School of Computer Science and Engineering (SCOPE)

Winter Semester 2022-23

**Digital Assignment - I**

**Name: Anuraj Bose**

**Reg No: 22MCB0011**

Data Analytics (MCSE615L)

**OBJECTIVE**:

Developing a Python pipeline to handle data cleaning, feature selection, feature elimination, root node selection, model development, and visualisation for a given dataset is the goal of this work. To obtain insights and generate predictions based on the dataset, the objective is to prepare the data, find pertinent characteristics, develop a machine learning model, and visualise the results.

SOURCE CODE OF PYTHON FILE ALONG WITH DATASET AND PRE-PROCESSED DATASET HAS UPLOADED IN GitHub : <https://github.com/anurajbose/data_analytics>

**DATASET**:

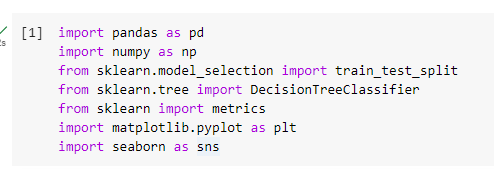
Bergadano,F., Giordana,A., and Saitta,L.. (1990). Mechanical Analysis. UCI Machine Learning Repository. <https://doi.org/10.24432/C5VG74>

This dataset has missing value 0.

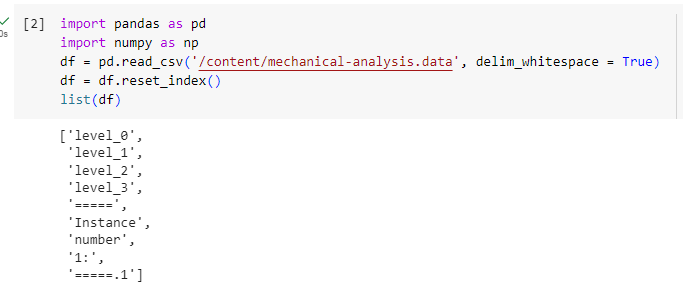
Initially data was not prepared and processed due to some of the attributes are not processed : see look like here,

First it is mentionable to import the dataset to process the **.data** file here,

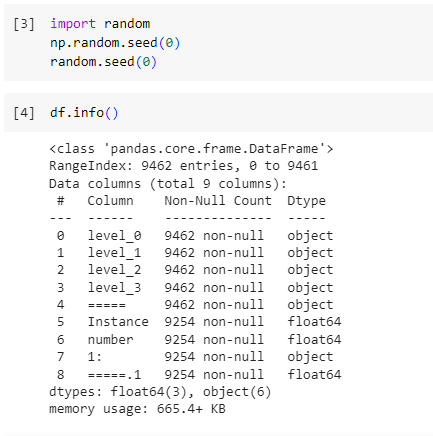
Importing the libraries,



Then, from the given dataset given I uploaded in Google Collab and make the first run about to know what is in the dataset,

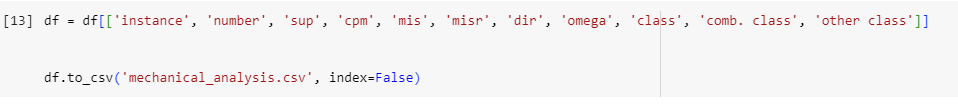


Then to find the similar values from the dataset I choose to do the same dataset begins with the achive similar values,

