iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** XII **Month of publication:** December 2022

DOI: https://doi.org/10.22214/ijraset.2022.48225

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XII Dec 2022- Available at www.ijraset.com

Detection of Parkinson's Disease Using XGBOOST Algorithm

Sanika Narayanpethkar[1], M. Rishitha[2], S. Chandana[3], Dr. T. Vijaya Saradhi[4]
[1, 2, 3]*Department of Computer Science & Engineering, SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY*

Abstract: PREDICTING PARKINSON'S DISEASE using XGBOOST ALGORITHM. In this project we will be using python to build a model using which we can accurately detect the presence of Parkinson's disease in one's body. XGBOOST algorithm is a technique for regression and classification problems. It produces a prediction model in form of a decision tree. Data is loaded, features and label are specified, data is split, XGBClassifier is produced and calculate the accuracy of our model.
Parkinson's disease is caused by the disruption of the brain cells that produce a substance to allow brain cells to communicate with each other, called dopamine. The cells that produce dopamine in the brain are responsible for the control, adaptation, and fluency of movements. When 60–80%of these cells are lost, then enough dopamine is not produced and Parkinson's motor symptoms appear. It is thought that the disease begins many years before the motor (movement-related) symptoms and therefore, researchers are looking for ways to recognize the non-motor symptoms that appear early in the disease as early as possible, thereby halting the progression of the disease. In this project, diagnosis of Parkinson's disease is presented. Using the python libraries scikit-learn, NumPy, pandas, and XGBoost, we will build a model using an XGBClassifier. We'll load the data, get the features and labels, scale the features, then split the dataset, build an XGBClassifier, and then calculate the accuracy of our model. 94.87% accuracy was achieved with the least number of voice features for Parkinson's diagnosis

I. INTRODUCTION

A. Project Introduction

PREDICTING PARKINSON'S DISEASE using XGBOOST ALGORITHM. In this project we will be using python to build a model using which we can accurately detect the presence of Parkinson's disease in one's body. XGBOOST algorithm is a technique for regression and classification problems. It produces a prediction model in form of a decision tree. Data is loaded, features and label are specified, data is split, XGBClassifier is produced and calculate the accuracy of our model.
Parkinson's disease is caused by the disruption of the brain cells that produce a substance to allow brain cells to communicate with each other, called dopamine. The cells that produce dopamine in the brain are responsible for the control, adaptation, and fluency of movements. When 60–80%of these cells are lost, then enough dopamine is not produced and Parkinson's motor symptoms appear

B. Scope

The data mining techniques must be used for accurate and precise data classification.
The algorithms which we are being used in this group project is XGBOOST algorithm. The idea is to split the whole data set into training data and testing data, and creating a classification which takes the training data and predicts the accuracy or efficiency of the model on the testing data, this will give us an absolute accuracy metric for the classification of the data set based on the ailment metric. The reference class-label that we are using for classifying the data is "class" which refers to vocal frequency. We try to create a model the yields to give a better accuracy, there must not be much difference between the training data set accuracy and testing data set accuracy, this tells us that the model performance is poor, if the accuracies are in a mere difference and the accuracy is above par, we use the enhancing algorithms like grid search and regularization, this would increase the training and testing accuracies.

C. Project Overview

The basic idea is to modulate an efficient xgboost model, that would give us satisfiable output accuracies, the model must not be over-fitted or under-fitted, both means that the model development is poor, in this project we create one model, which is "XGBOOST classifier". After developing the model, we test the accuracies based on the training data set. Then the various insights are observed and recorded.

D. Objectives

The objective is to classify whether the person will suffer from Parkinson's disease or not, there are various traits that are used in the data sets, but the traits or metrics that have the most correlation with the output metric have the more dependency, based on that idea we can know whether an individual will suffer from the ailment or not (the ailment is Parkinson's disease). Then the classification report which gives us the accuracies the frequencies of the metrics that are called and used for classification. The major objective is to show the accuracy of the model and show that the proposed model are more efficient than the existing models.

E. Data Set

Source: The dataset was created by Max Little of the University of Oxford, who recorded the speech signals. The original study published the feature extraction methods for general voice disorders.

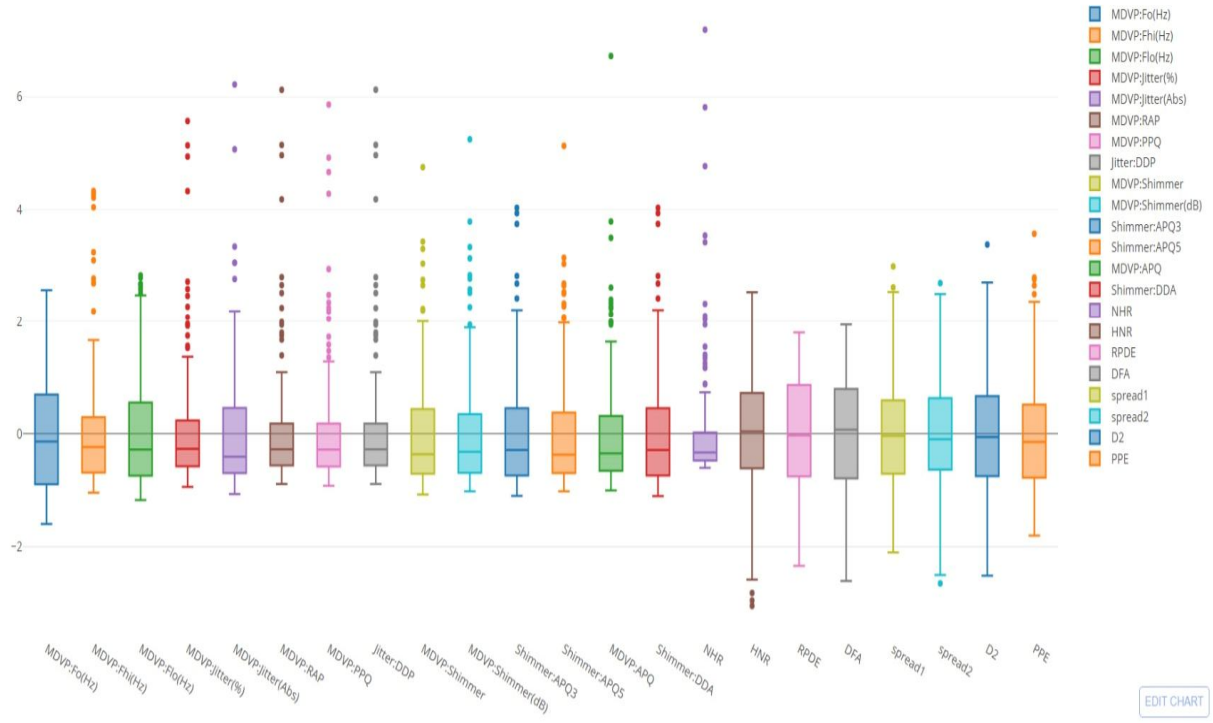


Fig (1.1)

F. Attribute Information

Matrix column entries (attributes):
name - ASCII subject name and recording number
MDVP:Fo(Hz) - Average vocal fundamental frequency
MDVP:Fhi(Hz) - Maximum vocal fundamental frequency
MDVP:Flo(Hz) - Minimum vocal fundamental frequency
MDVP:Jitter(%),MDVP:Jitter(Abs),MDVP:RAP,MDVP:PPQ,Jitter:DDP - Several measures of variation in fundamental frequency
MDVP:Shimmer,MDVP:Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5,MDVP:APQ,Shimmer:DDA - Several measures of variation in amplitude
NHR,HNR - Two measures of ratio of noise to tonal components in the voice
status - Health status of the subject (one) - Parkinson's, (zero) - healthy
RPDE,D2 - Two nonlinear dynamical complexity measures
DFA - Signal fractal scaling exponent
spread1,spread2,PPE - Three nonlinear measures of fundamental frequency variation

II. LITERATURE SURVEY

Literature survey is a piece of discursive prose, not a list describing or summarizing one piece of literature from another, it is a continuous and iterative process of gaining knowledge and insights from the information and enhancing it, the key purpose of literature survey is to address a point which has not been addressed.

