# MACHINE LEARNING

(​Predicting of GENDER RECOGNITION BY VOICE)  Summer Internship Report Submitted in partial fulfilment​ of the requirement for undergraduate degree of

**Bachelor of Technology**​

​ In​

**Computer Science Engineering**​

​ By​

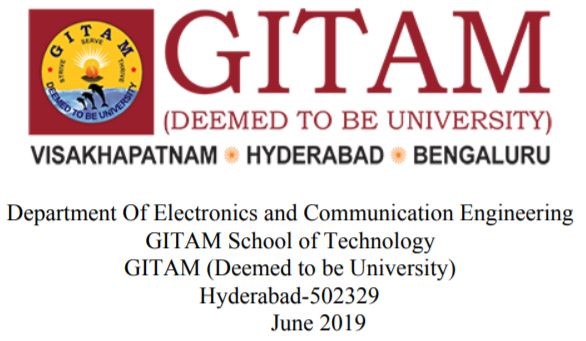
**G Nikhila**

**221710315014**

​ ​Under the Guidance of

**Mr…………………..**

Assistant Professor​



i​

# DECLARATION

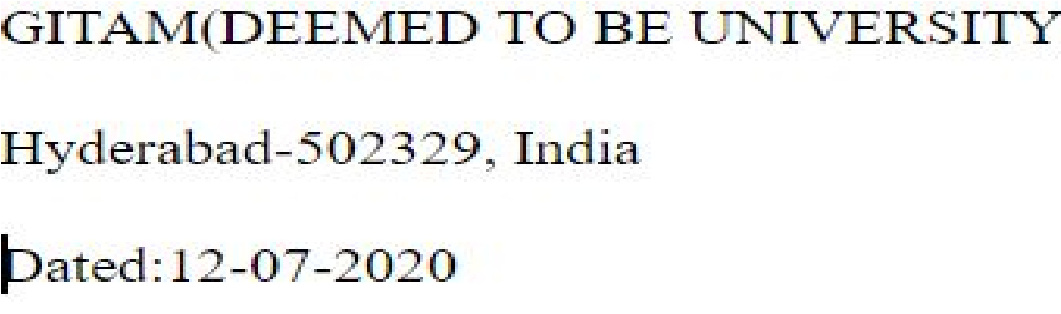
I submit this industrial training work entitled ​**“ GENDER RECOGNITION​ BY VOICE​”** ​to GITAM (Deemed To Be University) , Hyderabad in partial fulfilment of the requirements for the award of the degree of ​ **“Bachelor of Technology ”**​ in ​**“ Computer Science Engineering ”**​. I declare that it was carried out independently by me under the guidance of ​**Mr……………..**​, Asst. Professor, GITAM (Deemed To Be University), Hyderabad, India.

The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Place: HYDERABAD G Nikhila

Date:12-07-2020 221710315014

ii​

## ​CERTIFICATE

This is to certify that the Industrial Training Report entitled **“GENDER**​ **RECOGNITION BY VOICE ”** is being submitted by G Nikhila(221710315014)​ in partial fulfilment of the requirement for the award of **Bachelor of Technology in Computer**​ **Science Engineering** at GITAM (Deemed To Be University), Hyderabad during the academic​ year 2020-21

It is faithful record work carried out by her at the **Computer Science**​

**Engineering Department,** GITAM University Hyderabad Campus under my guidance and​ supervision.

### Mr……………………. ​ Dr.S.Phani Kumar​

Assistant Professor Professor and HOD

Department of CSE Department of CSE

**ACKNOWLEDGEMENT**

Apart from my effort, the success of this internship largely depends on the encouragement and guidance of many others. I take this opportunity to express my gratitude to the people who have helped me in the successful competition of this internship.

I would like to thank respected **Dr. N. Siva Prasad**​ , Pro Vice Chancellor, GITAM​ Hyderabad and **Dr. CH. Sanjay**​, Principal, GITAM Hyderabad​.

I would like to thank respected **Dr. S.Phani Kumar**​ , Head of the Department of​ Computer Science Engineering for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present an internship report. It helped me a lot to realize what we study for.

I would like to thank the respected faculties **Mr. ………………..**​ who helped me to​ make this internship a successful accomplishment.

I would also like to thank my friends who helped me to make my work more organized and well-stacked till the end.

G Nikhila​

221710315014

# ABSTRACT

Machine learning algorithms are used to predict the values from the​ data set by splitting the data set in to train and test and building Machine learning algorithms models of higher accuracy to predict the values is the primary task to be performed on the Train and Test data set. My perception of understanding the given data sets has been in the view of undertaking different activities performed by 30 subjects and prediction of the activity.

To get a better understanding and work on a strategic approach for solution of the client, I have adapted to look at the activities performed by plotting few graphs to understand it in a better way,for further deep understanding of the problem, I have taken the describe function to know more about my data set , and my primary objective of this case study was to predict the activity performed with at most accuracy,so i have used several algorithms to know which one yields the highest accuracy to predict my outcome.

