Abstraction Layer  User Name Time Zone Installed Physical Memory (RAM Total Physical Memory Available Physical Memory Total Virtual Memory Available Virtual Memory Page File Space Page File Kernel DMA Protection Virtualization-based security	Version = "10.0.21996.1" Activate Windows y DESKTDP-88TFFC4\supri GMT Standard Time 4) 160 GB 18.8 GB 8.87 GB 18.2 GB 10.3 GB 2.38 GB C\pagefile.sys Off Not enabled
Find what:	Hardware Abstraction Layer User Name User Name Time Zone Installed Physical Memory (RAM Total Physical Memory Available Physical Memory Available Physical Memory Page File Space Page File Kernel DMA Protection Virtualization-based security Device Encryption Support	Version = "10.0.21996.1"  Activate Windows  Find Go to Settin Glose Find.ate Windows  V  DESKTOP-88TFFC4/supri GMT Standard Time 4) 16.0 G8 15.8 G8 8.87 G8 15.8 G8 2.8 G8 2.8 G8 2.8 G8 2.8 G8 2.8 G8 C/apagefile.sys Off Not enabled Elevation Required to View
Find what:	Hardware Abstraction Layer User Name Time Zone Installed Physical Memory (RAN Total Physical Memory Available Physical Memory Available Physical Memory Page File Space Page File Kernel DMA Protection Virtualization-based security Device Encryption Support Hyper-V-V Monitor Mode Ei	Version = "10.021996.1"  Activate Windows  Find Go to \$ettin\$Close Find.ate Windows  U  DESKTOP-88TFFC4\supri GMT Standard Time () 16.0 G8 18.8 G8 8.87 G8 18.2 G8 18.2 G8 18.2 G8 18.3 G8 2.38 G8 C\pagefile.sys Off Not enabled Elevation Required to View Xes Yes
Find what:	Hardware Abstraction Layer User Name User Name Time Zone Installed Physical Memory (RAM Total Physical Memory Available Physical Memory Available Physical Memory Available Physical Memory Page File Page File Kernel DMA Protection Virtualization-based security Device Encryption Support Hyper-V - VM Monitor Mode E Hyper-V - Second I eval Addrese	Version = "10.0.21996.1"  Activate Windows  Find Go to Settin Glose Find.ate Windows  V  DESKTOP-88TFFC4/supri GMT Standard Time  () 16.0 G8 15.8 G8 8.87 G8 15.2 G8 15.2 G8 15.2 G8 15.2 G8 CVpagefile.sys Off Not enabled Elevation Required to View x Yes Yes
Find what:	Hardware Abstraction Layer User Name Time Zone Installed Physical Memory (RAN Total Physical Memory Available Physical Memory Available Physical Memory Available Virtual Memory Page File Kernel DMA Protection Virtualization-based security Device Encryption Support Hyper-V - Second Level Addres Hyper-V - Second Level Addres Hyper-V - Second Level Addres	Version = "10.021996.1"  Activate Windows  Find Go to \$ettin\$tose Find.ate Windows  V  DE5KTOP-88TFFC4\supri GMT Standard Time 4) 16.0 GB 15.8 GB 15.8 GB 15.8 GB 15.8 GB 16.2 GB 10.3 GB 2.38 GB 10.3 GB 2.38 GB 10.3 GB 2.38 GB 10.4 GB Elevation Required to View K. Yes S. Yes Ves
Find what:	Hardware Abstraction Layer     User Name     Time Zone     Installed Physical Memory (RAM     Total Physical Memory     Available Physical Memory     Total Virtual Memory     Available Virtual Memory     Available Virtual Memory     Page File     Remel DMA Protection     Virtualization-based security     Device Encryption Support     Hyper-V - Second Level Addres     Hyper-V - Nirtualization Enable     Hyper-V - Data Securite memory	Version = "10.0.21996.1"  Activate Windows  Find Go to Settin Close Find.ate Windows  DESKTOP-88TFFC4/supri GMT Standard Time  () 16.0 G8  15.8 G8  8.87 G8  13.2 G8  13.2 G8  13.3 G8  2.38 G8  C/apagefle.sys Off Not enabled Elevation Required to View  x Yes  Yes  Yes  Yes