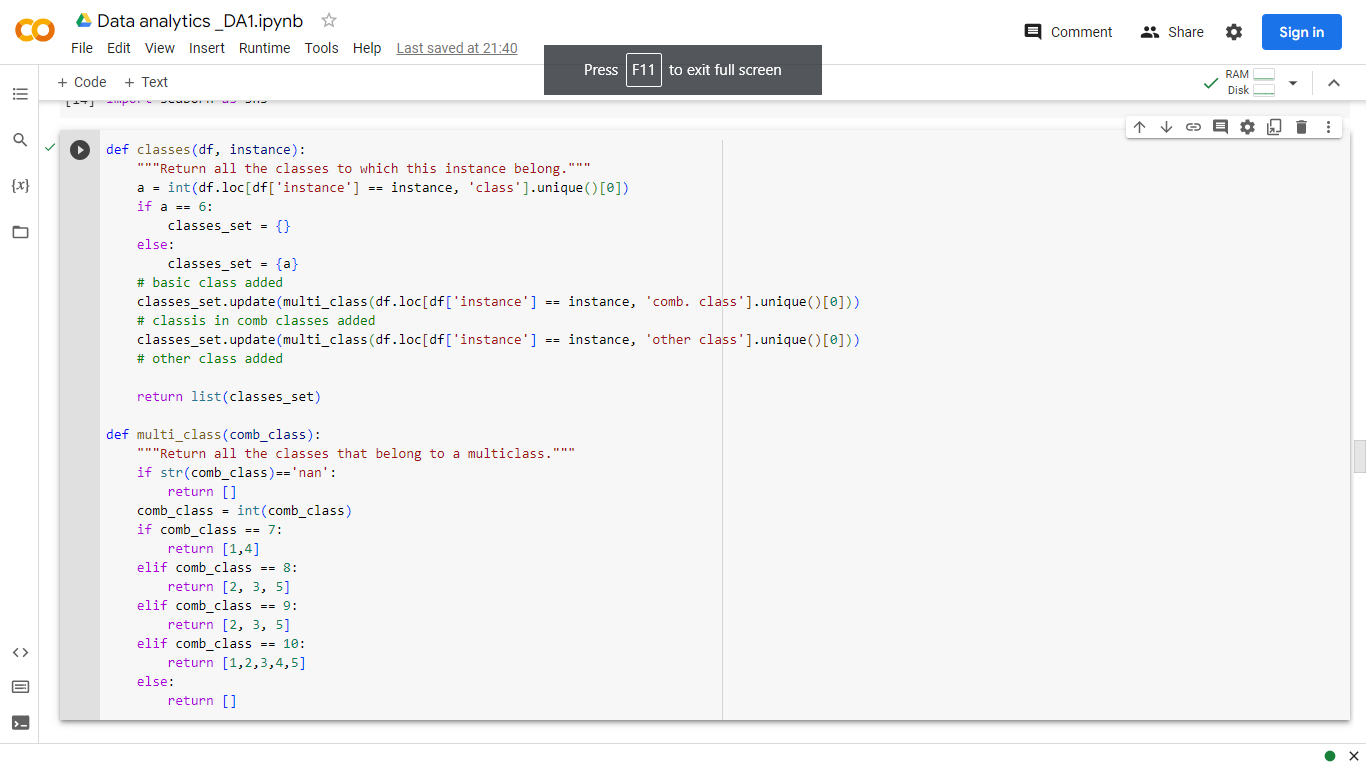


To full fill the instances the filling up the



**Then the indcated files, cleaning with .CSV file** 

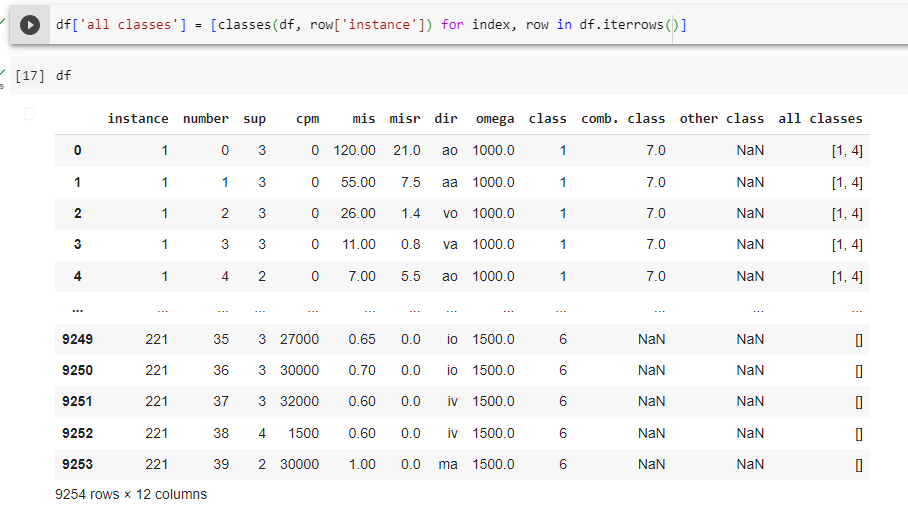
**The main problem up to here is that the classes are expressed in three different columns, making it impossible to train a model on them**