A. Proposed System

It is a disease which is a disorder in the nervous system. Parkinson's disease affects the movement of the human body. In today's world, around 1 million people are suffering from this disease. This is a disorder which produces neurodegenerative dopamine-producing neurons in the brain. The following system will detect Parkinson's symptoms in the human body. Since there is no definitive laboratory test to diagnose PD, diagnosis is often difficult, particularly in the early stages when motor effects are not yet severe. This helps us detect the disease at an early stage and required treatment can be started off as soon as it is detected

B. Decision Trees

The decision tree classifier creates the classification model by building a decision tree. Each node in the tree specifies a test on an attribute, each branch descending from that node corresponds to one of the possible values for that attribute. We even use enhancing algorithms like regularizations and grid-search for the proposed system models.

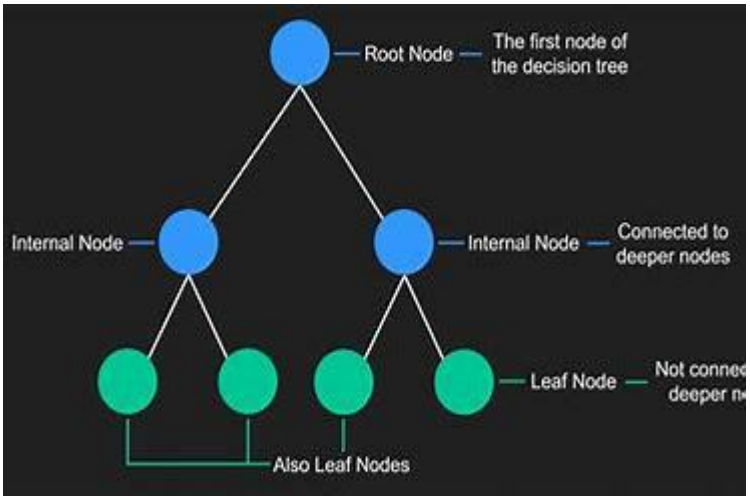


Fig (2.1)

C. Scope of the Proposed System

1) The better the classifier algorithm, the better the accuracy.

2) When the recorded accuracy is better for classification of suffering of the diabetes ailment, the people can feel safe about the data insights and believe that the data is correct and take precautionary measures.

III. SYSTEM ANALYSIS

A system is "an orderly grouping of interdependent components linked together according to a plan to achieve a specific goal". System analysis is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

Analysis specifies what the system should do.

A. Functional Requirements

Functional requirements are product features that developers must implement to enable the users to achieve their goals. They define the basic system behaviour under specific conditions, in other words they are the steps that are to be implemented to achieve our goal.

The steps or sequence of actions that are to be performed are:

1) *Data collection:* The first step is to collect the raw data from various online websites or any other desired sources.
2) *Data pre-processing:* Then we have to clean the raw data which has a lot of inconsistent value.
3) *Understand the Data:* Try to gain knowledge about the dataset and understand each and every class labels and find out the independent and dependent class labels.
4) *Split the Data:* The data must be split into:
    a. Training data
    b. Testing data

NOTE: We use 20-80 split for our data set, 80% data is the training data and 20%data is the testing data.

5) *Create the Model:* Create the machine learning models for the algorithms "Naïve Bayes", "Decision Trees" and "MLP Classifier".
6) *Record the Outputs:* Find the accuracies of the testing and training data. Also find the classification reports and confusion matrix for the testing and training data and intimate the "recall" and "precision" values.
7) *Gain Insights:* Record all the other outputs and visualization behaviours and record them.

B. Performance Requirements

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use. The requirement specification for any system can be broadly stated as given below:

1) The system should be able to interface with the existing system
2) The system should be accurate
3) The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

C. Software Requirements

1) Supported Operating System

a) Windows 7(32 or 64 bit), Windows 8(32 or 64 bit), Windows 10(32 or 64 bit)
b) Linux (Ubuntu Linux)
c) MacOS

2) Supported Development Environment

a) Jupyter Notebook (anaconda3)
b) Python 3
    >Matplotlib
    >Pandas
    >NumPy
    >Seaborn
a) sklearn

D. Hardware Requirements

The hardware requirements are as follows:

1) Processor: 1.80 GHz
2) RAM: 8 GB
3) ROM: 300 GB
4) Disk Space: 1 GB

E. Feasibility Study

A feasibility study, as the name suggests, is designed to reveal whether a project/plan is feasible. It is an assessment of the practicality of a proposed project/plan. This project is very useful in the area of the medical sciences, this will boost the medical sector, it is an analysis-based project to predict whether an individual will suffer from diabetes or not, so they can take precautionary measures to cure their disease, it is an analysis-based project because we use three machine learning algorithms to find out the accuracies and say that the proposed system models have a greater accuracy and recall than the existing system machine learning model.

IV. SYSTEM DESIGN

A. System Architecture

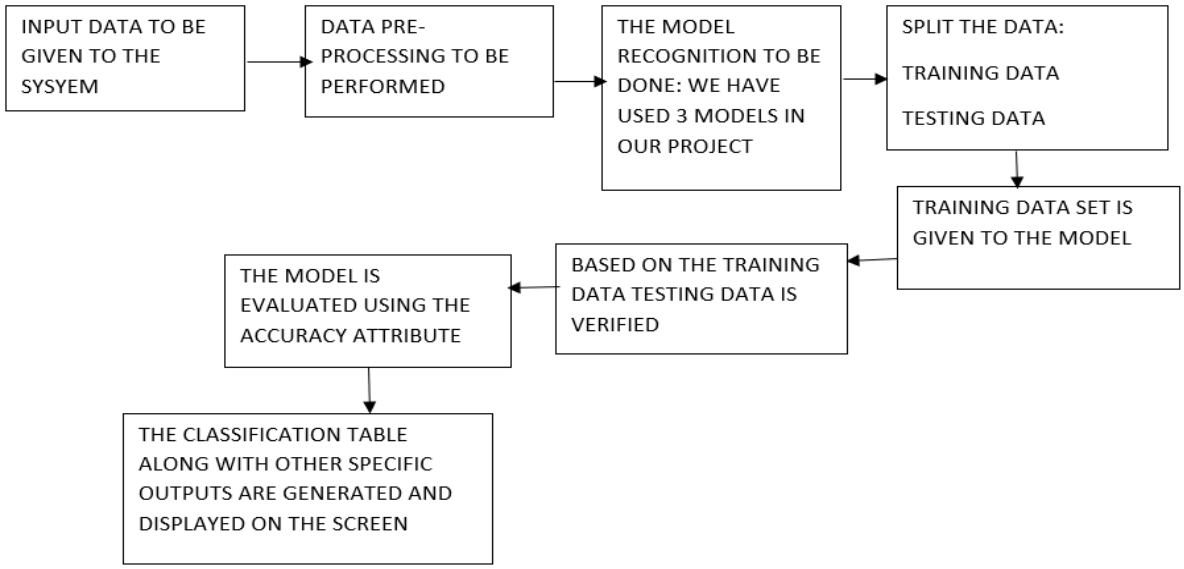


Fig (4.1)

B. Data Flow Diagram

A data flow diagram (DFD) is a visual representation of the information flow through a process or system**.** DFDs help you better understand process or system operation to discover potential problems, improve efficiency, and develop better processes.