Find what:	User Name Time Zone Installed Physical Memory (RAM Total Physical Memory Available Physical Memory Total Virtual Memory Available Physical Memory Page File Space Page File Kernel DMA Protection Virtualization-based security Device Encrybion Support Hyper-V - VM Monitor Mode El Hyper-V - VM Monitor Enable. Hyper-V - Data Execution Prote	Version = "10.0.21996.1"  Activate Windows  Find Go to Settin Close Find.ate Windows  DESKTOP-88TFFC4/supri GMT Standard Time  () 16.0 G8 13.8 G8 8.87 G8 13.8 G8 13.8 G8 2.38 G8 C(pagefile.sys Off Not enabled Elevation Required to View  x Yes  Yes  Yes Activate Windows

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, aud
        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, aud
        from sklearn.preprocessing import label_binarize
        import pandas as pd
        import numpy as no
        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, aud
        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

ssification Model	] the Models	KNN (Ibl) Approach	nsemble Model using RF Classifier
vn import SVC atrics import classification report, confusion matrix, roc curve, auc	and Testing the Random Forest Madel	sighbors import ElleighborsClassifian	im sklearn.ensembla import RandomForestClassifier
odal_salaction import train_test_split, dridSearchCV lis.pyplot as plt	<pre>rn.ensemble import RandomForestClassifier rn.metrics import classification_report, accuracy_score</pre>	reprocessing import classification_report, confusion_metrix, roc_curve, auc reprocessing import label_binarize	Training the KandomForest model _model = RandomForestClassifier(n_estimators=100, random_state=42) _model fit(r ranks v ranks)
<pre>pet variable is 'y' and features are all other columns achine failure', axis=1)</pre>	ze and train the model		Descention the ender
e failure']	RandomForestClassifier(n_estimators=100, random_state=42)	-NV classifier 	<pre>ired = rf_model.predict(X_test) if(continue</pre>
<pre>t, y_train, y_test = train_test_split(%, y, test_size=0.2, random_state=</pre>	<pre>l2jit(X_train, y_train)</pre>	1, y_train)	<pre>int(classification_report(y_test, y_pred)) int(classification_report(y_test, y_pred))</pre>
gomma tuning C(probability=True)	ons		10C for multi-class
X_train, y_train)	= rt_model.predict(X_test)	redict(X_test)	<pre>srob = rf_model.predict_probe(X_test)</pre>
n_matrix(y_test, y_pred))	on	, n_matrix(y_test, y_pred))	<pre>' = dict() '_auc = dict()</pre>
	<pre>lassification Report:\n", classification_report(y_test, y_pred</pre>	tation_report(y_test, y_pred)) _pr	<pre>i im range(n_classes): fpr[i], tpr[i], _ = roc_curve(y_test_bin[:, i], y_prob[:, i])</pre>
e e 1 ej		0 0 5 0] 0 0 1 0]	<pre>roc_auc[i] = auc(fpr[i], tpr[i])</pre>
e e e e]	est Model Accuracy: 1.0	e e e e]	<pre>% stating ROC curves % figure()</pre>
whing and Testing the SWW Model sklearn.svn import SVC	ing and Testing the NW Classifier Model	assifier	: # Ensemble Hode Using RF Classifier