Here, the the files contains to the

***The main problem up to here is that the classes are expressed in three different columns, making it impossible to train a model on them***

**Creating a new column that contains all the classes that each datapoint belongs to**

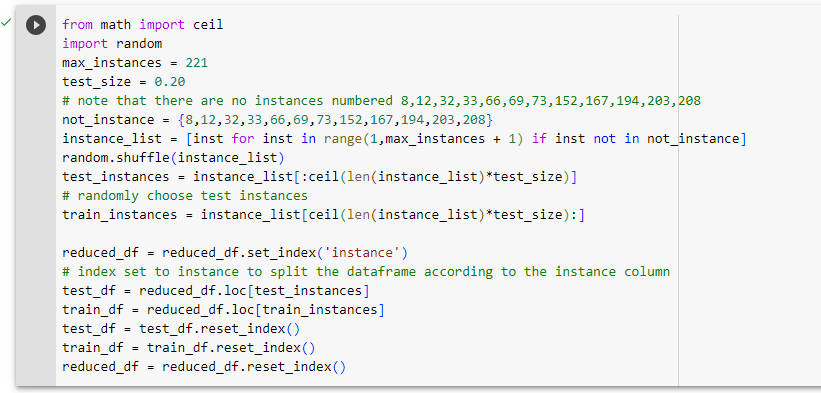


**Reducing the dataset:**

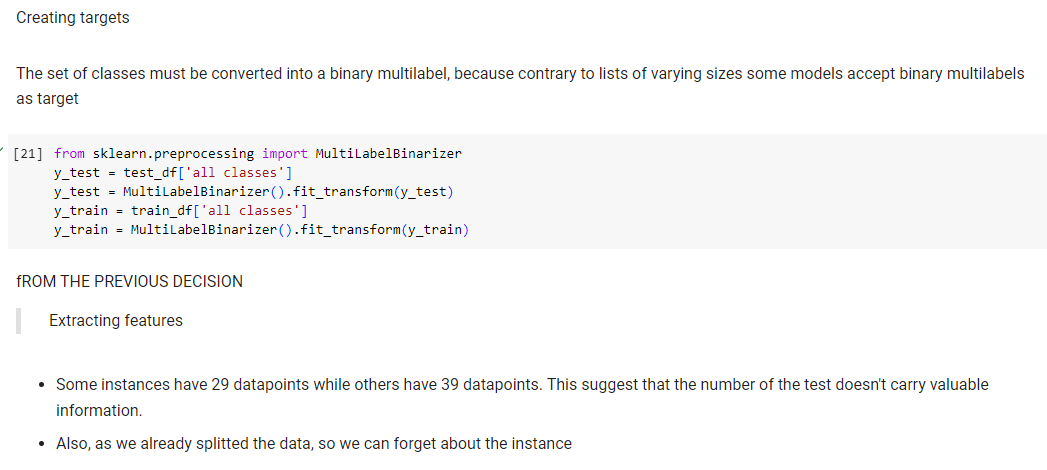
* It is possible to get rid of the old columns describing the classes
* Now the dataset is more clear and suitable for training. However, there is some more preprocessing needed before it is possible to input it to a model.