**Table of Contents:**

**LIST OF FIGURES**

## CHAPTER 1: MACHINE LEARNING ………………………………………………...11

1.1 INTRODUCTION…………………………………………………………11

1.2 IMPORTANCE OF MACHINE LEARNING…………………………….11

1.3 USES OF MACHINE LEARNING……………………………………….12

1.4 TYPES OF LEARNING ALGORITHMS………………………………...13

1.4.1 Supervised Learning……………………………………………. 13

1.4.2 Unsupervised Learning………………………………………….14

1.4.3 Semi Supervised Learning…………………… …………………15

1.5 RELATION BETWEEN DATA MINING, MACHINE LEARNING AND

DEEP LEARNING……………………………………………………………………………16

## CHAPTER 2: PYTHON……...……………………………………………….…………….17

2.1 INTRODUCTOIN TO PYTHON…………………………………….……17

2.2 HISTORY OF PYTHON……………………………………………….….17

2.3 FEATURES OF PYTHON………………………………………………...17

2.4 HOW TO SETUP PYTHON……………………………………………....18

2.4.1 Installation (using python IDLE) ……………………………….18

2.4.2 Installation (using Anaconda) …………………………………..19

2.5 PYTHON VARIABLE TYPES…………………………………………….21

2.5.1 Python Numbers……………………………………………….. 22

2.5.2 Python Strings…………………………………………………..22

2.5.3 Python Lists…………………………………………………….23

2.5.4 Python Tuples……………………………………………………23

2.5.5 Python Dictionary……………………………………………......24

2.6 PYTHON FUNCTION……………………………….………………............25

2.6.1 Defining a Function……………………………………………...25

2.6.2 Calling a Function………………………………………………..26

2.7 PYTHON USING OOP’s CONCEPTS………….…………………………...26

2.7.1 Class……………………………………………………………...26

2.7.2 \_init\_ method in class………………………………………......27

## CHAPTER 3: CASE STUDY . .………..………………………………………………….28

3.1​ PROBLEM STATEMENT…………………………………………………..28

3.2 DATA SET………………………………………………………..………......28 3.3 OBJECTIVE OF THE CASE STUDY……..…………………………………………....29

## CHAPTER 4: MODEL BUILDING …..…………………………………………….……30

4.1​ PREPROCESSING OF THE DATA…………………….……………………30

4.1.1 Getting the Data Set………………………………………………...30

4.1.2 Importing the Libraries……………………………………………..30

4.1.3 Importing the Data-Set……………………………………………...31

4.1.4 Handling the Missing values………………………………………..34

4.1.5 PROCESSING OF DATASET……………………………………..36

4.2 TRAINING THE MODEL…………………....……………………………...38

4.2.1 PREPROCESSING USING LABELENCODER AND NORMALIZATION ………………………………………………………………………...38

4.2.2 Visualize the correlation……………………………………….39

4.2.3 Visualization…………………………………………………...41

4.2.4 SPLITTING OF DATA ……………………………………….43

4.3 APPLYING THE ALGORITHMS ………………………………………...............46

4.3.1 K-NEAREST-NEIGHBORS(KNN) ……….....…….…………...46

4.3.2 NAÏVE BAYES………………………………………………….48

4.3.3 RANDOM FOREST …………………………………………….49

4.3.4 XGBOOST ………………………………………………………51

4.4 EVALUATING THE MODEL ……………………………………………………..52

4.4.1 LABEL ENCODER………………………………………………52

4.4.2 IMPORTING LIBRARIES……………………………………….52

4.4.3 READING THE DATASET……………………………………...53

4.4.4 HANDLING MISSING VALUES ……………………………….53

4.4.5 VIZUALIZE .THE CORRELATION…………………………….55

4.4.6 VIZUALIZE THE DATA………………………………………...56

4.4.7 SPLITTING DATA……………………………………………….57

4.4.8 CLASSIFICATION REPORT ……………………………………60

4.4.9 RANDOM FOREST………………………………………………60

CONCLUSION………………………………………………………………..................62

**LIST OF FIGURES:**

Figure 1 : The Process Flow……………………..……………………………….……...12

Figure 2 : Unsupervised Learning………………………..………………………………15

Figure 3 : Semi Supervised Learning……………………………………..……………...16

Figure 4 : Python download…………………….…....…………………………………...19

Figure 5 : Anaconda download…………....…….………………………………………..21

Figure 6 : Jupyter notebook……………………….……………………………………...21

Figure 7 : Defining a Class………………………..……………………...……………...27

Figure 8: How it Works ………………………………………………………………….30

Figure 9: Importing Libraries…………………………………………………………….31

Figure 10: Reading the dataset……………………………………………………………32

Figure 11: checking the dataset information……………………………………………...33

Figure 12:describing the dataset…………………………………………………………..33

Figure 13: defining correleation…………………………………………………………..34

Figure 14: Handling missing values………………………………………………………35

Figure 15: print null values……………………………………………………………….35

Figure 16: printing columns ……………………………………………………………...35

Figure 17: processing the dataset………………………………………………………....37

Figure 18: column datatype…………………………………………………………….....37

Figure 19: describe the data……………………………………………………………....37

Figure 20: normalization ………………………………………………………………...38

Figure 21: preprocessing using label encoder……………………………………………39

Figure 22: visualization …………………………………………………………………40

Figure 23:Box plot ………………………………………………………………………41

Figure 24:visualization…………………………………………………………………..42

Figure 25: splitting the data……………………………………………………………...43

Figure 26 : test and train data…………………………………………………………....44

Figure 27: classify model………………………………………………………………...45

Figure 28: defining the knn model……………………………………………………….46

Figure 29: knn error vs k value graph……………………………………………………47

Figure 30:defining naïve bayes model…………………………………………………...48

Figure 31:naïve bayes accuracy………………………………………………………….48

Figure 32: defining the random forest model…………………………………………….49

Figure 33: ensemble error rate vs k value………………………………………………...50

Figure 34: import xgboost ………………………………………………………………..51

Figure 35: xgboost error rate vs k value…………………………………………………..51

Figure 36: label encoder and importing libraries…………………………………………52

Figure 37: reading data set……………………………………………………………….53

Figure 38: handling and print null values………………………………………………..53

Figure 39: data types……………………………………………………………………...54

Figure 40: vizualize the correlation……………………………………………………….55

Figure 41:bar graph……………………………………………………………………….55

Figure 42:visualization…………………………………………………………………...56

Figure 43: splitting the data as train and test……………………………………………..57

Figure 44:printing training and testing data……………………………………………...59

Figure 45: classification report…………………………………………………………..60

Figure 46: code for random forest……………………………………………………….61

Figure 47: ensemble graph-error rate vs k value…………………………………...........61

**CHAPTER 1**

**MACHINE LEARNING**

# 1.1​ INTRODUCTION​ :​

Machine Learning(ML) is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of Artificial Intelligence(AI).

# 1.2​ IMPORTANCE OF MACHINE LEARNING:​

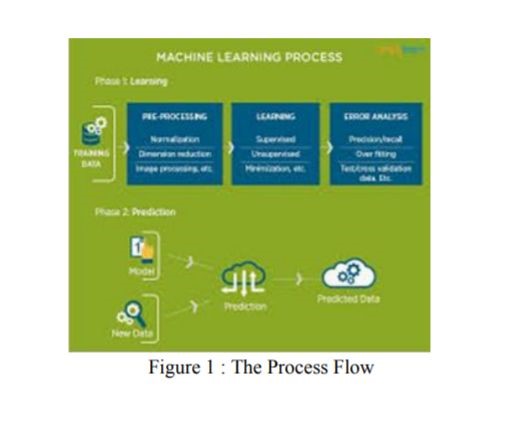
​Consider some of the instances where machine learning is applied: the self-driving Google car, cyber fraud detection, online recommendation engines—like friend suggestions on Facebook, Netflix showcasing the movies and shows you might like, and “more items to consider” and “get yourself a little something” on Amazon—are all examples of applied machine learning. All these examples echo the vital role machine learning has begun to take in today’s data-rich world.

Machines can aid in filtering useful pieces of information that help in major advancements, and we are already seeing how this technology is being implemented in a wide variety of industries.

With the constant evolution of the field, there has been a subsequent rise in the uses, demands, and importance of machine learning. Big data has become quite a buzzword in the last few years; that’s in part due to increased sophistication of machine learning, which helps analyze those big chunks of big data.

Machine learning has also changed the way data extraction, and interpretation is done by involving automatic sets of generic methods that have replaced traditional statistical techniques.

The process flow depicted here represents how machine learning works



# 1.3​ USES OF MACHINE LEARNING:​

Earlier in this article, we mentioned some applications of machine learning. To understand the concept of machine learning better, let’s consider some more examples: web search results, real-time ads on web pages and mobile devices, email spam filtering, network intrusion detection, and pattern and image recognition. All these are by-products of applying machine learning to analyze huge volumes of data

Traditionally, data analysis was always being characterized by trial and error, an approach that becomes impossible when data sets are large and heterogeneous. Machine learning comes as the solution to all this chaos by proposing clever alternatives to analyzing huge volumes of data.

By developing fast and efficient algorithms and data-driven models for real-time processing of data, machine learning can produce accurate results and analysis

# 1.4​ TYPES OF LEARNING ALGORITHMS:​

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

**1.4.1**​ **Supervised Learning .**​

When an algorithm learns from example data and associated target

responses that can consist of numeric values or string labels, such as classes or tags, in

order to later predict the correct response when posed with new examples comes under the category of supervised learning.

Supervised machine learning algorithms uncover insights, patterns, and

relationships from a labelled training dataset – that is, a dataset that already contains a known value for the target variable for each record. Because you provide the machine learning algorithm with the correct answers for a problem during training, it is able to “learn” how the rest of the features relate to the target, enabling you to uncover insights and make predictions about future outcomes based on historical data.

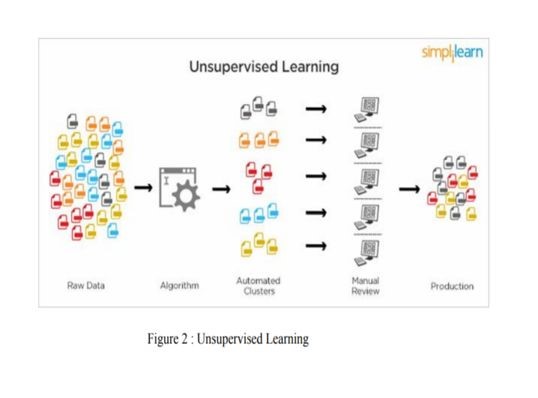
Examples of Supervised Machine Learning Techniques are Regression, in which the algorithm returns a numerical target for each example, such as how much revenue will be generated from a new marketing campaign.

Classification, in which the algorithm attempts to label each example by choosing between two or more different classes. Choosing between two classes is called binary classification, such as determining whether or not someone will default on a loan. Choosing between more than two classes is referred to as multiclass classification.

**1.4.2**​ **Unsupervised Learning:**​

When an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own.