tiplize and train the model with probabilitysTrue	the strength and a set of the set	.earn.neural_metwork import #LPClassifier	from sklaarn ansamble innort BandomForestClassifiar
<pre>odel = SVC(kernel='linear', probability=True, random_state=42) odel fir(X train u train)</pre>	- MLPClassifier(hidden_layer_sizes=(100,50), max_iter=1000, random_state finite and a factor.	ind the RLP model	# Topining the Rephylorest andel
DOBLITIC(_TTBIN, Y_TTBIN)	tit(x_train, y_train)	<pre>il = MLPClassifier(hidden_layer_sizes=(100, ), max_iter=1000, random_sta 1 discussion = total</pre>	<pre>ft rf_model = RandomForestClassifier(n_estimators=100, random_state=42)</pre>
d[svm = svm_model.predict(X_test)	<pre>itions in = nn_model.predict(X_test)</pre>	11.Tit(A_train, y_train)	<pre>rf_model.fit(X_train, y_train)</pre>
Leation	ntion	<pre>sting the model smlp_model.predict(X_test)</pre>	<pre># Evaluating the model y_pred = rf_model.predict(X_test)</pre>
:("SWM Model Accuracy:", accuracy_score(y_test, y_pred_svm)) ("InClassification Report is", classification record(y test, y pred_svm)	<pre>leural Network Model Accuracy:", accuracy_score(y_test, y_pred_nn)) () of assification Report (a", classification report (v test, v pred_nn))</pre>	<pre>nfusion_matrix(y_test, y_pred)) .assification recort(v_test, v_pred))</pre>	<pre>print(confusion_matrix(y_test, y_pred)) print(classification_report(y_test, y_pred))</pre>
Indel Accuracy: 1.0	atuoric Model Accuracy: 0.9035558312655887		a per des estats stats
	cation Banati	<pre>mlp_model.predict_probe(X_test)</pre>	<pre>y_prob - rf_model.predict_probe(X_test)</pre>
precision recall fl-score support	precision recall f1-score support	.ct() .ct()	<pre>tpr = dict() tpr = dict()</pre>
e 1.00 1.00 1.00 8799	0 0.59 1.00 1.00 8759	<pre>= dict() i range(n_classes):</pre>	roc_suc = dict() for i in range(n_classes):
1 1.00 1.00 1.00 2485	1 0.99 0.98 0.98 2485	<pre>i], tpr[i], _ = roc_curve(y_test_bin[:, i], y_prob[:, i]) aur[i] = aur(fpr[i], tpr[i])</pre>	<pre>fpr[i], tpr[i], _ = roc_curve(y_test_bin[:, i], y_prob[:, i]) roc_auc[i] = auc(for[i], tpr[i])</pre>
Iccuracy 1.00 11284	irecy 0.99 11284 h avg 0.99 0.99 0.99 11284	Tandial - and the fall the fall	# Blatting BV comes
ted evg 1.00 1.00 1.00 11284	1 avg 0.99 0.99 0.99 11284	ine()	plt.figure()
(v. classes=[0, 1, 2]) # replace (0, 1, 2) with your actual classes	learn.ensamble import RandomForestClassifier	Classification Model	Classifictaion Model
e[1]	ing the RandomForest model 1 = RandomForestClassifier(n estimators=100, random state=42)	sarn.meural network import MLPClassifier	condition on the
<pre>in_bin, y_test_bin = train_test_solit(X, y_bin, test_size=0.2, random</pre>	_sl.fit(X_train, y_train)	to the NR andel	orflow.kerss.models import Sequential
ning for multi-class	sting the model	<pre>i = MLPClassifier(hidden_layer_sizes=(100, ), max_iter=1000, random_state</pre>	orflow.keras.layers import Dense ""irn.proprocessing import DneHotEncoder
lassifier(SVC(probability=True)) t(X train, y train bin).decision function(X test)	<pre>= rf_model.predict(X_test) prfusion_matrix(y_test, y_pred))</pre>	1.Tit(A_train, y_train)	and the second
d RVC mass day parts plans	lassification_report(y_test, y_pred))	ting the model nlp_model.predict(X_test)	OneHotEncoder()