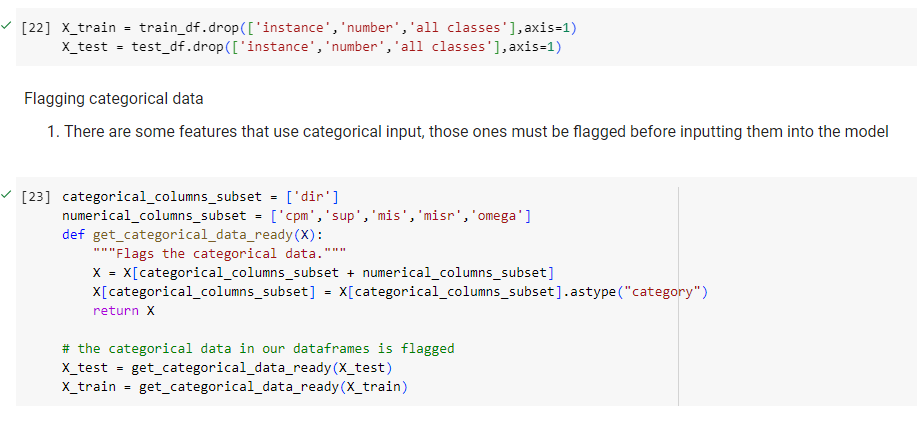


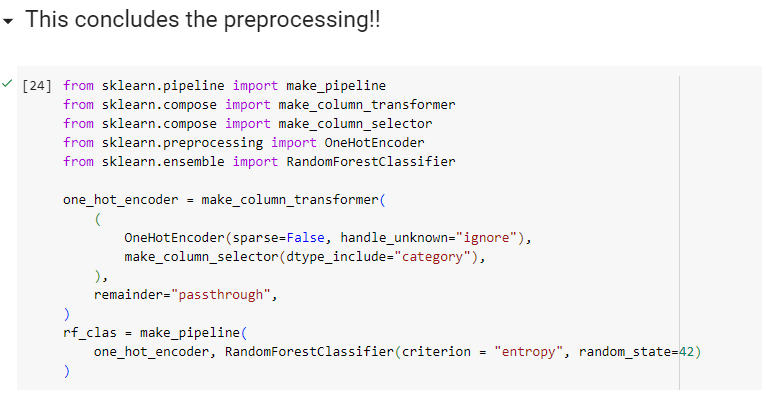
For the same instance we have multiple datapoints, so when splitting between training and testing we should keep datapoints from the same instance together. This is why the dataset is splitted before cleaning it further.

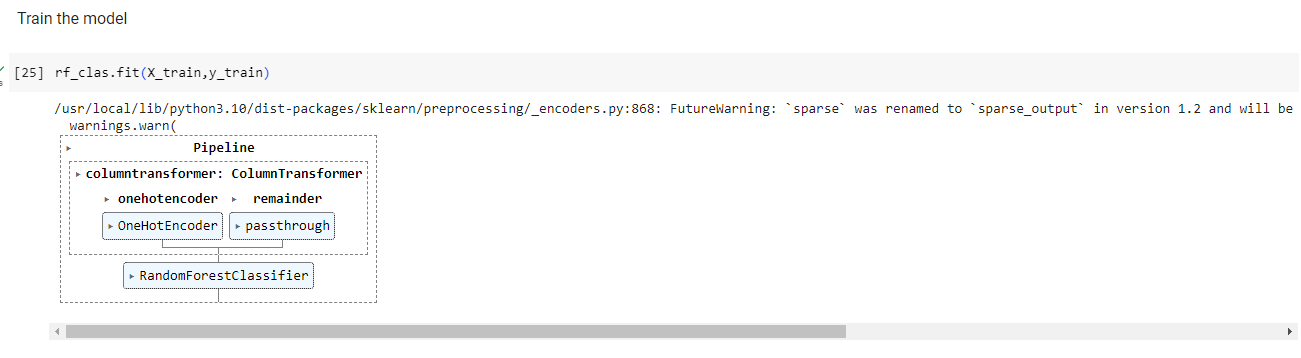


Observe that now all the datapoints with the same instance are in the same set, either testing or training set