C. UML Diagrams

UML diagrams are the ultimate output of the entire discussion. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it a complete one.

UML includes the following nine diagrams and the details are described in the following

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State-chart diagram
- Deployment diagram
- Component diagram

1) Goals

The primary goals in the design of the UML are as follows:

a) Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
b) Provide extendibility and specialization mechanisms to extend the core concepts.
c) Be independent of particular programming languages and development process.
d) Provide a formal basis for understanding the modelling language.
e) Encourage the growth of OO tools market.
f) Support higher level development concepts such as collaborations, frameworks, patterns and components.
g) Integrate best practices.

UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta- model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

2) Use Case Diagram

A use case diagram in the Unified Modelling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

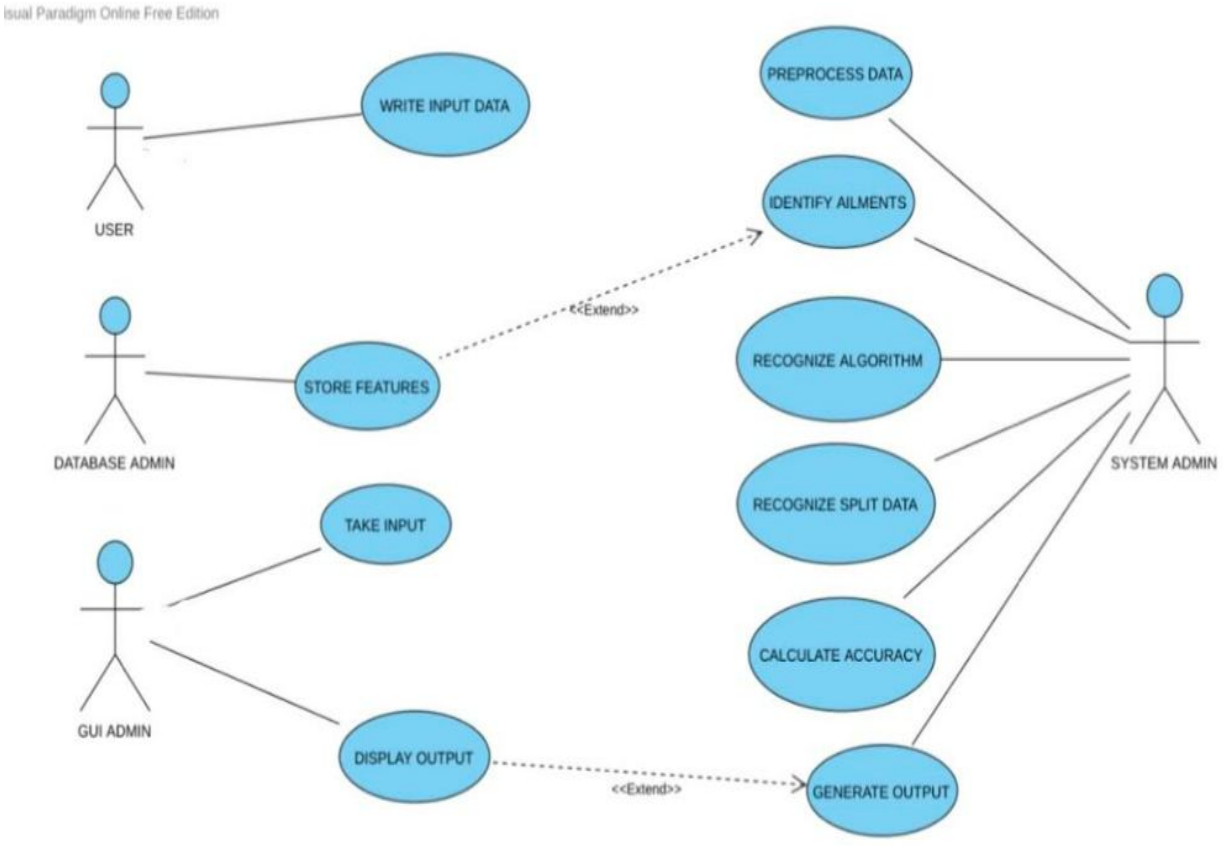


Fig (4.5)

3) Class Diagram

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

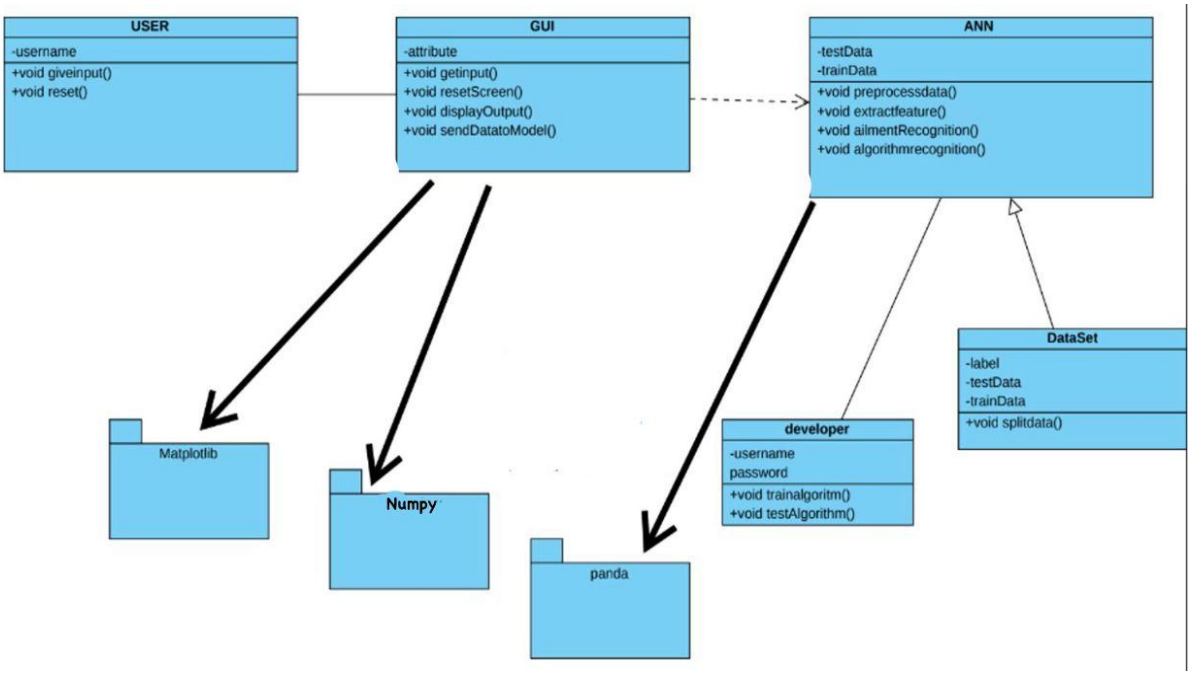


Fig (4.6)

4) Sequence Diagram

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios and timing diagrams.