a soc area far each crass	<pre>or multi-class - rf model.predict proba(X test)</pre>	<pre>infusion_matrix(y_test, y_pred)) assification_report(y_test, y_pred))</pre>	<pre>icoded = encoder.fit_transform(y_train.values.reshape(-1, 1)).toarrey() icoded = encoder.transform(y_test.values.reshape(-1, 1)).toarrey()</pre>
	ict() let()	and the form	ly determining the number of classes
es): = roc curve(y test bin[:, i], y score[:, i])	= dict()	<pre>nlp_model.predict_proba(%_test)</pre>	_classes = y_train_encoded.shape[1]
pr[i], tpr[i])	<pre>[i], tpr[i], _ = roc_curve(y_test_bin[:, i], y_prob[:, i])</pre>	it() it()	7 the ANN model
	_auc[1] = auc(tpr[1], tpr[1])	<pre>* dict() range(n_classes):</pre>	.add(Dense(128, activation-'relu', input_dim-X_train.shape[1]))
es):	leg ROC curves ure()	<pre>i], tpr[i], _ = roc_curve(y_test_bin[:, i], y_prob[:, i]) suc[i] = suc(fpr[i], tpr[i])</pre>	.add(Dense(correct_n_classes, activation="softmax")) # Using the correct n
pr[i], C curve of class (0) (area = {1:0.2f})' (i, roc_auc[i]))	<pre>s range(n_classe): .plot(far[i], tpr[i], label="ROC curve of class {0} (area = {1:0.2f})'</pre>	(1 NDC curves re()	<pre>.compile(optimizer='adam', loss='categorical_crossentropy', mstrics=['accur .fit(X_train, y_train_encoded, epochs=10, batch_size=32)</pre>
sining and Testing the GBM Model	lassification	to full and frequency and all subservation	earn.preprocessing import label_binarize
skies i ensemple angurs or salers poussing lassifier	lass up impet GM	is y and features are all other colu	<pre>/// /////////////////////////////////</pre>
<pre>intoile and train the model votel = GradientBootingClassifier(n_estimators=100, random_state=42) votel.fit(%_train, y_train)</pre>	learn.metrics import classification_report, confusion_matrix, roc_curv learn.model_selection import train_test_split, GridSearchCV attaletics and a plt	']	<pre>ize the output label_binarize(y, classes=[0, 1, 2]) = replace [0, 1, 2] with your actu</pre>
edictions	and the state of t		<pre>is = y_bin.shape[1]</pre>
ed_gem = gen_model.predict(X_test)	.drop('Hain Failure', axis=1)	test = train test split(X, y, test si	2 X_test, y_train_bin, y_test_bin = train_test_split(X, y_bin, test_size
sluotion t("OBM Model Accuracy:", accuracy_score(y_test, y_pred_gom))	['Hain Failure']		thout gamma tuning for multi-class
t("\nClassification Report:\s", classification_report(y_test, y_pred_gbm	<pre>) , X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rand</pre>	lon	<pre>il = OnevsRestClassifier(SVC(probability=True)) = svm_model.fit(X_train, y_train_bin).decision_function(X_test)</pre>
Hodel Accuracy: 1.0	thout gamma tuning	:True)	te ROC curve and ROC area for each class
sification Report:	el = 5vc(probability=inwe) el.fit(X_train, y_train)	ain)	.et()
a 1.00 1.00 1.00 8700	<pre>= svm_model.predict(X_test) pnfusion_matrix(y_test, y_pred))</pre>	( tost)	- dict()
1 1.00 1.00 1.00 2485	lassification_report(y_test, y_pred))	(_test)	<pre>i range(n_classes): 'i], tsr[i], _ = roc_curve(y_test_bin[:, i], y_score[:, i])</pre>
accuracy 1.00 11284		st, y_pred))	<pre>_auc[i] = auc(fpr[i], tpr[i])</pre>
acro avg 1.00 1.00 1.00 11254 hted avg 1.00 1.00 1.00 11284	e e e e]	(y_test, y_pred))	ILL ROC curves

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.