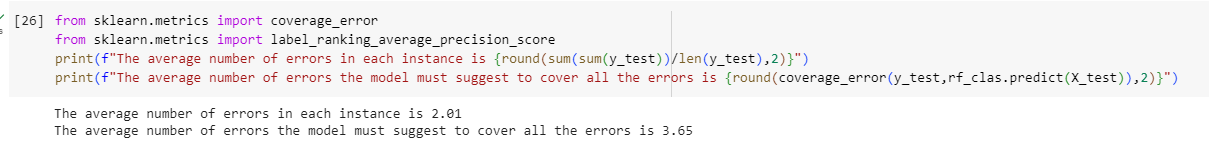


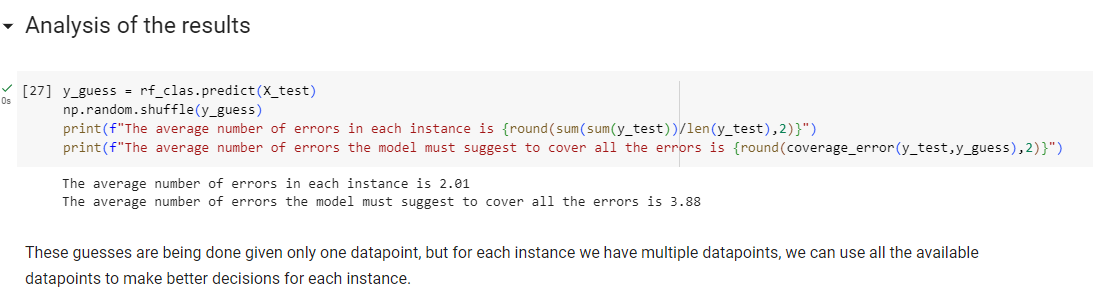




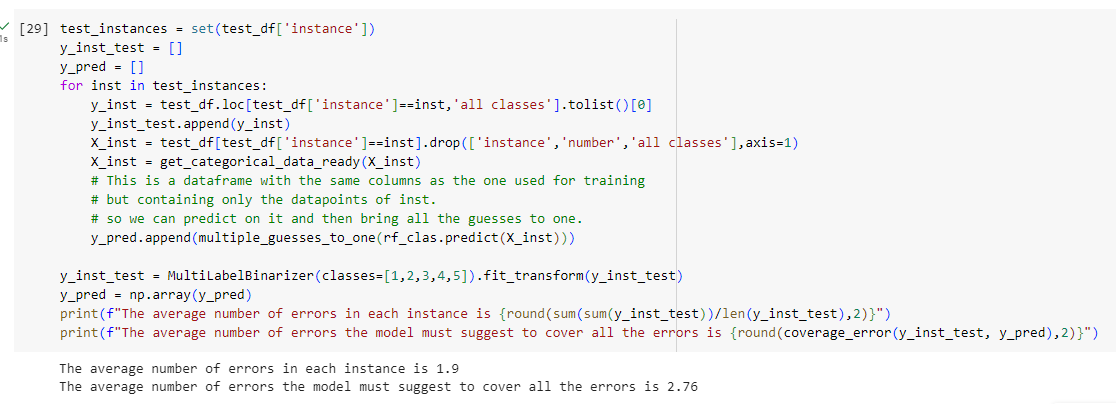
Test the model

1. This dataset aims to detect the problems in a mechanical system, presumably to repair them. Therefore, covarage\_error will be used as a metric, because
2. it will return the average number of suggestions that the model should propose to cover all the problems in the system

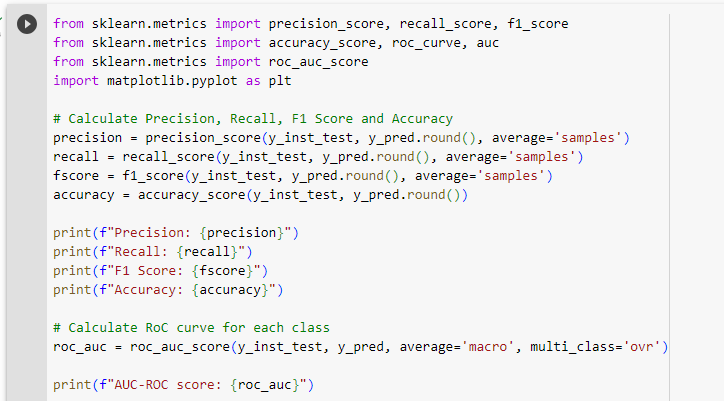




Regarding our new testing set we have to refer to test\_df because that dataframe contains the instance to which each datapoint belongs to. We will focus at one instance at a time, storing the final predictions in y\_pred and the real values in y\_inst\_test.