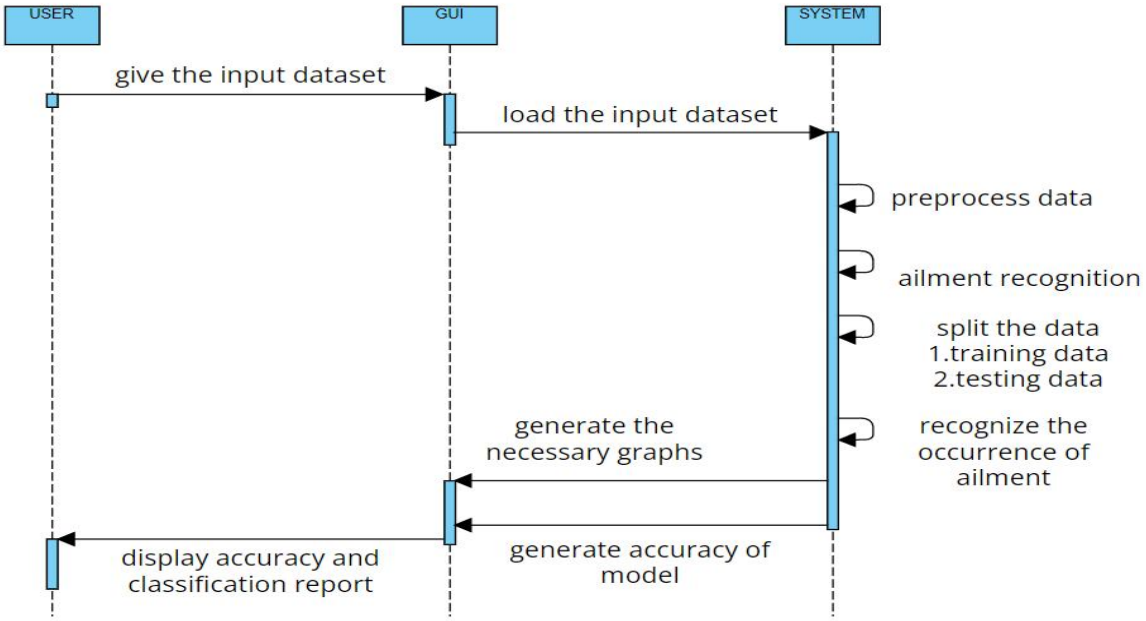


Fig (4.7)

5) Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step- by-step workflows of components in a system. An activity diagram shows the overall flow of control.

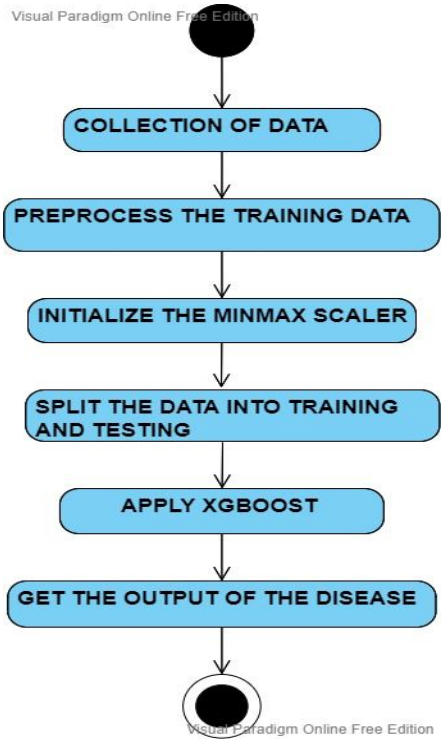


Fig (4.8)

V. IMPLEMENTATION AND RESULTS

A. Language/ Technology Used

1) Juypter Notebook (anaconda3): The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Its uses include data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more. It uses anaconda prompt for its initiation and the program is written in "PYTHON3" language.

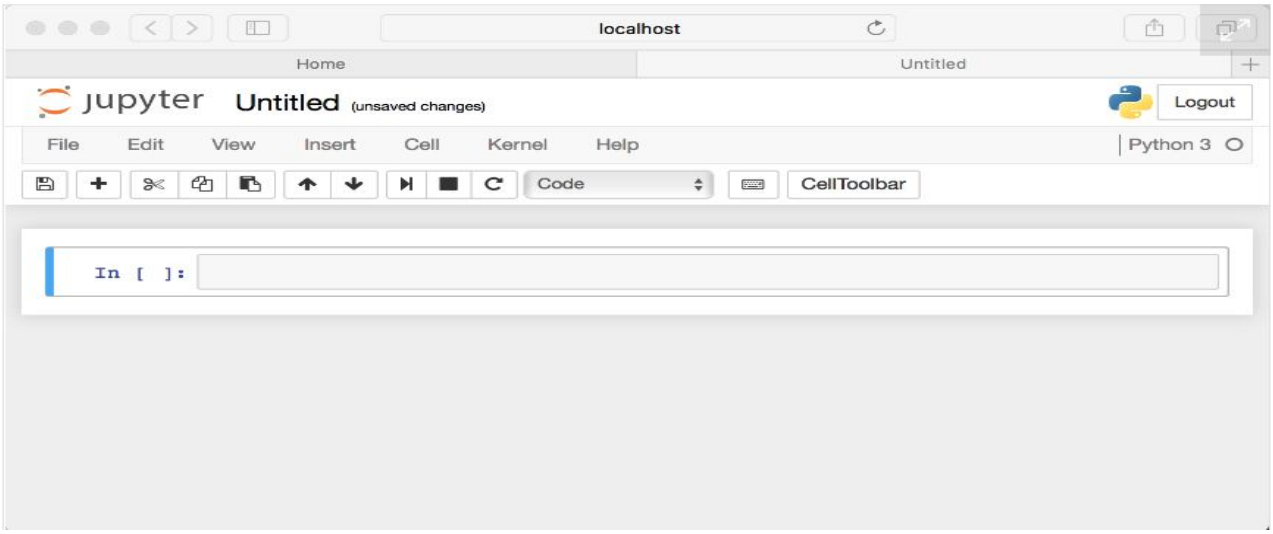


Fig (5.4)