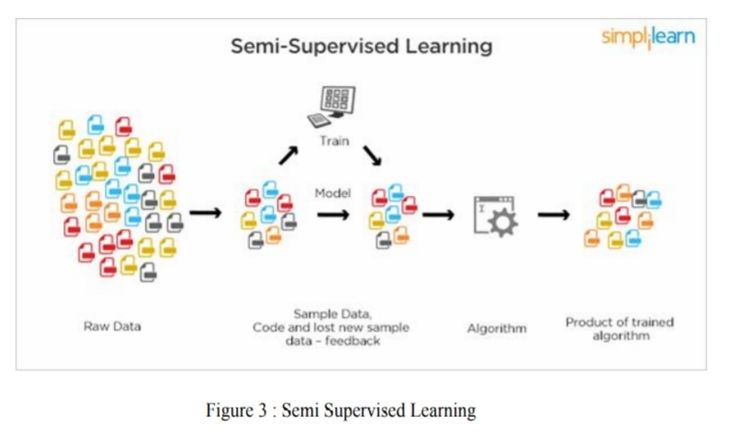
This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of uncorrelated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms



Popular techniques where unsupervised learning is used also include self-​ organizing maps, nearest neighbor mapping, singular value decomposition, and k-means clustering. Basically, online recommendations, identification of data outliers, and segment text topics are all examples of unsupervised learning.

**1.4.3**​ **Semi Supervised Learning:**​

As the name suggests, semi-supervised learning is a bit of both supervised and unsupervised learning and uses both labelled and un labelled data for training. In a typical scenario, the algorithm would use a small amount of labelled data with a large amount of un labelled data.



**1.5**​ **RELATION BETWEEN DATA MINING,MACHINE**​  **LEARNING AND DEEP LEARNING:**

Machine learning and data mining use the same algorithms and techniques as data mining, except the kinds of predictions vary. While data mining discovers previously unknown patterns and knowledge, machine learning reproduces known patterns and knowledge—and further automatically applies that information to data, decision-making, and actions.

Deep learning, on the other hand, uses advanced computing power and special 5 types of neural networks and applies them to large amounts of data to learn, understand, and identify complicated patterns. Automatic language translation and medical diagnoses are examples of deep learning.

**CHAPTER 2**

# PYTHON

Basic programming language used for machine learning is : PYTHON

**2.1 INTRODUCTION TO PYHTON:**

* Python is a high-level, interpreted, interactive and object-oriented scripting language.
* Python is a general purpose programming language that is often applied in scripting roles
* Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.
* Python is Interactive: You can sit at a Python prompt and interact with the interpreter directly to write your programs.
* Python is Object-Oriented: Python supports the Object-Oriented style or technique of programming that encapsulates code within objects.

**2.2 HISTORY OF PYTHON:**

* Python was developed by GUIDO VAN ROSSUM in early 1990’s
* Its latest version is 3.7 , it is generally called as python3

**2.3**​  **FEATURES OF PYTHON:**

* Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax, This allows the student to pick up the language quickly.
* Easy-to-read: Python code is more clearly defined and visible to the eyes.
* Easy-to-maintain: Python's source code is fairly easy-to-maintaining.
* A broad standard library: Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* Databases: Python provides interfaces to all major commercial databases.
* GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

**2.4 HOW TO SETUP PYTHON:**

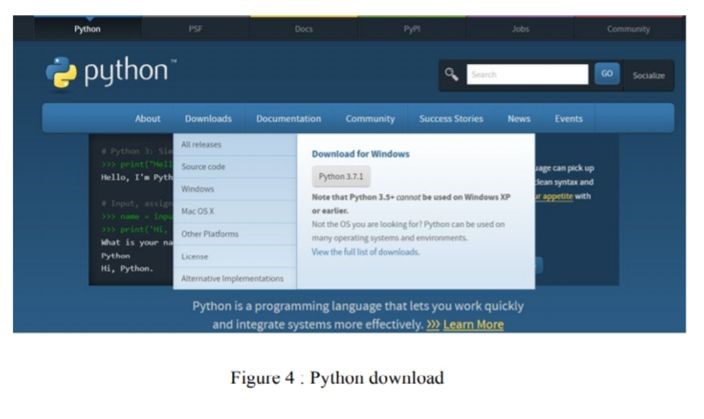
* Python is available on a wide variety of platforms including Linux and Mac OS X.

Let's understand how to set up our Python environment.

* The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python.

**2.4.1 Installation(using python IDLE):**

* Installing python is generally easy, and nowadays many Linux and Mac OS distributions include a recent python.
* Download python from www.python.org
* When the download is completed, double click the file and follow the instructions to install it.
* When python is installed, a program called IDLE is also installed along with it. It provides a graphical user interface to work with python.



**2.4.2 Installation(using Anaconda):**

* Python programs are also executed using Anaconda.​

* Anaconda is a free open source distribution of python for large scale data processing, predictive analytics and scientific computing.

* Conda is a package manager quickly installs and manages packages.

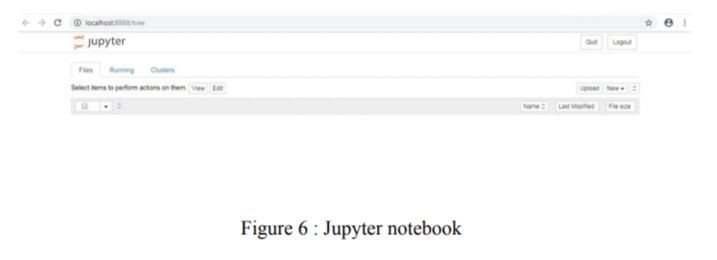
* In WINDOWS:

* In window:
* Step 1: Open Anaconda.com/downloads in web browser.
* Step 2: Download python 3.4 version for (32-bitgraphic installer/64 -bit graphic installer)
* Step 3: select installation type( all users)
* Step 4: Select path(i.e. add anaconda to path & register anaconda as

default python 3.4) next click install and next click finish

* Step 5: Open jupyter notebook ( it opens in default browser)





**2.5 PYTHON VARIABLE TYPES:**

* Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

* Variables are nothing but reserved memory locations to store values.

* Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory.

* Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable.

* Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

* Python has five standard data types –

* + - * + Number. o Strings o Lists o Tuples
        + Dictionary

**2.5.1**​  **Python Numbers:**

* Number data types store numeric values. Number objects are created when you assign a value to them.

* Python supports four different numerical types − int (signed integers) long (long integers, they can also be represented in octal and hexadecimal) float (floating point real values) complex (complex numbers).

**2.5.2 Python Strings:**

* Strings in Python are identified as a contiguous set of characters represented in the quotation marks.

* Python allows for either pairs of single or double quotes.

* Subsets of strings can be taken using the slice operator ([ ] and [:] ) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

* The plus (+) sign is the string concatenation operator and the asterisk (\*) is the repetition operator.

**2.5.3**​  **Python Lists:**

* + - Lists are the most versatile of Python's compound data types.

* + - A list contains items separated by commas and enclosed within square brackets

([]).

* + - To some extent, lists are similar to arrays in C. One difference between them is

that all the items belonging to a list can be of different data type.

* + - The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1.

* + - The plus (+) sign is the list concatenation operator, and the asterisk (\*) is the repetition operator**.**​

**2.5.4 Python Tuples:**

* A tuple is another sequence data type that is similar to the list.

* A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

* The main differences between lists and tuples are: Lists are enclosed in brackets ( [ ] ) and their elements and size can be changed, while tuples are enclosed in parentheses ( ( ) ) and cannot be updated.

* Tuples can be thought of as read-only lists.

* For example − Tuples are fixed size in nature whereas lists are dynamic. In other words, a tuple is immutable whereas a list is mutable. You can't add elements to a tuple. Tuples have no append or extend method. You can't remove elements from a tuple. Tuples have no remove or pop method.

**2.5.5**​  **Python Dictionary:**

* Python's dictionaries are kind of hash table type. They work like associative arrays 12 or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

* Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).

* You can use numbers to "index" into a list, meaning you can use numbers to find out what's in lists. You should know this about lists by now, but make sure you understand that you can only use numbers to get items out of a list.

* What a dict does is let you use anything, not just numbers. Yes, a dict associates one thing to another, no matter what it is.

**2.6**​  **PYTHON FUNCTION:**

**2.6.1**​  **Defining a Function:**

You can define functions to provide the required functionality. Here are simple rules to define a function in Python. Function blocks begin with the keyword def followed by the function name and parentheses (i.e.()).

Any input parameters or arguments should be placed within these parentheses.

You can also define parameters inside these parentheses.

The code block within every function starts with a colon (:) and is indented. The statement returns [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

**2.6.2 Calling a Function:**

Defining a function only gives it a name, specifies the parameters that are to be included in the function and structures the blocks of code. Once the basic structure of a function is finalized, you can execute it by calling it from another function or directly from the Python prompt.

**2.7**​  **PYTHON USING OOP’s CONCEPTS:**

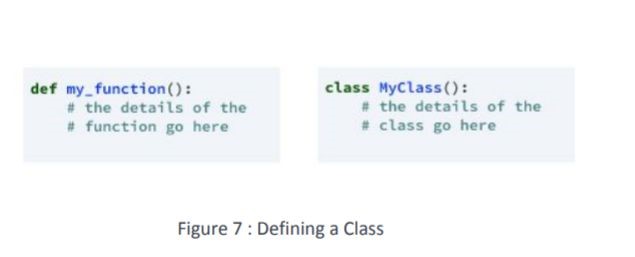
**2.7.1 Class:**

* Class: A user-defined prototype for an object that defines a set of attributes that characterize any object of the class. The attributes are data members (class variables and instance variables) and methods, accessed via dot notation.

* Class variable: A variable that is shared by all instances of a class. Class variables are defined within a class but outside any of the class's methods. Class variables are not used as frequently as instance variables are.
* Data member: A class variable or instance variable that holds data associated with a class and its objects.
* Instance variable: A variable that is defined inside a method and belongs only to the current instance of a class.

## ● Defining a Class:​

* We define a class in a very similar way how we define a function.
* Just like a function ,we use parentheses and a colon after the class name(i.e. ():) when we define a class. Similarly, the body of our class is 14 indented like a functionns body is.



**2.7.2 \_\_init\_\_ method in Class:**

* + The init method — also called a constructor — is a special method that runs when an instance is created so we can perform any tasks to set up the instance.
  + The init method has a special name that starts and ends with two underscores:\_\_init\_\_().

**CHAPTER 3**

# CASE STUDY

**3.1 PROBLEM STATEMENT:**

RECOGNITION OF GENDER BY THEIR VOICE

To predict the accuracy of the activity recognized by different algorithms and choose the best one.

* 1. **DATA SET:**

DESCRIBING THE COLUMNS:

* **meanfreq :** mean frequency (in kHz)
* **sd :** standard deviation of frequency
* **median :** median frequency (in kHz)
* **Q25 :** first quantile (in kHz)
* **Q75 :** third quantile (in kHz)
* **IQR :** interquantile range (in kHz)
* **skew :** skewness (see note in specprop description)
* **kurt :** kurtosis (see note in specprop description)
* **sp.ent :** spectral entropy
* **sfm :** spectral flatness
* **mode :** mode frequency
* **centroid :** frequency centroid (see specprop)
* **peakf :** peak frequency (frequency with highest energy)
* **meanfun :** average of fundamental frequency measured across acoustic signal
* **minfun :** minimum fundamental frequency measured across acoustic signal
* **maxfun :** maximum fundamental frequency measured across acoustic signal
* **meandom :** average of dominant frequency measured across acoustic signal
* **mindom :** minimum of dominant frequency measured across acoustic signal
* **maxdom :** maximum of dominant frequency measured across acoustic signal
* **dfrange :** range of dominant frequency measured across acoustic signal
* **modindx :** modulation index. Calculated as the accumulated absolute difference between adjacent measurements of fundamental frequencies divided by the frequency range
* **label :** male or female

**How it works?**

IMPORTING LIBRARIES

IMPORT

LOADING AND CLEANING DATA

ENCODING THE DATA

VIZUALIZATION

ANALYSIS

BUILDING THE MACHINE LEARNING MODEL

RA

**3.3 OBJECTIVE OF THE CASE STUDY:**

The objective is to predict​ the gender based on their voice.

Checking the accuracy by using random forest Should get high accuracy

# CHAPTER 4​

## MODEL BUILDING

**4.1**​  **PREPROCESSING OF THE DATA:**

Preprocessing of the data actually involves the following steps:

**4.1.1**​  **GETTING THE DATASET:**

​We can get the data set from the database or we can get the data from client.

**Dataset:**

The data set is taken from the given drive link.

The obtained dataset has been randomly partitioned into two sets

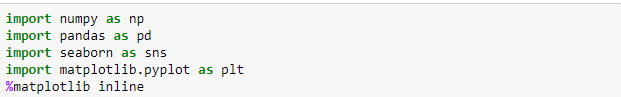
* Training data
* Testing data

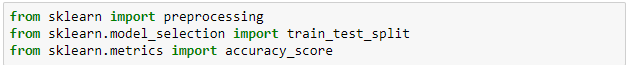
**4.1.2 IMPORTING THE LIBRARIES:**

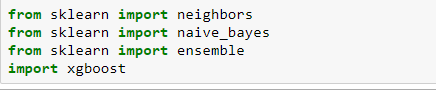
​We have to import the libraries as per the requirement of the algorithm.

​ ​**Importing the necessary packages and modules**

* **numpy** package can be used to perform mathematical operations like 'mean'.​
* **pandas** package can be used to process dataframes.​
* **seaborn** package can be used to visualise data in the form of various effective​ graphs and plots.
* **sklearn** is the main package which is used for machine learning.​
* Accuracy\_score: the set of labels predicted for a sample must exactly match the corresponding set of labels
* **Matplotlib:** Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.
* **LabelEncoder** is used to encode the non-numeric data into numericals so that​ machine learning model can be built.
* **train\_test\_split:**  module is used to split the data into training and testing sets.​
* KNN ALGORITHM: KNN can be used for both classification and regression predictive problems.
* ENSEMBLE: **ensemble** module includes two averaging algorithms based on randomized decision trees
* RANDOM FOREST: It consists of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated **forest** of trees whose prediction by committee is more accurate than that of any individual tree.
* XGBOOST: **XGBoost** is an optimized distributed gradient boosting library designed to be highly **efficient**, **flexible** and **portable**. It implements machine learning algorithms under the GRADIENT BOOSTING framework.



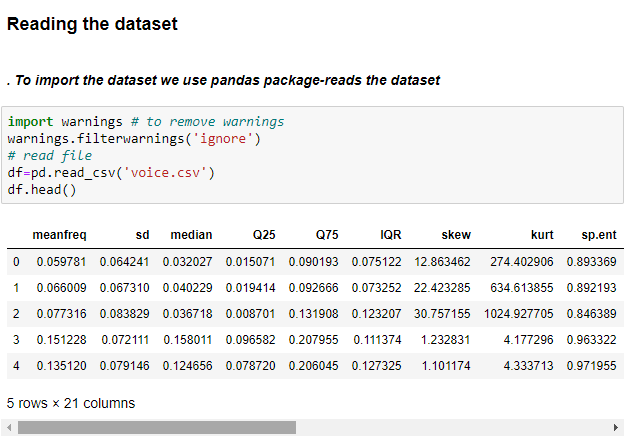


****

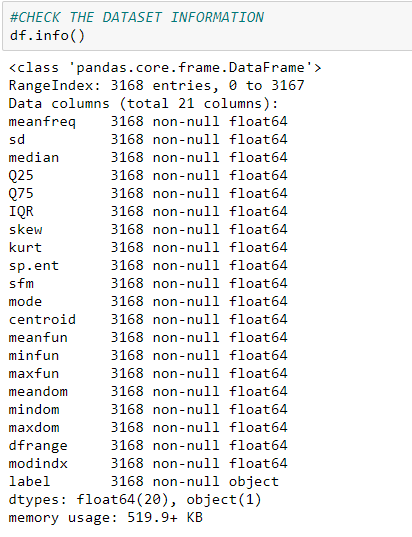
**4.1.3 IMPORTING THE DATA-SET:**

Pandas in python provide an interesting method read\_csv(). The read\_csv function reads the entire dataset from a comma separated values file and we can assign it to a DataFrame to which all the operations can be performed. It helps us to access each and every row as well as columns and each and every value can be access using the dataframe. Any missing value or NaN value have to be cleaned.

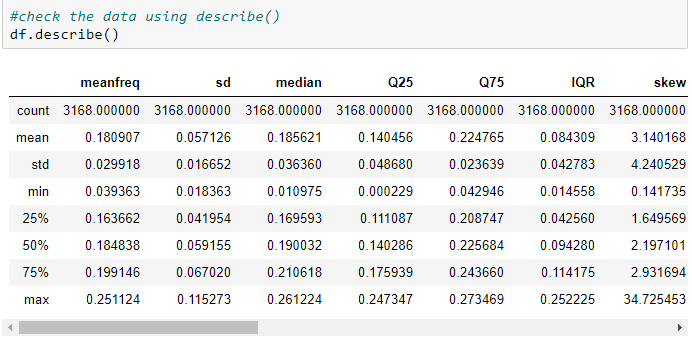
Reading data using pandas read.csv() and displaying the first 5 rows using head()

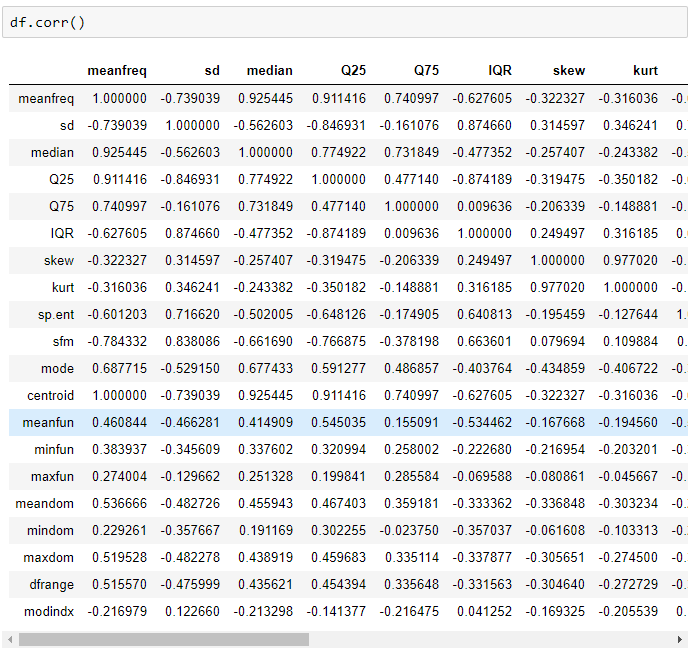


Reading data using pandas read.csv() and displaying the first 5 rows using head()



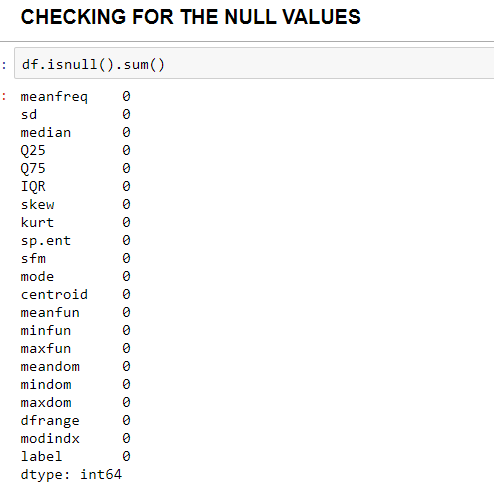
Describing the dataset:

****

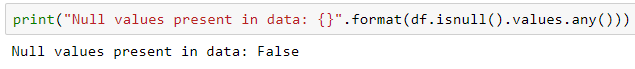
****

**4.1.4 HANDLING MISSING VALUES:**

Missing values can be handled in many ways using some inbuilt methods

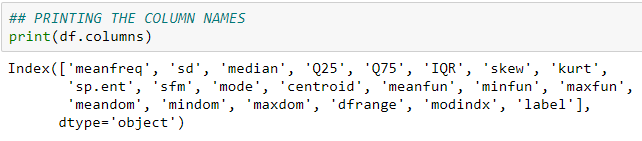


There are no null values in the dataset

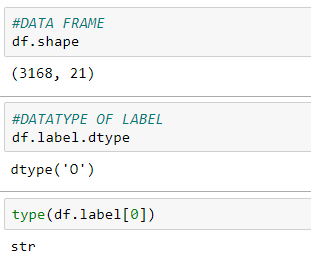


So,when we tried to print null values it says

“Null values present in the training data: False” , that there are no null values.

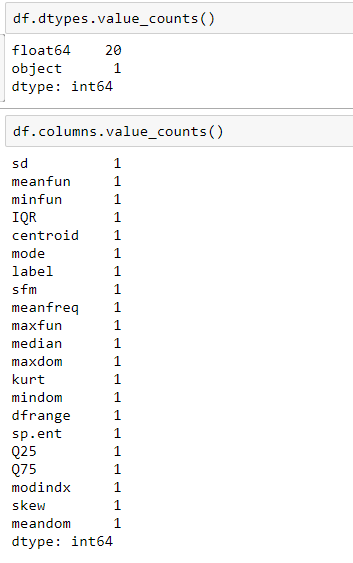


Checking the datatype :

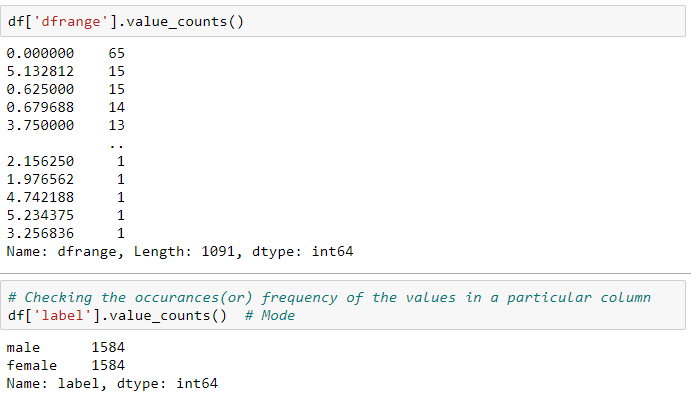


### 4.1.5 Processing the dataset

After the data has been imported, we have to pre processing the data



This image displays the data type of values present in dataset



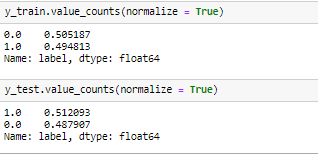
4.2: TRAINING THE DATA:

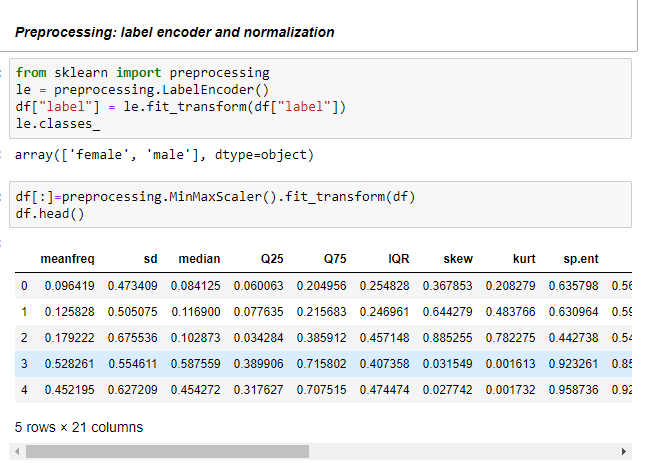
4.2.1: PREPROCESSING THE DATASET USING LABEL ENCODER AND NORMALIZATION:

Using label encoder encoding and fitting the target variables range as i,i-1

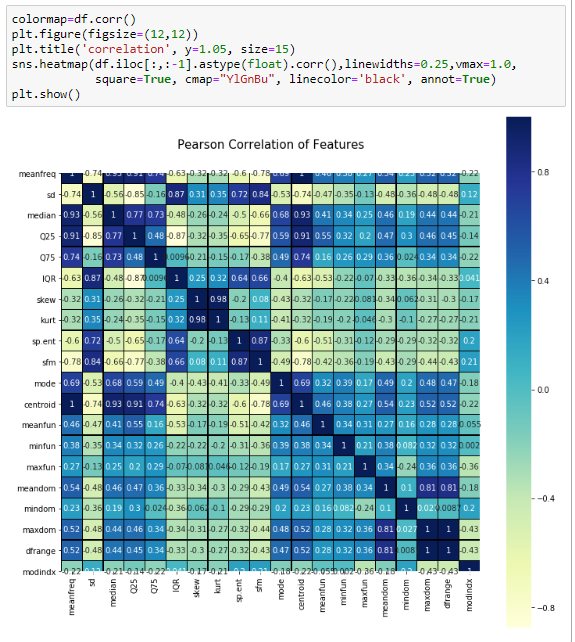
Min Max Scalar():Transform features by scaling each feature to a given range.

NORMALIZATION:



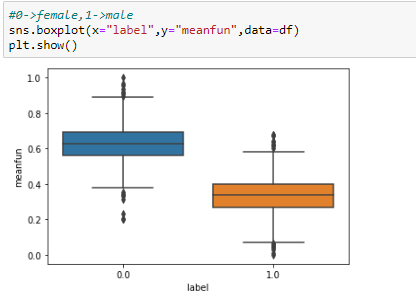


Vizualizing the data using correlation:



the correlation summarizes that association among all the columns.

Now by using barplot showing the mean function for both male and female category

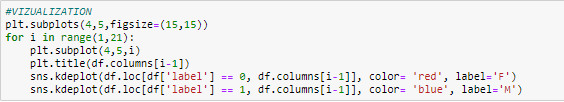


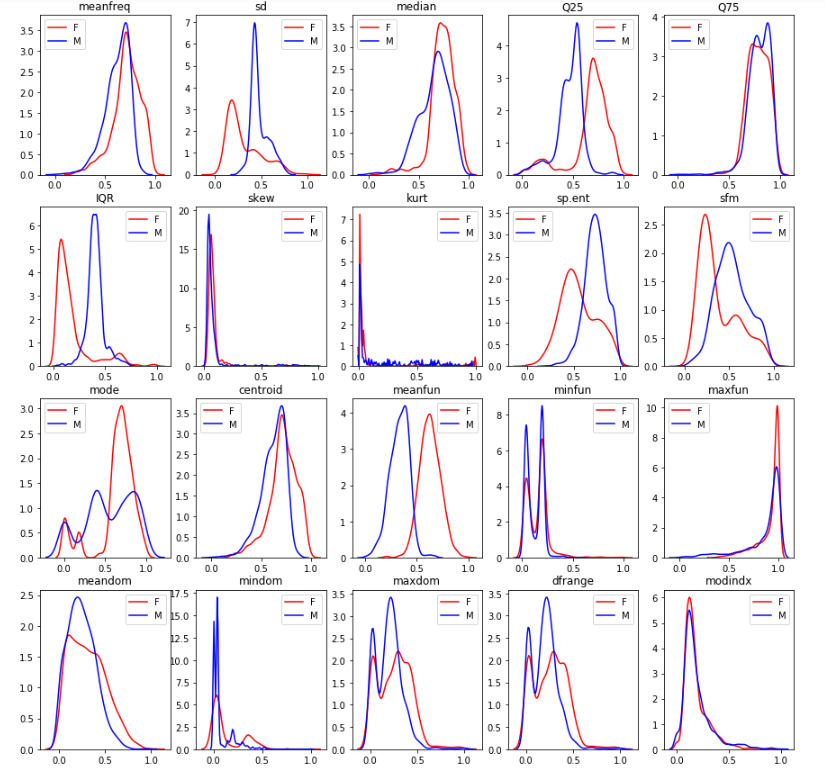
Female are more compared to male.

4.2.3: VIZUALIZATION:

Vizualizing the data by using kdeplot

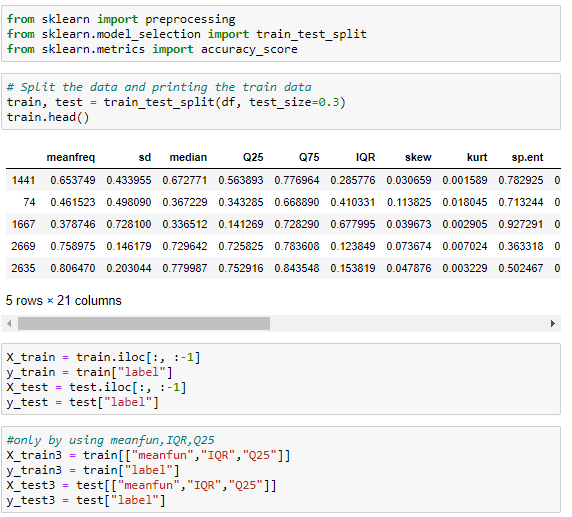
I will visualise the training data to get a better understanding of the available dataset.





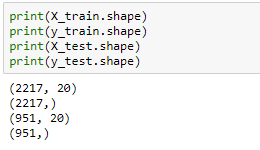
At first glance, most of the significant features are Q25, IQR and meanfun. We will build models by using the 20 features and the 3 distinct features.

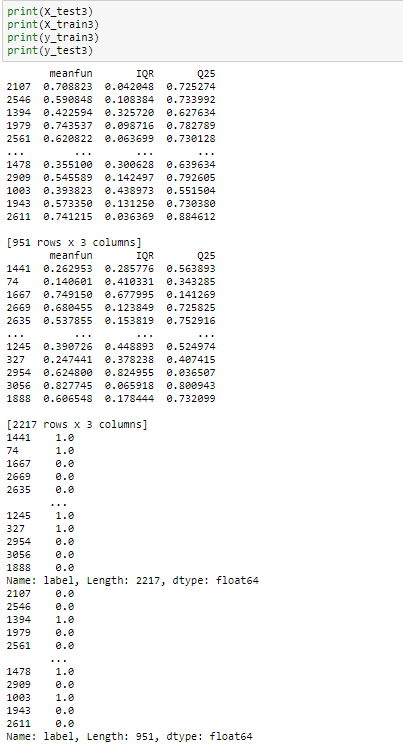
**4.2.4: Splitting the data into train and validation**

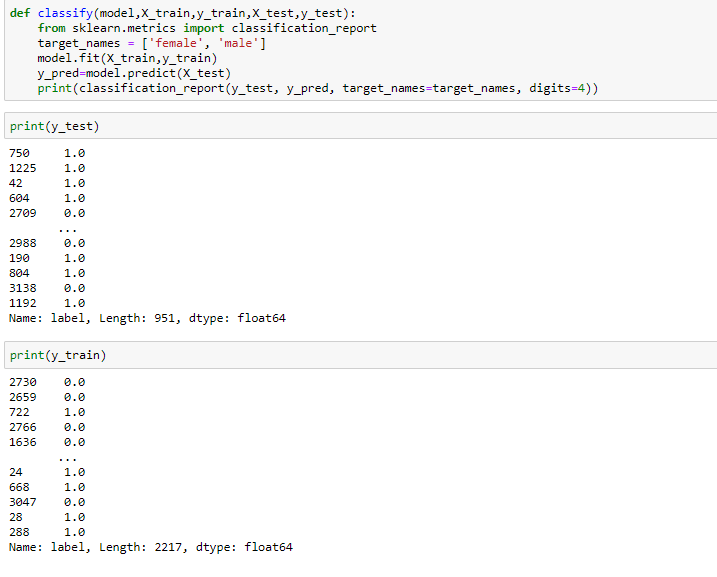


Here splitting of training and testing data is done.

Now defining the shape of the train and test data





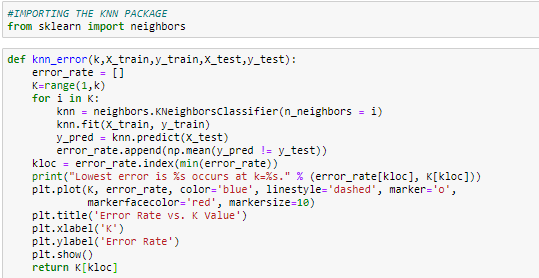
****

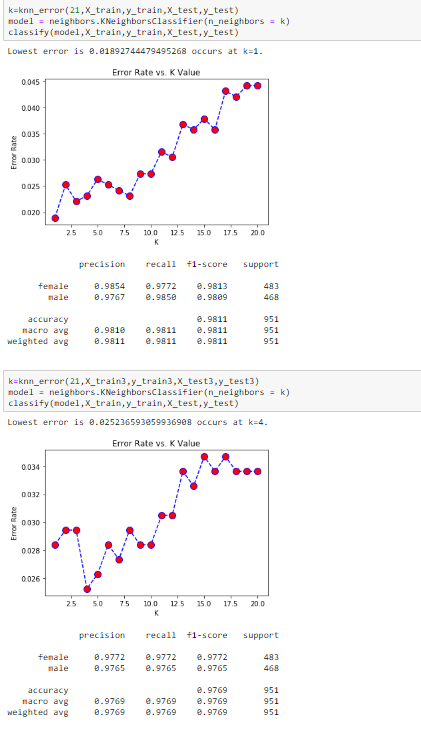
applying the classification report for the training and testing data.

4.3APPLYING THE ALGORITHMS:

4.3.1: K-NEAREST-NEIGHBORS(KNN):

Using neighbor. K Neighbors Classifier() to build the model

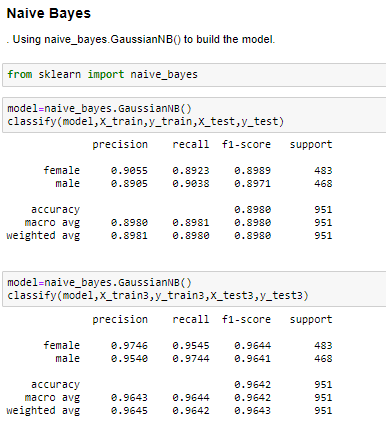




By using the KNN im getting 97.6% of accuracy

4.3.2: Naïve Bayes:

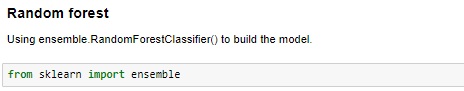
Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

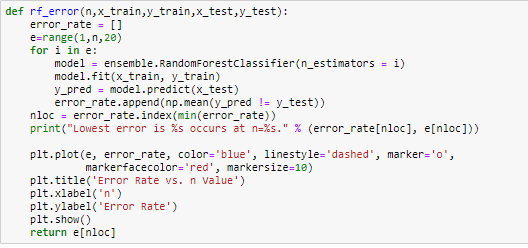


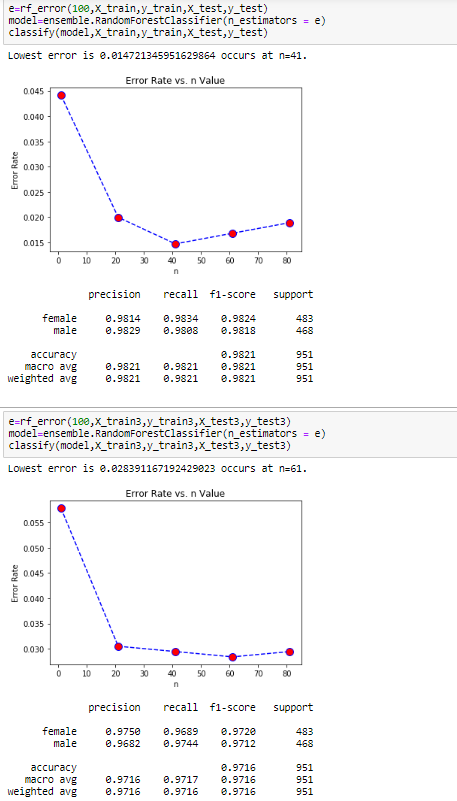
By using the naïve bayes getting the accuracy as 96%

4.3.3: RANDOM FOREST:

Random forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees.



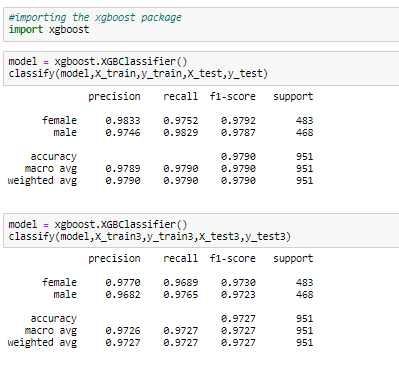




By using the random forest im getting the accuracy as 98%

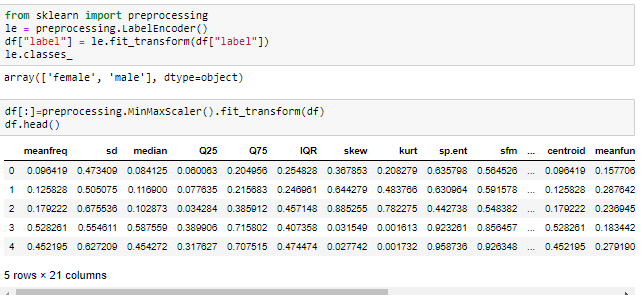
4.3.4: XGBOOST:

Using xgboost.XGBClassifier() to build the model.

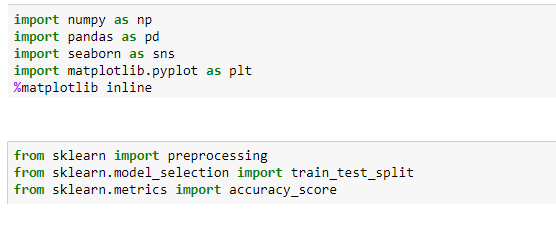


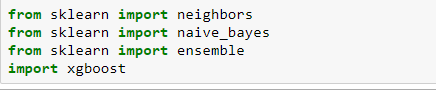
By using xgboost im getting the accuracy as 97%

* 1. **EVALUATING THE CASE STUDY:** 
     1. **LABEL ENCODER:**

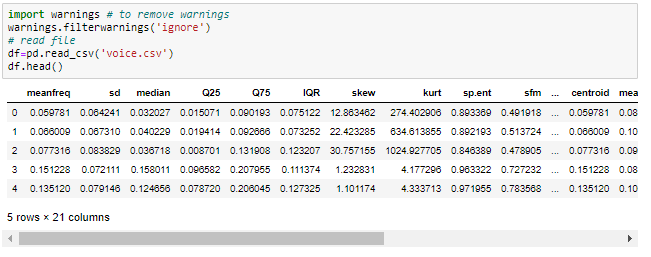


4.4.2 Importing the required libraries



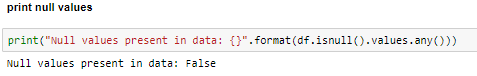
****

* + 1. Reading the Data-Set:

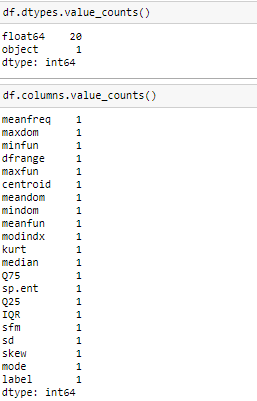


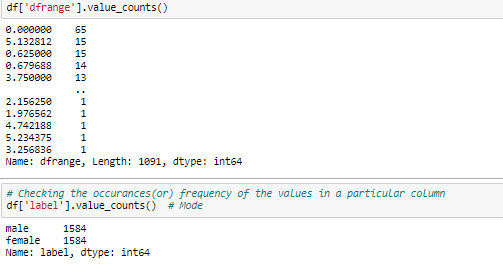
4.4.4 Handling the missing values :

o There is a method called isnull() which gives the number of missing values in each and every column. o Using fillna() method each and every missing value is replaced by 0.

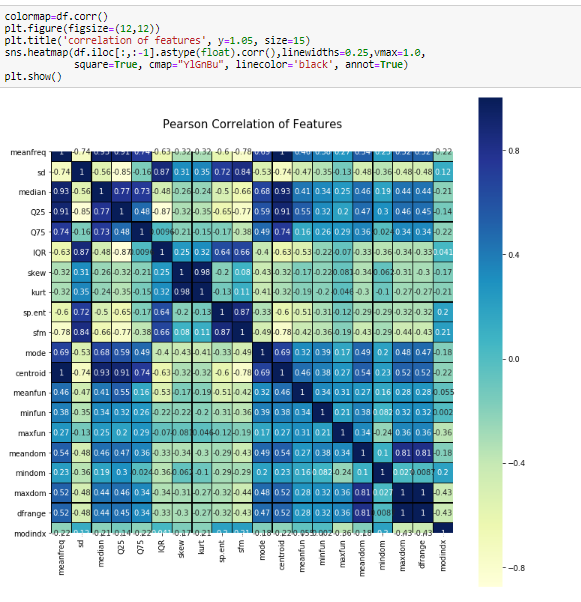


Data types :

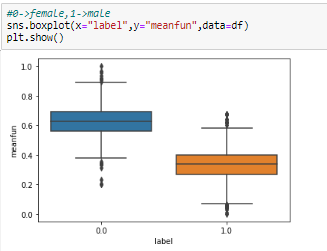




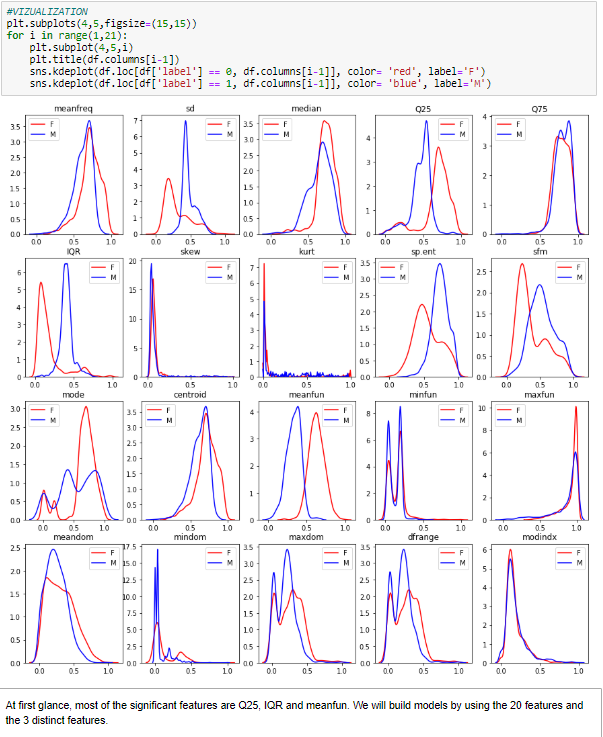
4.4.5 Visualize the correlation :



Now plotting the boxplot for the label

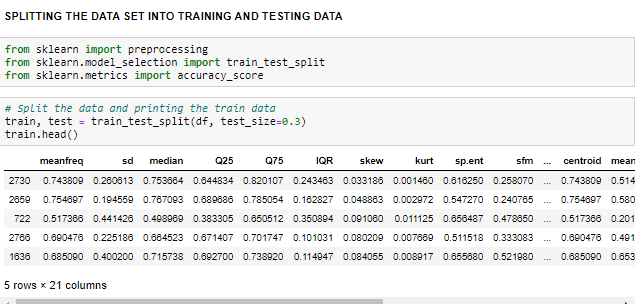


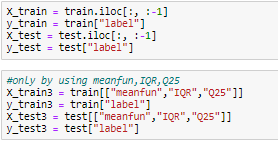
4.4.6:VIZUALIZATION:

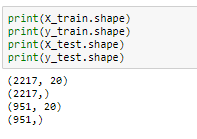


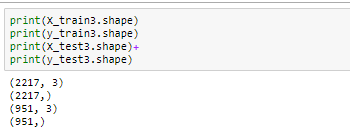
4.4.7 : Splitting data :

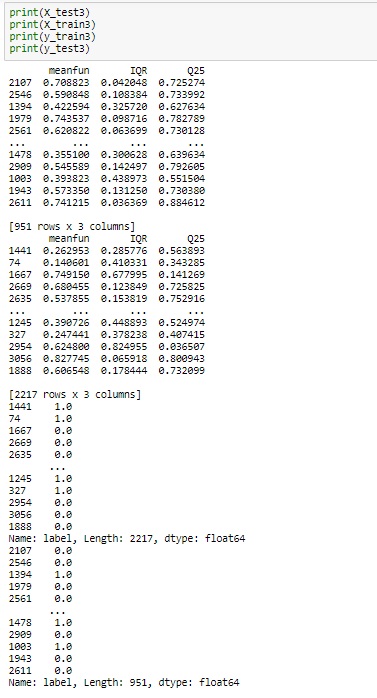
In the above splitting we drop subject column,this is because the values of the subject are all 1 in the dataset. Hence even if we drop it there would be no change to the dataset We find the no of rows and columns of the split data using shape



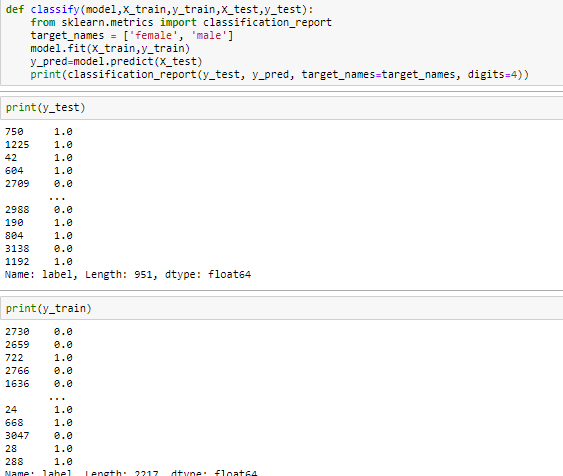




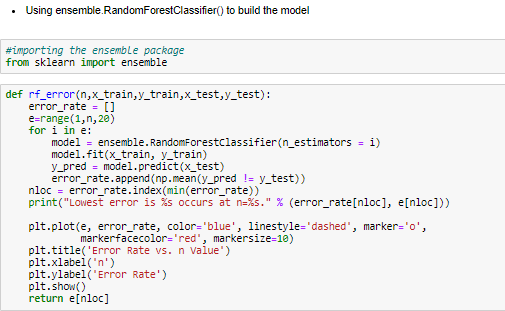


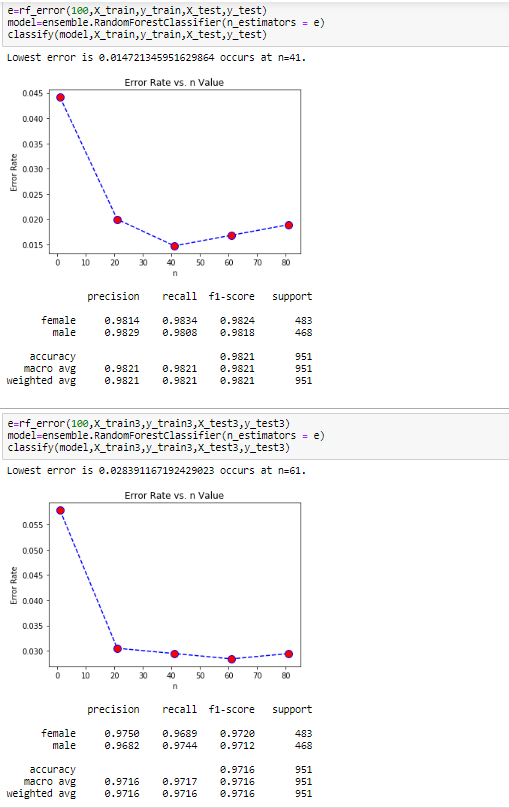


4.4.8 Classification report:



4.4.9: Random forest:

****



After evaluating this model it is giving 98 of accuracy

**Conclusion:**

In this particular project, I explored the activity recognition dataset. I visualized that by using different algorithms .Finally I got the highest accuracy by using RANDOM FOREST.

So I conclude that RANDOM FOREST is the best algorithm for solving this problem.