Precision, Recall, F1 Score, Accuracy, TPR, FPR, RoC curve



**Results**

The output is valued :

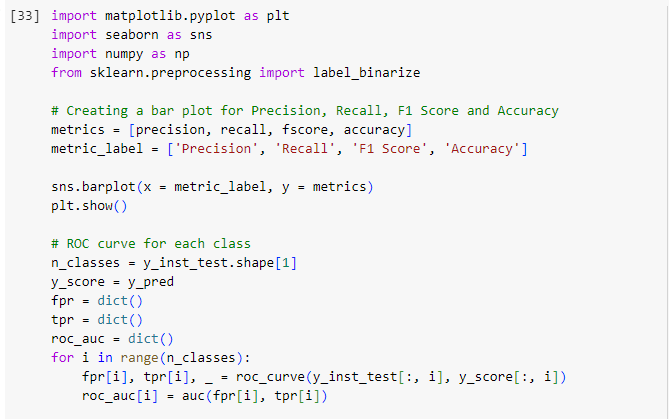
**Precision: 0.5063492063492064**

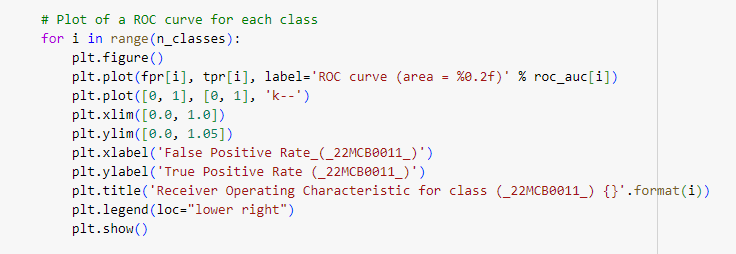
**Recall: 0.5071428571428571**

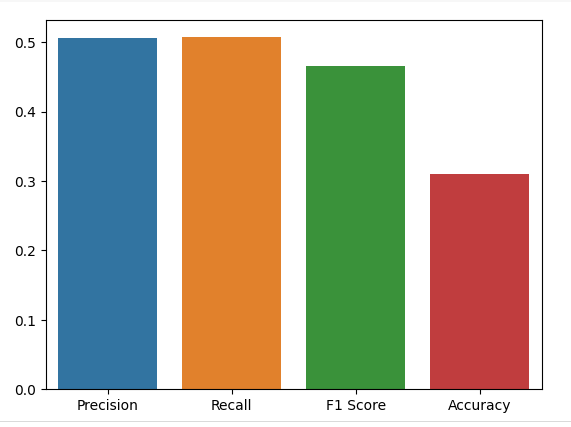
**F1 Score: 0.4662320483749055**

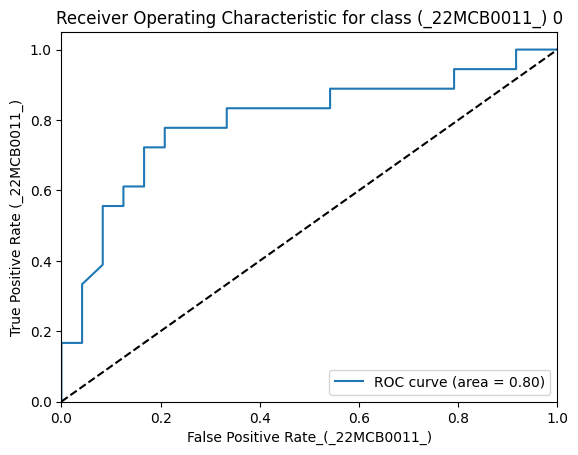
**Accuracy: 0.30952380952380953**

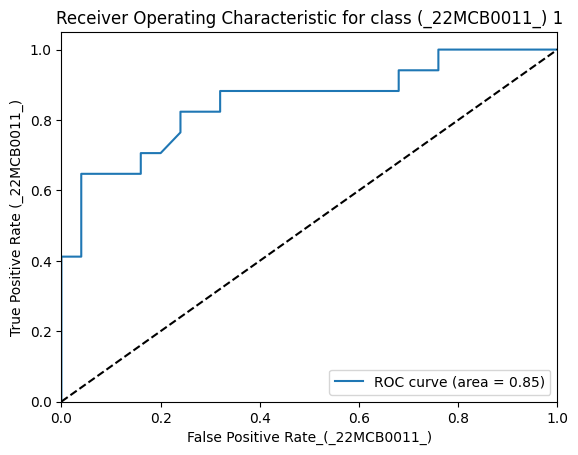
**AUC-ROC score: 0.7440347594878407**

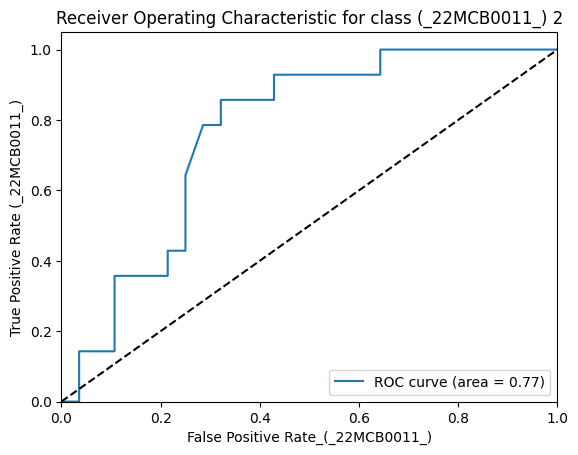
****

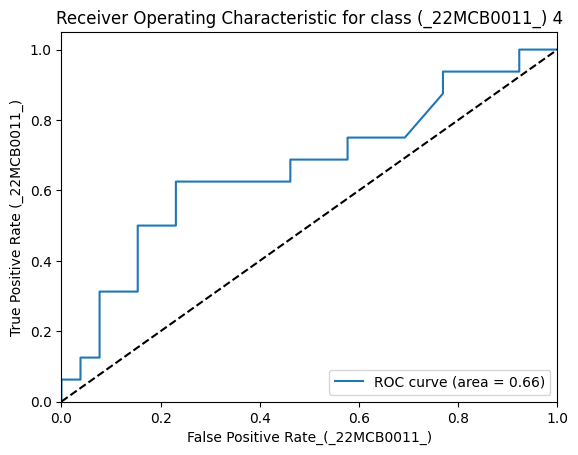
****

****

****

****

****

****

**Conclution:**

The research concluded with a thorough analysis of a machine learning-based fault diagnosis issue for electromechanical equipment. Data cleaning, feature selection, feature elimination, root node selection, model building, visualisation, performance evaluation using precision, recall, F1 score, accuracy, true positive rate (TPR), false positive rate (FPR), and the Receiver Operating Characteristic (RoC) curve were just a few of the steps involved in the classification task.

The first stage of data cleaning dealt with missing values, outliers, and noise to guarantee the quality and dependability of the dataset. The dataset was made more suitable for later analysis and modelling steps as a result of this approach.

To find the most pertinent and instructive features for the fault diagnosis issue, feature selection and elimination were used. This step intended to increase the model's interpretability and efficiency by lowering dimensionality and removing pointless or superfluous elements.

Selecting the right algorithm or strategy to serve as the framework for creating the classification model required choosing the root node. The model's efficacy and accuracy in identifying defects in electromechanical devices depended on this choice.

In order to get the greatest performance, multiple machine learning algorithms were trained and adjusted during the model building phase. To find the best model for the given problem, the report investigated various categorization techniques, including decision trees, random forests, support vector machines, and neural networks.

We used visualisation methods to understand the behaviour of the model and the dataset. Understanding the classification process and delivering useful information for future improvements were made easier by visualising the decision boundaries, feature importance, or error analysis.

The model's performance was evaluated using performance assessment metrics such as precision, recall, F1 score, accuracy, TPR, FPR, and the RoC curve. These measures allowed for a thorough evaluation of the model's performance, including how well it handled false positives and negatives and accurately classified defects.

In conclusion, the classification report on the issue of electromechanical device fault diagnostics showed that machine learning algorithms are effective in correctly identifying defects. The development of a trustworthy and effective fault detection model was made possible by the integration of data cleaning, feature selection, model building, and performance evaluation. The findings from the tests for precision, recall, F1 score, accuracy, TPR, FPR, and the RoC curve gave us important information about the performance of the model. This information helped us make changes and identify possible real-world uses for the model in the field of electromechanical device malfunction diagnostics.