2) *Python3:* Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.
3) *NumPy:* NumPy is the python library for linear algebra. It is an open-source python package that stands for Numerical python. NumPy supports large multidimensional arrays and matrices. It is an extension module for python. It is the free library and is used to do scientific computation tasks. The ancestor of NumPy numeric was created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing numarray into numeric with extensive modifications. NumPy has many contributions. NumPy has been built to work with N-dimensional array. It can be integrated into C/C++ and Fortran.
4) *Pandas:* The name pandas are derived from the "Panel data". It was developed by Wes McKinney. Pandas is a library written for the Python programming language for data manipulation and data analysis. It provides data structures and operations for manipulating numerical tables and time series. Pandas, library is built on the NumPy package. It provides data manipulation and analysis easy and easy to use data structures. Data frame is one of the data structures. Python along with pandas is used in various fields such as statistics, economics, analytics, etc. Pandas is the BSD-licensed python library.
5) *Matplotlib:* Matplotlib is one of the python libraries which provides functions to plot various data sets. In the year 2002, Matplotlib was developed by John Hunter. It uses NumPy to handle the large arrays of data sets. It is used for data visualization. Matplotlib is used for 2D plots of an array. The advantage of visualization is that it provides visual access to huge amounts of data. Matplotlib consists of several plots such as bar, line, histogram, etc. Plots help in understanding the patterns and is used in making correlations.
6) *Sklearn:* Scikit-Learn is a free machine learning library for Python. It supports both supervised and unsupervised machine learning, providing diverse algorithms for classification, regression, clustering, and dimensionality reduction. The library is built using many libraries you may already be familiar with, such as NumPy and SciPy. It also plays well with other libraries, such as Pandas and Seaborn.
7) *Seaborn:* Seaborn is an amazing visualization library for statistical graphics plotting in Python. It is built on the top of matplotlib library and also closely integrated into the data structures from pandas. Pandas and Seaborn is one of those packages and makes importing and analyzing data much easier. In this article, we will use Pandas and Seaborn to analyze data.

B. *Methods/ Algorithms Used*

1) *XGboost Algorithm:* XGBoost stands for "Extreme Gradient Boosting". XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements under the Gradient Boosting framework. It provides a parallel tree boosting to solve many data science problems in a fast and accurate way.
2) *Boosting:* Boosting is an ensemble learning technique to build a strong classifier from several weak classifiers in series. Boosting algorithms play a crucial role in dealing with bias-variance trade-off. Unlike bagging algorithms, which only controls for high variance in a model, boosting controls both the aspects (bias & variance) and is considered to be more effective.

Below are the few types of boosting algorithms:

a) AdaBoost (Adaptive Boosting)
b) Gradient Boosting
c) XGBoost
d) CatBoost
e) Light GBM

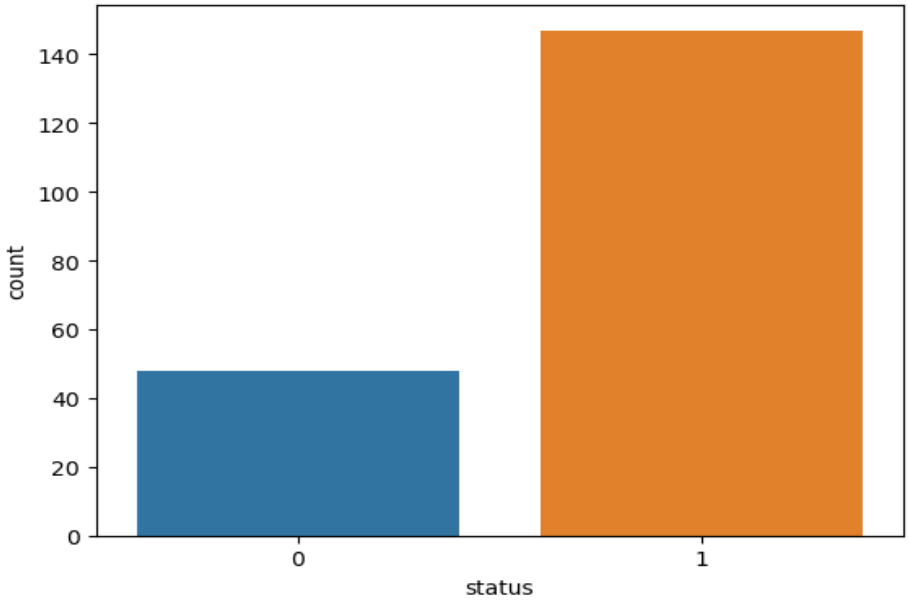
3) *XGBoost:* XGBoost stands for eXtreme Gradient Boosting. It became popular in the recent days and is dominating for structured data because of its scalability. XGBoost is an extension to gradient boosted decision trees (GBM) and specially designed to improve speed and performance.

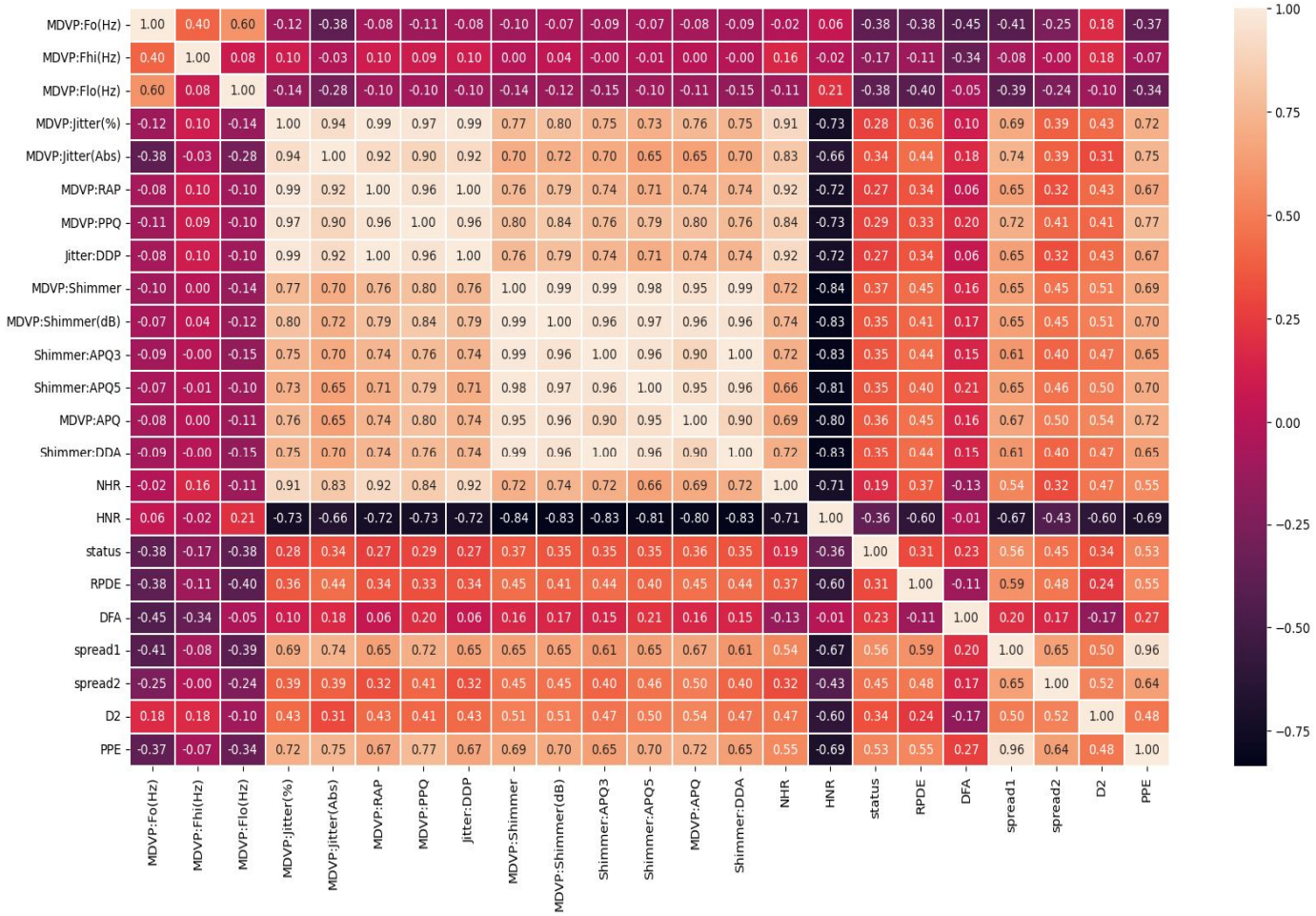
C. *Sample Code*

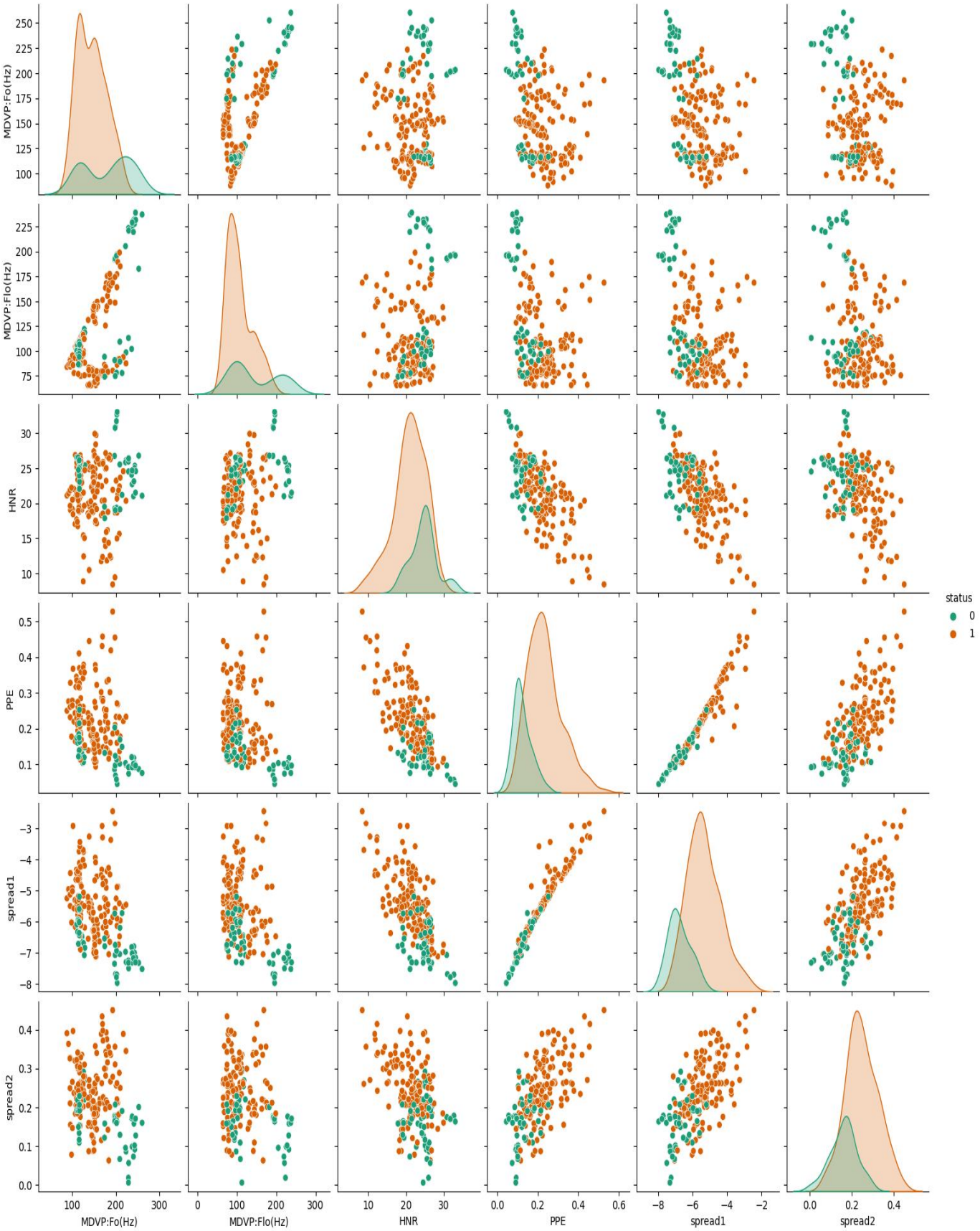
```
import numpy as np
import pandas as pd
import os
```

```
import sys
get_ipython().system('{sys.executable} -m pip install xgboost')
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import cross_val_score,cross_validate
from sklearn.model_selection import StratifiedKFold
from sklearn.preprocessing import MinMaxScaler
from xgboost import XGBClassifier
os.getcwd()
df = pd.read_csv("parkinsons.data")
df.head()
df.info()
df.shape
df.describe()
sns.countplot(df['status'])
plt.show()
plt.figure(figsize=(20,10))
sns.heatmap(df.corr(),annot=True,fmt=".2f",linewidths="1.2")
plt.show()
plt.figure(figsize = (15,10))
sns.pairplot(df, vars=['MDVP:Fo(Hz)','MDVP:Flo(Hz)','HNR','PPE','spread1','spread2'],hue='status',palette='Dark2')
plt.savefig('Relationship')
plt.show()
X = np.array(df.drop(['name','status'], axis = 1))
y = np.array(df['status'])
print(f'X shape: {X.shape} Y Shape: {y.shape}')
scaler = MinMaxScaler()
scaled_X = scaler.fit_transform(X)
def crossValidate(model):
  strat_k_fold = StratifiedKFold(n_splits = 10, shuffle = True, random_state = 42)
  scoring = ["accuracy","precision","recall","f1","roc_auc"]
  cv = cross_validate(model, scaled_X, y, cv = strat_k_fold, scoring = scoring)
  result = [round(cv[score].mean(),3) for score in cv]
  return result
  model = XGBClassifier()
result = crossValidate(model)
plt.figure(figsize = (6,2))
model_preformance = pd.Series(data=result[2:],
      index=['Accuracy','Precision','Recall','F1-Score','AUC (ROC)'])
model_preformance.sort_values().plot.barh()
plt.title('Model Performance')
```

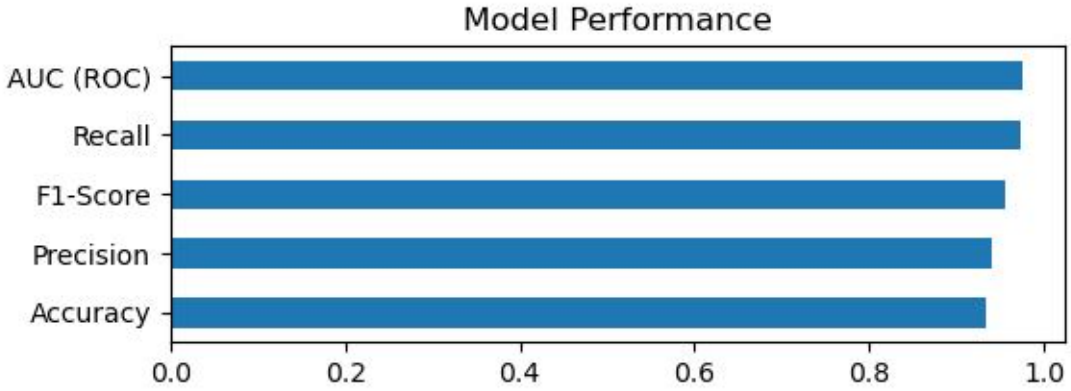
D. Output


140
120
100
80
60
40
20
0
count
0
1
status


MDVP:Fo(Hz) 1.00 0.40 0.60 -0.12 -0.38 -0.08 -0.11 -0.08 -0.10 -0.07 -0.09 -0.07 -0.08 -0.09 -0.02 0.06 -0.38 -0.38 -0.45 -0.41 -0.25 0.18 -0.37
MDVP:Fhi(Hz) 0.40 1.00 0.08 0.10 -0.03 0.10 0.09 0.10 0.00 0.04 -0.00 -0.01 0.00 -0.00 0.16 -0.02 -0.17 -0.11 -0.34 -0.08 -0.00 0.18 -0.07
MDVP:Flo(Hz) 0.60 0.08 1.00 -0.14 -0.28 -0.10 -0.10 -0.10 -0.14 -0.12 -0.15 -0.10 -0.11 -0.15 -0.11 0.21 -0.38 -0.40 -0.05 -0.39 -0.24 -0.10 -0.34
MDVP:Jitter(%) -0.12 0.10 -0.14 1.00 0.94 0.99 0.97 0.99 0.77 0.80 0.75 0.73 0.76 0.75 0.91 -0.73 0.28 0.36 0.10 0.69 0.39 0.43 0.72
MDVP:Jitter(Abs) -0.38 -0.03 -0.28 0.94 1.00 0.92 0.90 0.92 0.70 0.72 0.70 0.65 0.65 0.70 0.83 -0.66 0.34 0.44 0.18 0.74 0.39 0.31 0.75
MDVP:RAP -0.08 0.10 -0.10 0.99 0.92 1.00 0.96 1.00 0.76 0.79 0.74 0.71 0.74 0.74 0.92 -0.72 0.27 0.34 0.06 0.65 0.32 0.43 0.67
MDVP:PPQ -0.11 0.09 -0.10 0.97 0.90 0.96 1.00 0.96 0.80 0.84 0.76 0.79 0.80 0.76 0.84 -0.73 0.29 0.33 0.20 0.72 0.41 0.41 0.77
Jitter:DDP -0.08 0.10 -0.10 0.99 0.92 1.00 0.96 1.00 0.76 0.79 0.74 0.71 0.74 0.74 0.92 -0.72 0.27 0.34 0.06 0.65 0.32 0.43 0.67
MDVP:Shimmer -0.10 0.00 -0.14 0.77 0.70 0.76 0.80 0.76 1.00 0.99 0.99 0.98 0.95 0.99 0.72 -0.84 0.37 0.45 0.16 0.65 0.45 0.51 0.69
MDVP:Shimmer(dB) -0.07 0.04 -0.12 0.80 0.72 0.79 0.84 0.79 0.99 1.00 0.96 0.97 0.96 0.96 0.74 -0.83 0.35 0.41 0.17 0.65 0.45 0.51 0.70
Shimmer:APQ3 -0.09 -0.00 -0.15 0.75 0.70 0.74 0.76 0.74 0.99 0.96 1.00 0.96 0.90 1.00 0.72 -0.83 0.35 0.44 0.15 0.61 0.40 0.47 0.65
Shimmer:APQ5 -0.07 -0.01 -0.10 0.73 0.65 0.71 0.79 0.71 0.98 0.97 0.96 1.00 0.95 0.96 0.66 -0.81 0.35 0.40 0.21 0.65 0.46 0.50 0.70
MDVP:APQ -0.08 0.00 -0.11 0.76 0.65 0.74 0.80 0.74 0.95 0.96 0.90 0.95 1.00 0.90 0.69 -0.80 0.36 0.45 0.16 0.67 0.50 0.54 0.72
Shimmer:DDA -0.09 -0.00 -0.15 0.75 0.70 0.74 0.76 0.74 0.99 0.96 1.00 0.96 0.90 1.00 0.72 -0.83 0.35 0.44 0.15 0.61 0.40 0.47 0.65
NHR -0.02 0.16 -0.11 0.91 0.83 0.92 0.84 0.92 0.72 0.74 0.72 0.66 0.69 0.72 1.00 -0.71 0.19 0.37 -0.13 0.54 0.32 0.47 0.55
HNR 0.06 -0.02 0.21 -0.73 -0.66 -0.72 -0.73 -0.72 -0.84 -0.83 -0.83 -0.81 -0.80 -0.83 -0.71 1.00 -0.36 -0.60 -0.01 -0.67 -0.43 -0.60 -0.69
status -0.38 -0.17 -0.38 0.28 0.34 0.27 0.29 0.27 0.37 0.35 0.35 0.35 0.36 0.35 0.19 -0.36 1.00 0.31 0.23 0.56 0.45 0.34 0.53
RPDE -0.38 -0.11 -0.40 0.36 0.44 0.34 0.33 0.34 0.45 0.41 0.44 0.40 0.45 0.44 0.37 -0.60 0.31 1.00 -0.11 0.59 0.48 0.24 0.55
DFA -0.45 -0.34 -0.05 0.10 0.18 0.06 0.20 0.06 0.16 0.17 0.15 0.21 0.16 0.15 -0.13 -0.01 0.23 -0.11 1.00 0.20 0.17 -0.17 0.27
spread1 -0.41 -0.08 -0.39 0.69 0.74 0.65 0.72 0.65 0.65 0.65 0.61 0.65 0.67 0.61 0.54 -0.67 0.56 0.59 0.20 1.00 0.65 0.50 0.96
spread2 -0.25 -0.00 -0.24 0.39 0.39 0.32 0.41 0.32 0.45 0.45 0.40 0.46 0.50 0.40 0.32 -0.43 0.45 0.48 0.17 0.65 1.00 0.52 0.64
D2 0.18 0.18 -0.10 0.43 0.31 0.43 0.41 0.43 0.51 0.51 0.47 0.50 0.54 0.47 0.47 -0.60 0.34 0.24 -0.17 0.50 0.52 1.00 0.48
PPE -0.37 -0.07 -0.34 0.72 0.75 0.67 0.77 0.67 0.69 0.70 0.65 0.70 0.72 0.65 0.55 -0.69 0.53 0.55 0.27 0.96 0.64 0.48 1.00
MDVP:Fo(Hz)
MDVP:Fhi(Hz)
MDVP:Flo(Hz)
MDVP:Jitter(%)
MDVP:Jitter(Abs)
MDVP:RAP
MDVP:PPQ
Jitter:DDP
MDVP:Shimmer
MDVP:Shimmer(dB)
Shimmer:APQ3
Shimmer:APQ5
MDVP:APQ
Shimmer:DDA
NHR
HNR
status
RPDE
DFA
spread1
spread2
D2
PPE
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75


MDVP:Fo(Hz)
MDVP:Flo(Hz)
HNR
PPE
spread1
spread2
status
0
1

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XII Dec 2022- Available at www.ijraset.com



VI. TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail unacceptably. There are various types of tests. Each test type addresses a specific testing requirement.

A. Types Of Testing

1) Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at the component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

2) Integration Testing

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

3) Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:
Valid Input: Identified classes of valid input must be accepted.
Invalid Input: Identified classes of invalid input must be rejected.
Functions: Identified functions must be exercised.
Output: Identified classes of application outputs must be exercised.
Systems/Procedures: Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests are focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

4) System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5) White Box Testing

White Box Testing is a testing in which the software tester knows the inner workings, structure and language of the software, or at least its purpose. It is a purpose. It is used to test areas that cannot be reached from a black-box level.

6) Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, like most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works

VII. CONCLUSION

This entire project has been developed and deployed as per the requirements stated, it is found to be bug free as per the tested standards that are implemented. Any specification untraced errors will be concentrated in the coming versions, which are planned to be developed. It is implemented using python and it has been designed to cater the document safety needs. The conclusions that we have drawn from this project are:

1) Based on the correlation between the class labels the chance of suffering of a person from diabetes is predicted, the metric or classification is done based on this relation
2) When we have a close look at the heat map, the correlation values are shown; the darker the shade the less is the correlation.
3) The same class labels like "Preg", "Plas", "Pres", "skin" have completely positive correlation between them which is 1.
4) Some class label pairs like "Preg-age", "class-Plas", "skin-test" have a positive correlation them.
5) Based in the correlation value of the attributes between the independent class labels (Preg, Plas, Pres, skin, test, mass, pedi, age) and the dependent class label(class) the possibility of occurrence of the ailment diabetes is predicted.
6) Therefore, the attributes that are most likely to cause diabetes are "Plas" and "mass".
7) The output is shown by the testing accuracies which depict the efficiency of the model.

Based on the above observations, we have implemented two algorithms which we have mentioned earlier, and the most adequate algorithm along with the accuracies are shown,

"Acc (MLP CLASSIFIER)" > "Acc (DECISION TREES)" >Acc (NAÏVE BAYES)"

VIII. FUTURE SCOPE

A constant form of silent evolution is machine learning. We thought computers were the big all-that that would allow us to work more efficiently; soon, machine learning was introduced to the picture, changing the discourse of our lives forever. The reshaping of the world started with teaching computers to do things for us, and now it has reached the stage where even that simple step is eliminated. It is no longer imperative for us to teach computers how to execute complex tasks like text translation or image recognition: instead, we built systems that let them do it themselves. It's as close to magic as the muggle community will ever reach!

So, in general terms, machine learning is a result of the application of Artificial Learning. Let's take the example of you shopping online — have you ever been in a situation where the app or website started recommending products that might in some way be associated or similar to the purchase you made? If yes, then you have seen machine learning in action. Even the "bought together" combination of products is another by-product of machine learning.

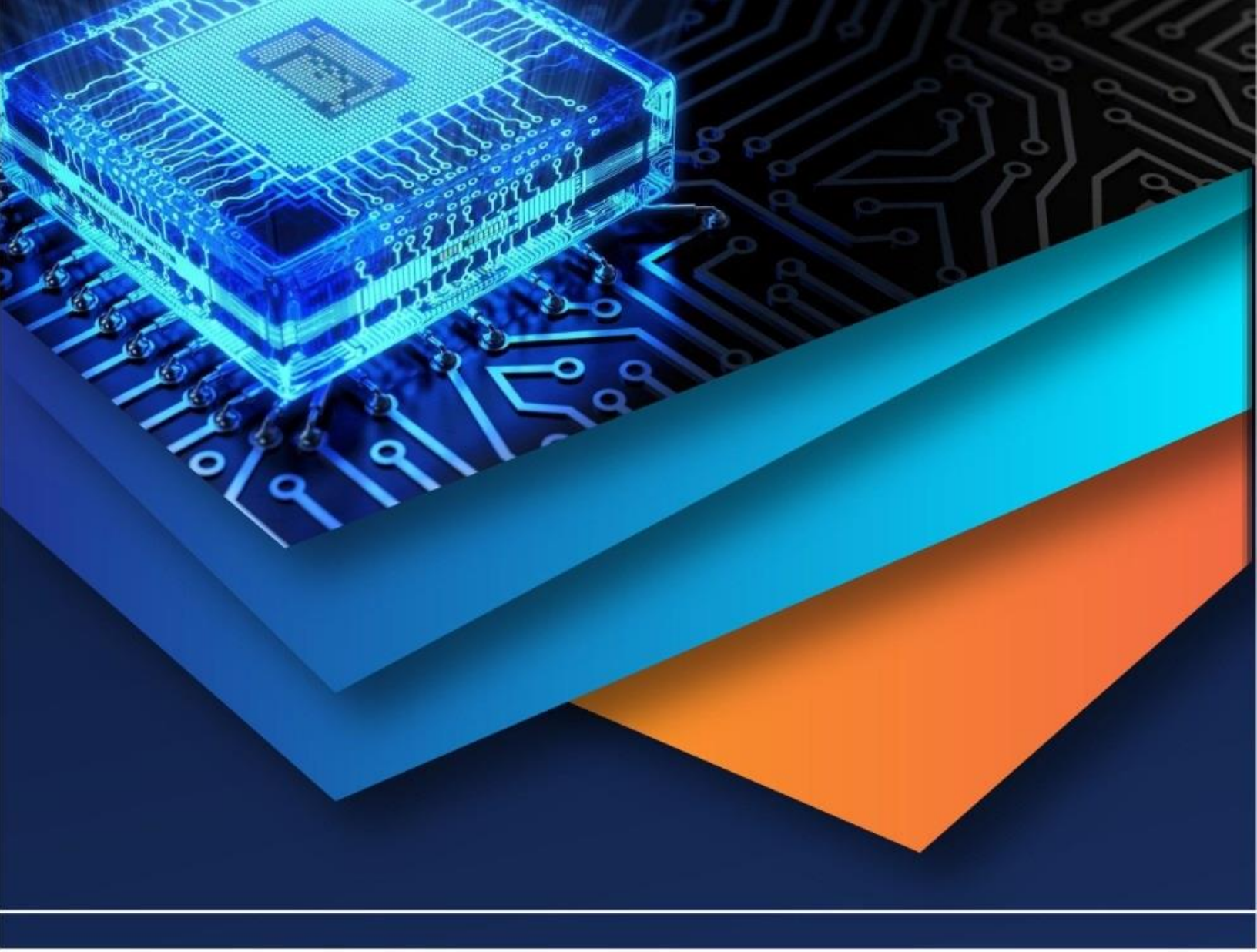
This is how companies target their audience, and divide people into various categories to serve them better, make their shopping experience tailored to their browsing behaviour.

Machine learning is merely based on predictions made based on experience. It enables machines to make data-driven decisions, which is more efficient than explicitly programming to carry out certain tasks. These algorithms are designed in a fashion that gives exposure to new data that can help organisations learn and improve their strategies.

BIBLIOGRAPHY

The references that I used for this project are:

[1] https://ieeexplore.ieee.org/document/9071471
[2] https://ieeexplore.ieee.org/document/8391453
[3] https://www.geeksforgeeks.org/machine-learning-with-python/
[4] https://www.mygreatlearning.com/blog/most-used-machine-learning-algorithms-in-python/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY