**MACHINE LEARNING**

(Predicting Amount of Accuracy of Human Activity Recognition)

Summer Internship Report Submitted in partial fulfillment

of the requirement for undergraduate degree of

**Bachelor of Technology**

In

**Computer Science Engineering**

By

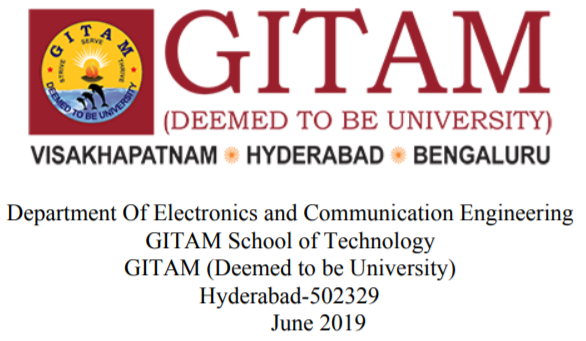
**Yashwini Sai Kodavati**

**221710315063**

Under the Guidance of

**Mr…………………..**

Assistant Professor

****

i

**DECLARATION**

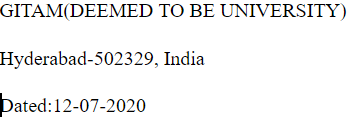
I submit this industrial training work entitled **“ HUMAN ACTIVITY RECOGNITION** **”** to GITAM (Deemed To Be University) , Hyderabad in partial fulfillment of the requirements for the award of the degree of **“ Bachelor of Technology ”** in **“ Computer Science Engineering ”**. I declare that it was 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 Yashwini Sai Kodavati

Date:12-07-2020 221710315063

ii

**CERTIFICATE**

This is to certify that the Industrial Training Report entitled **“ HUMAN ACTIVITY RECOGNITION ”** is being submitted by Yashwini Sai Kodavati (221710315063) in partial fulfillment 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.

Yashwini Sai Kodavati

221710315063

**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 LEARNING1**

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

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

1.3 USES OF MACHINE LEARNING……………………………………….13

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………………………………………………...18

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

2.4.1 Installation(using python IDLE)………………………………...19

2.4.2 Installation(using Anaconda)……………………………………20

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

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

2.5.2 Python Strings………………………………………………….23

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

2.5.4 Python Tuples…………………………………………………..24

2.5.5 Python Dictionary……………………………………………...25

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…………………………………………………………….27

2.7.2 \_\_init\_\_method in class………………………………………...28

**CHAPTER 3:CASE STUDY..………..…………………………………………………...29**

3.1PROBLEM STATEMENT………………………………………………….29

3.2 DATA SET………………………………………………………..………...29

3.3 OBJECTIVE OF THE CASE STUDY……..……………………………….31

**CHAPTER 4:MODEL BUILDING…..…………………………………………….…..32**

4.1 PREPROCESSING OF THE DATA…………………….………………….32

4.1.1 Getting the Data Set……………………………………………..32

4.1.2 Importing the Libraries………………………………………….32

4.1.3 Importing the Data-Set…………………………………………..33

4.1.4 Handling the Missing values…………………………………….35

4.1.5 Correlation .……………………………………………………...43

4.2 TRAINING THE MODEL…………………....…………………………….45

4.2.1 Visualize the data………………………………………………..46

4.2.2 Classify activities……………………………………….………..55

4.3 EVALUATING THE CASE STUDY……………………………………….59

4.3.1 Building the model(using logistic regression)...…….…………….60

4.3.2 Building the model(using splitting)………………………...…….61

CONCLUSION………………………………………………………………......65

**LIST OF FIGURES:**

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

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

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

Figure 4 : Python download…………………….…....…………………………………...17

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

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

Figure 7 : Defining a Class………………………..……………………...……………....28

Figure 8 : Importing Libraries……………………..………………………....…………..33

Figure 9 : Reading the Dataset…………...…………..…………………………………..34

Figure 10 : data using dropna()...........………………...……………………..…………..37

Figure 11 : data using isnull()………...………...………...…………………………….. 37

Figure 12 : datatypes in dataset…..…………....………………………………………...38

Figure 13 : describing the dataset……….……………………..…..…..………………...39

Figure 14 : heatplot……...…....…………...……………………………………………..40

Figure 15 : object feature….....……………..…………....………………………………41

Figure 16 : activities performed……………………………………………………...….42

Figure 17 : calculating correlation rows..…….……………………..………………….44

Figure 18 : calculating correlation columns…...…………....………………………….44

Figure 19 : importing the method……………………………..……………………….45

Figure 20 : importing train\_test\_split…………………...…………………………….45

Figure 21 : visualize data………...……………………………………………………46

Figure 22 : bar plot……………………………………………...…………...………..47

Figure 23 : pie plot…………………….……………………………………..……….48

Figure 24 : plotting different subjects…………………..…………..………………...53

Figure 25 : barplot(accuracy of test data)…………………………………………….57

Figure 26 : barplot(accuracy of train data)…….…………………………………......58

Figure 27 : importing the libraries………….………………………………….……..59

Figure 28 : reading the dataset………………………………….…………….………60

Figure 29 : handling the missing values…………...………….…………….………..60

Figure 30 : data types……………………....……………………….…….………….60

Figure 31 : activity column…………………...……………………….….………….61

Figure 32 : correlation……………………..….………………….…….....………….61

Figure 33 : splitting data……………………………..…....………….…..………….61

Figure 34 : shape of splitted data……………………….…..…………….………….62

Figure 35 : visualize the data…………….….…………………………………….…62

Figure 36 : plotting activities………………...…………………..…………………..62

Figure 37 : calculating standing activity………….…….……………………………63

Figure 38 : classify activities…………….……………..…………………………….64

Figure 39 : logistic regression (test data)……….…………………...………………..64

Figure 40 : logistic regression (train data)……...…………………...…...…………...64

Figure 41 : Barplot for accuracy…………….…………………………..……………65

**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:**

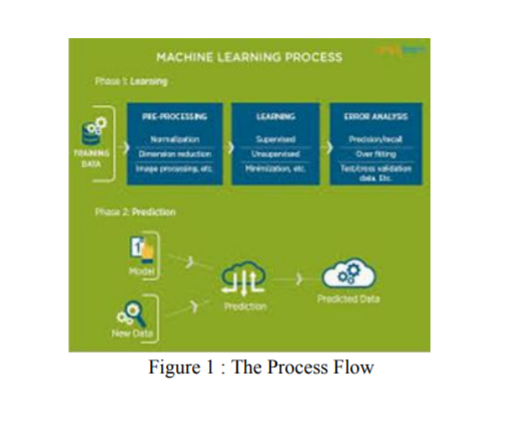
Consider some of the instances where machine learning is applied: the self-driving Google car, cyber fraud detection, online recommendation engines—like friend suggestions on Facebook, Netflix showcasing the movies and shows you might like, and “more items to consider” and “get yourself a little something” 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 learng 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.

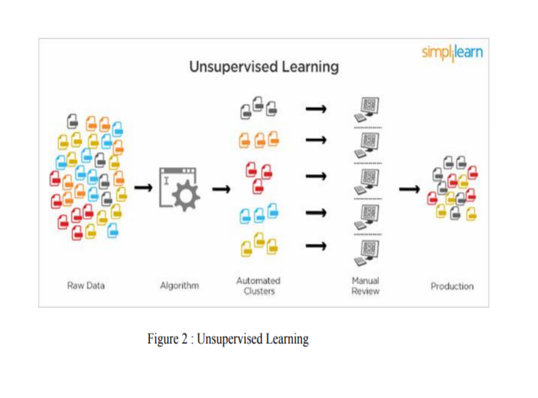
Examples of Supervised Machine Learning Techniques are Regression, in which the algorithm returns a numerical target for each example, such as how much revenue will be generated from a new marketing campaign.

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.

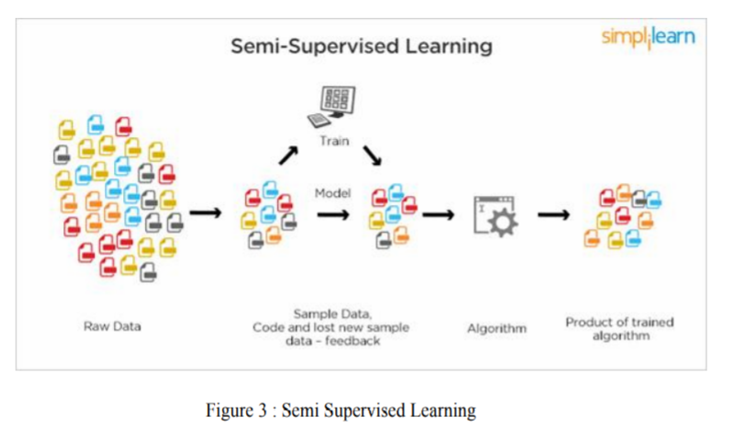
This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of uncorrelated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms



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 labeled and unlabeled data for training. In a typical scenario, the algorithm would use a small amount of labeled data with a large amount of unlabeled 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.

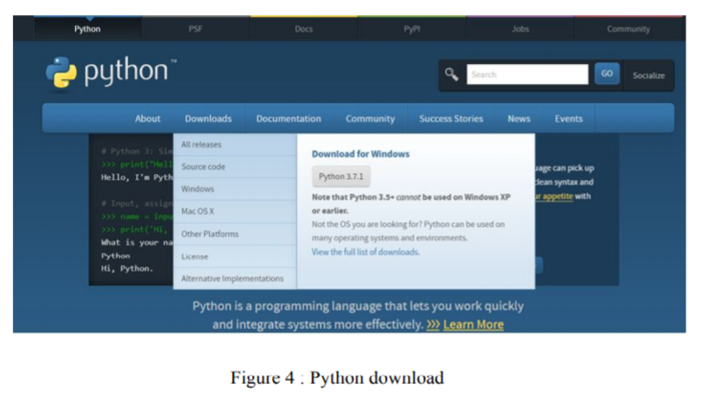
**2.4.1 Installation(using python IDLE):**

● 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

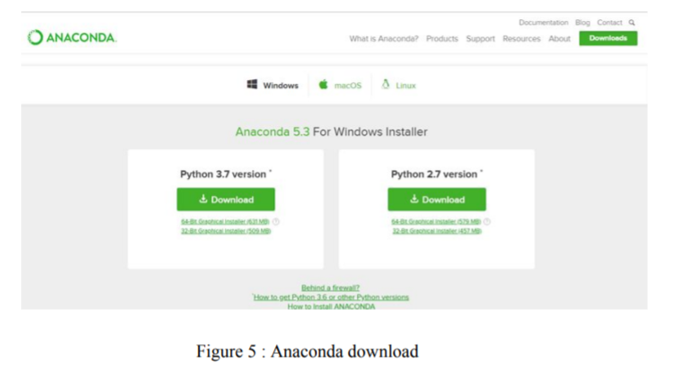
graphic installer)

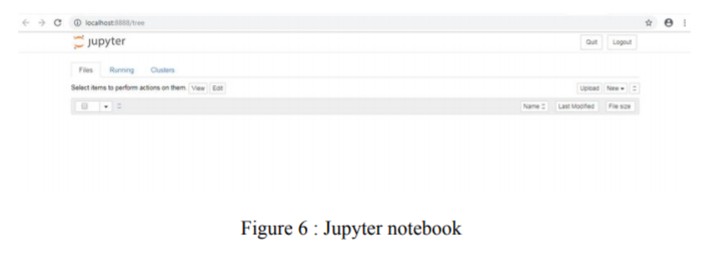
● Step 3: select installation type( all users)

● 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 –

o Number.

o Strings

o Lists

o Tuples

o 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.

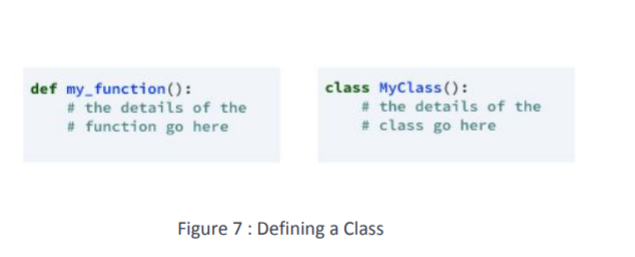
● Data member: A class variable or instance variable that holds data associated with a class and its objects.

● Instance variable: A variable that is defined inside a method and belongs only to the current instance of a class.

● **Defining a Class:**

o We define a class in a very similar way how we define a function.

o 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:**

**Human Activity Recognition or HAR for short, is the problem of predicting what a person is doing based on a trace of their movement.**

Movements are often normal indoor activities such as standing, sitting, jumping, and going up stairs. Sensors are often located on the subject such as a smartphone or vest and often record accelerometer data in three dimensions (x, y, z).

The idea is that once the subject’s activity is recognized and known, an intelligent -computer system can then offer assistance.

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

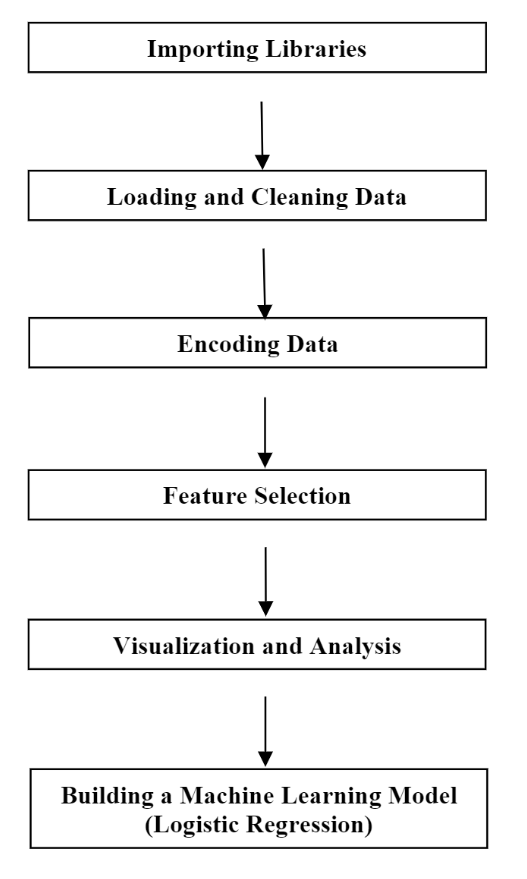
**3.2 DATA SET:**

## Attribute information

For each record in the dataset the following is provided:

* Triaxial acceleration from the accelerometer (total acceleration) and the estimated body acceleration.
* Triaxial Angular velocity from the gyroscope.
* A 561-feature vector with time and frequency domain variables.
* Its activity label.
* An identifier of the subject who carried out the experiment.

**How it works?**



**3.3 OBJECTIVE OF THE CASE STUDY:**

The objective is to classify activities into one of the six activities performed and predict the highest accuracy among few different algorithms and take the best one..Each person performed six activities

1.WALKING

2.WALKING\_UPSTAIRS

3.WALKING\_DOWNSTAIRS

4.SITTING

5.STANDING

6.LAYING

wearing a smartphone on the waist.

The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain.

**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 Train and Test datasets which are considered in this notebook file are

taken from kaggle

The obtained dataset has been randomly partitioned into two sets, where 70% of

volunteers was selected for generating the training data and 30% the test data.

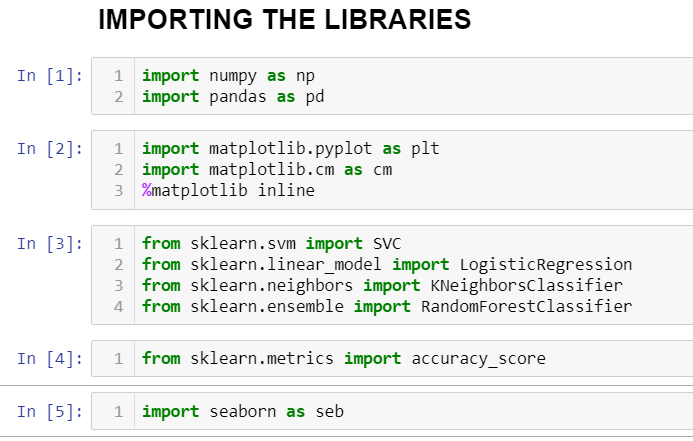
[**https://www.kaggle.com/uciml/human-activity-recognition-with-smartphon**](https://www.kaggle.com/uciml/human-activity-recognition-with-smartphones)

**4.1.2 IMPORTING THE LIBRARIES:**

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

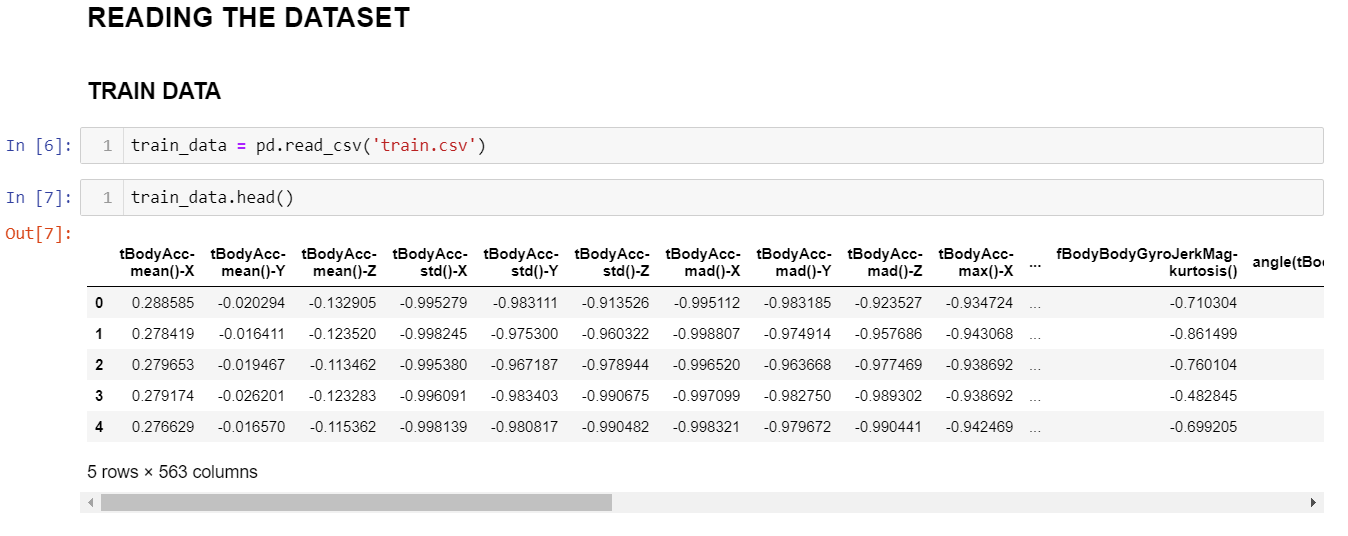
**Importing the necessary packages and modules**

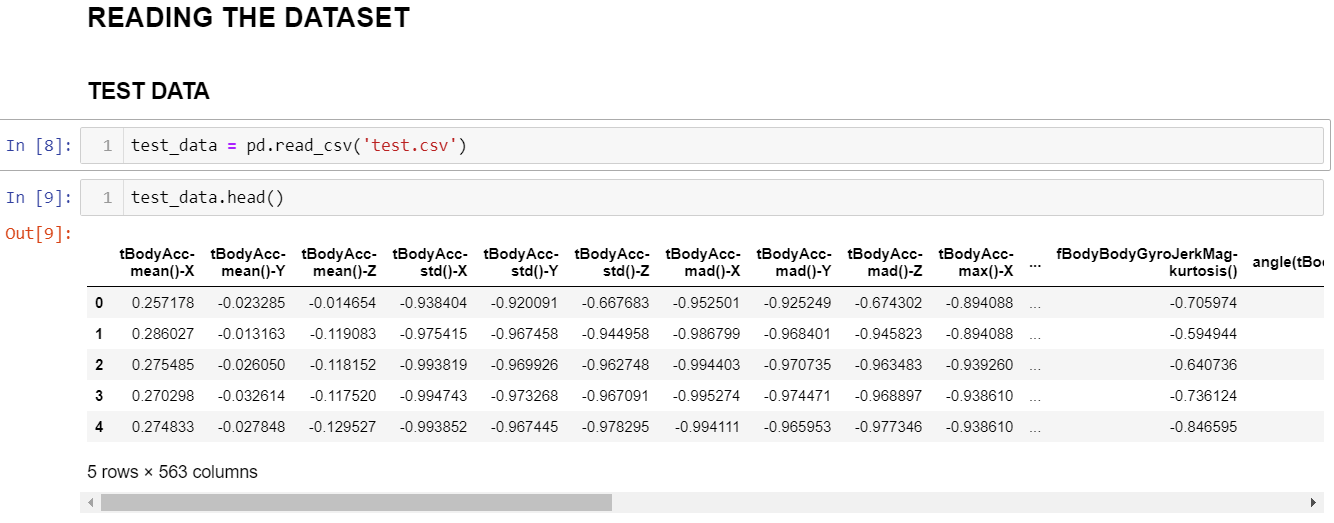
* **numpy** package can be used to perform mathematical operations like 'mean'.
* **pandas** package can be used to process dataframes.
* **seaborn** package can be used to visualise data in the form of various effective graphs and plots.
* **sklearn** is the main package which is used for machine learning.
* **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.
* **LinearRegression** module is used to fit a LinearRegression model.
* **sklearn.metrics** can be used to calculate statistical results like mean squared error, root mean squared error, etc.



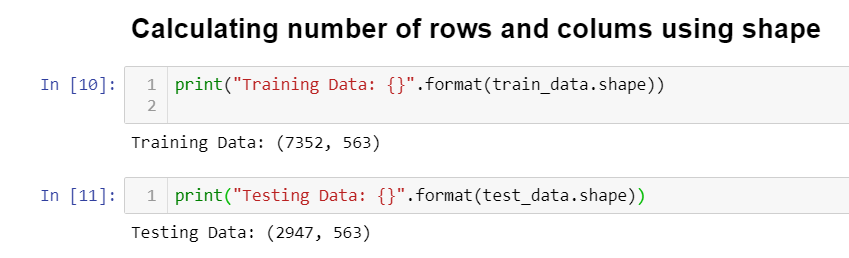
**4.1.3 IMPORTING THE DATA-SET:**

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()



Hence,We have 7352 rows and 563 columns in the train dataset

We have 2947 rows and 563 columns in the test dataset

**4.1.4 HANDLING MISSING VALUES:**

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

**(a)dropna():**

dropna() is a function which drops all the rows and columns which are having

the missing values(i.e. NaN)

● dropna() function has a parameter called how which works as follows

● if how = ’all’ is passed then it drops the rows where all the columns of the

particular row are missing

● if how = ’any’ is passed then it drops the rows where all the columns of the

particular row are missing

**(b)fillna():**

fillna() is a function which replaces all the missing values using different

ways.

● fillna() also have parameters called method and axis

● if we use method = ’ffill’ where ffill is a method called forward fill, which

carry forwards the previous row’s value

● if we use method = ‘bfill’ where bfill is a method called backward fill, which

carry backward the next row’s value

● if we use method = ‘ffill’ , axis = ‘columns’ then it carry forwards the

previous column’s value

● if we use method = ‘bfill’ , axis = ‘columns’ then it carry backward the next

column’s value

**(c)interpolate():**

● interpolate() is a function which comes up with a guess value based on the

other values in the dataset and fills those guess values in the place of missing

values

**(d)mean imputation and median imputation:**

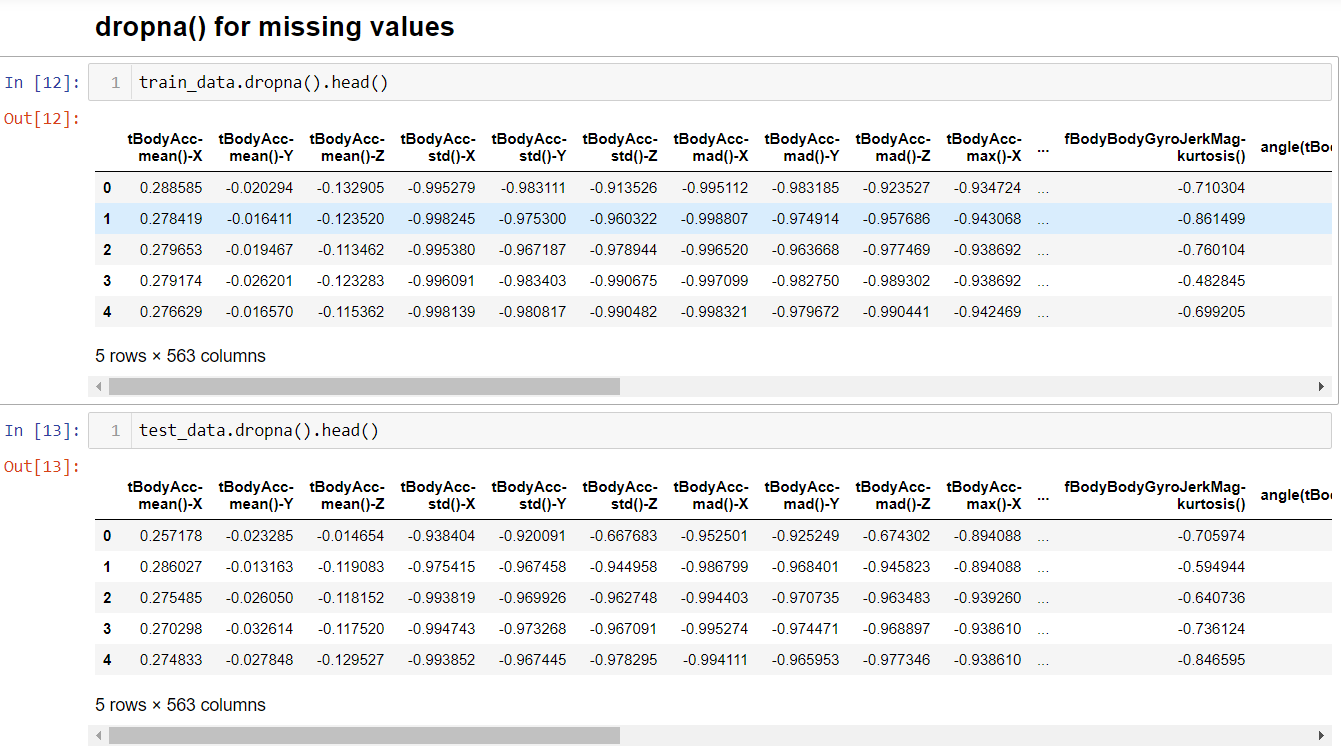
● mean and median imputation can be performed by using fillna().

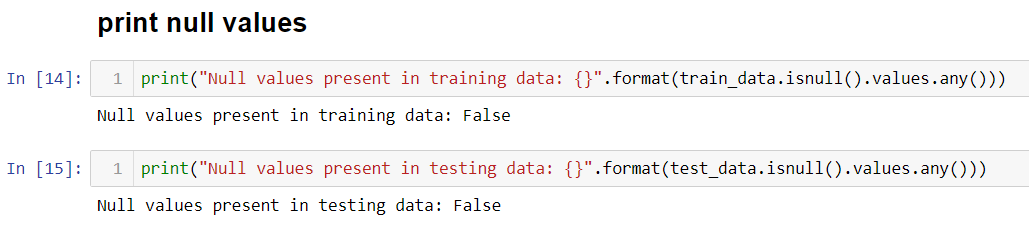
● mean imputation calculates the mean for the entire column and replaces

the missing values in that column with the calculated mean.

● median imputation calculates the median for the entire column and

replaces the missing values in that column with the calculated median.



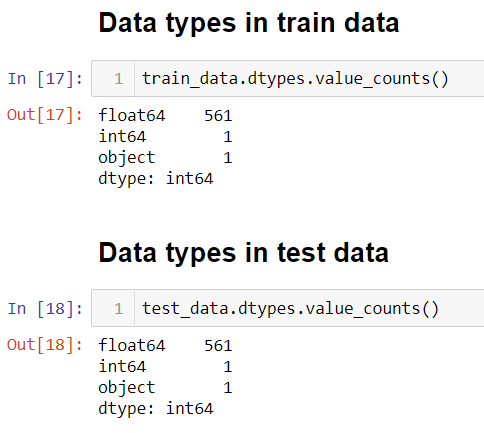


When we used dropna() the dataset didn’t change and displayed same number of columns,stating that there are no missing values in the data set.

So,when we tried to print null values it says

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



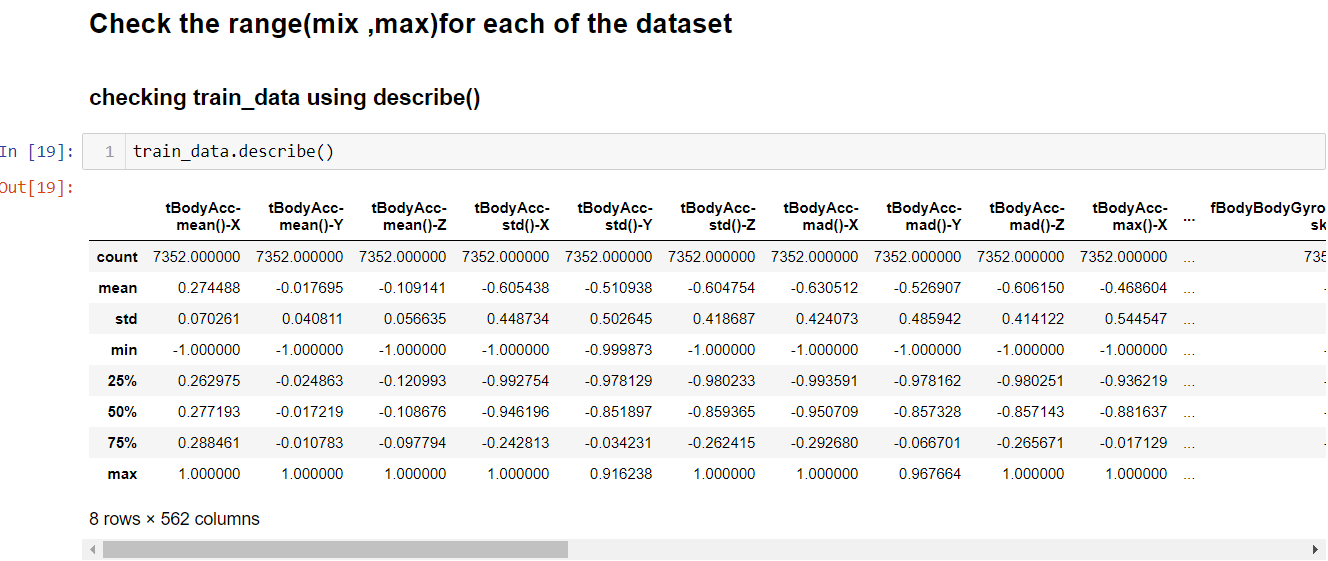


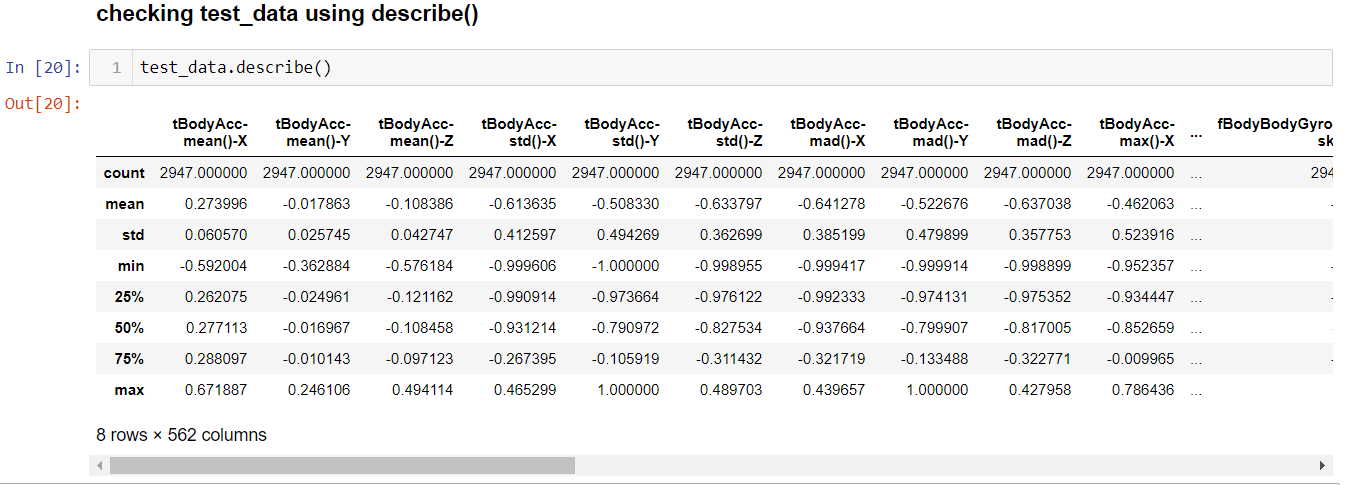
This image displays the data type of values present in both train and test dataset

**Scaling** a dataset usually produces better dataset and more accurate predictions.

First we check the range( the min and the max) for each of the datasets.

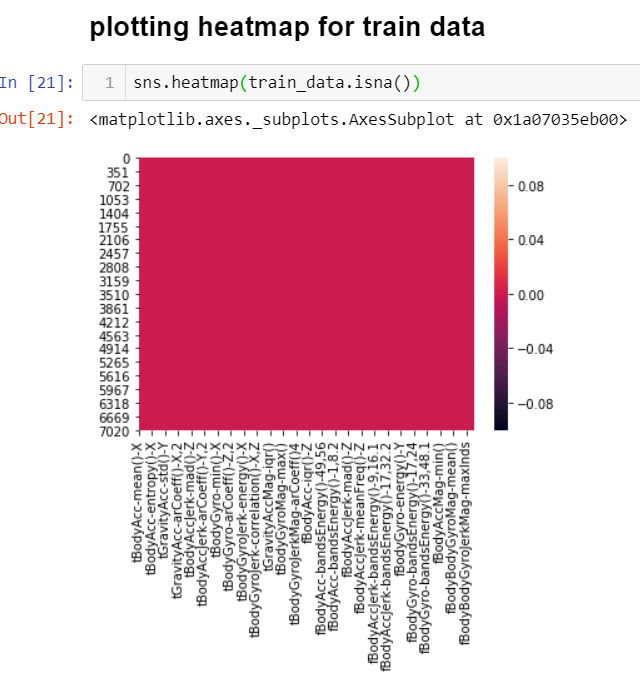
Lets try using the .describe() method and lets exclude the activity column which is the last column.

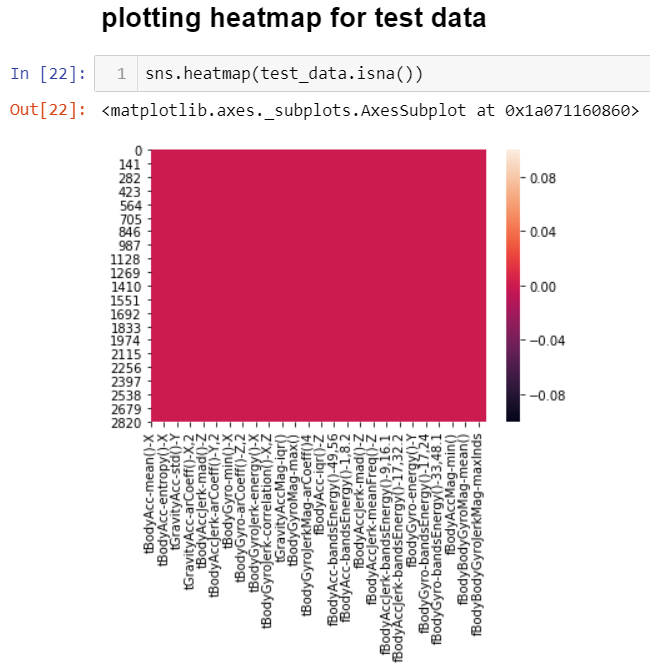


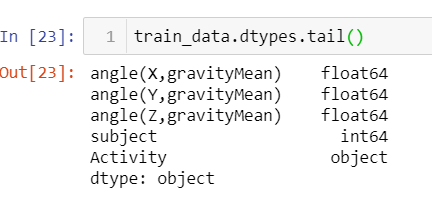


Plotting heat map for the dataset

* The heatmap does not have any empty cells because there are no null values in both the test and train data sets.

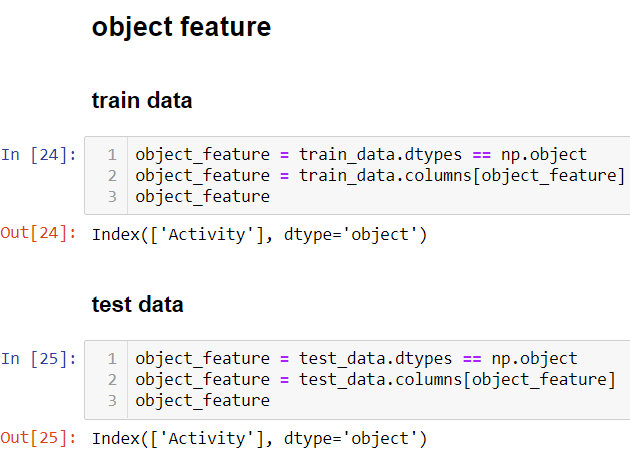




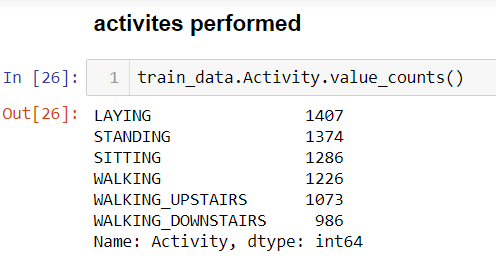


They have the same data types. That is, mostly float and one object feature.

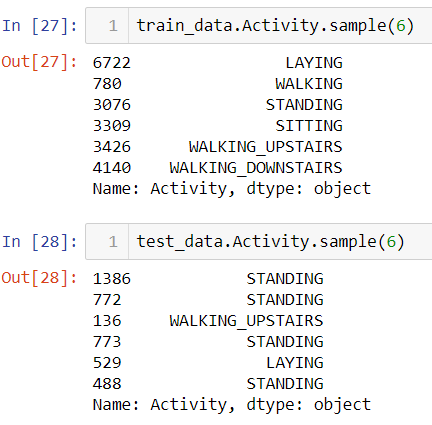
Lets extract the object feature from the rest and see what it is about.



As we can see, the only object data type in both train and the test dataset is the Activity feature. Let's look about the activities in detail…..



These 6 activities are performed with their count values.



## **Finding the Correlation/ Relationships between the features**

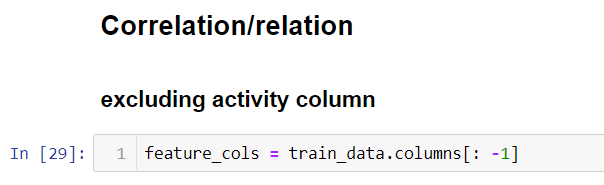
Correlation refers to the mutual relationship and association between quantities and it is generally used to express one quantity in terms of its relationship with other quantities. The can either be Positive(variables change in the same direction), negative(variables change in opposite direction or neutral(No correlation).

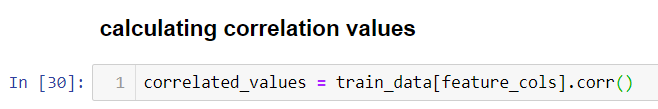
Variable within a dataset can be related in lots of ways and for lost of reasons:

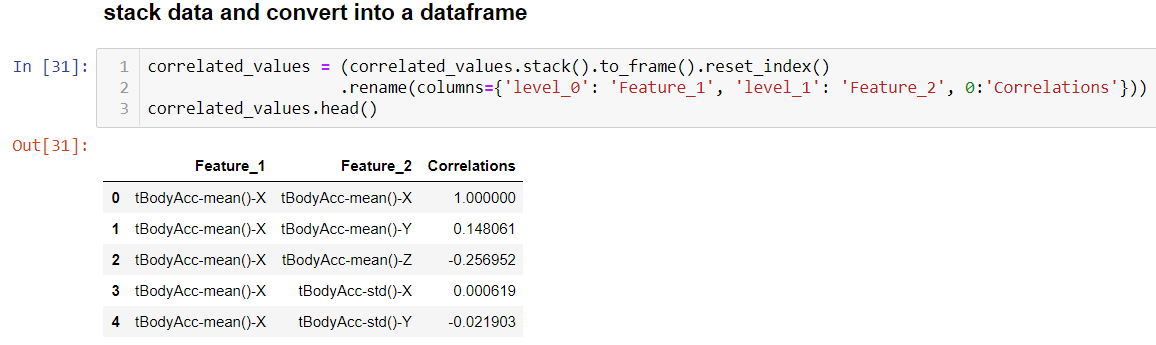
- They could depend on values of other variables.

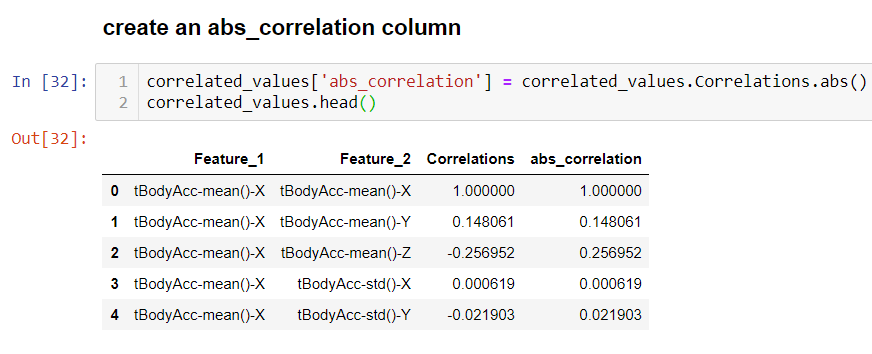
- They could be associated with each other.

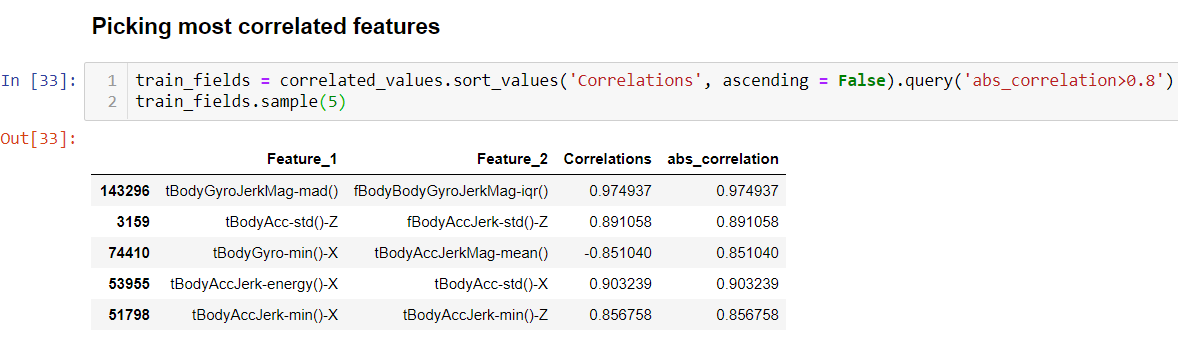
- They could both depend on a third variable.











## **Splitting the data into train and validation**

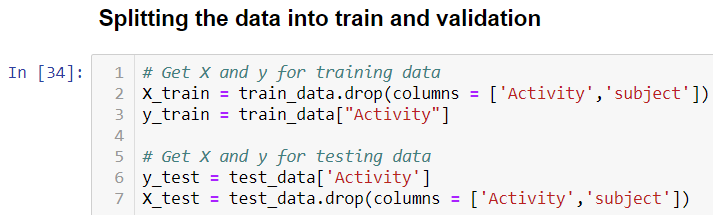
We can see that the dataset consists of accelerometer and gyroscope sensor values for each

record.

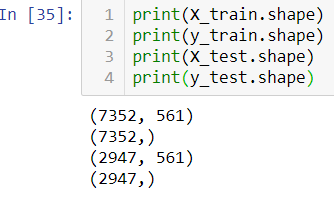
Further, the last two columns are subject which refers to subject number and Activity which

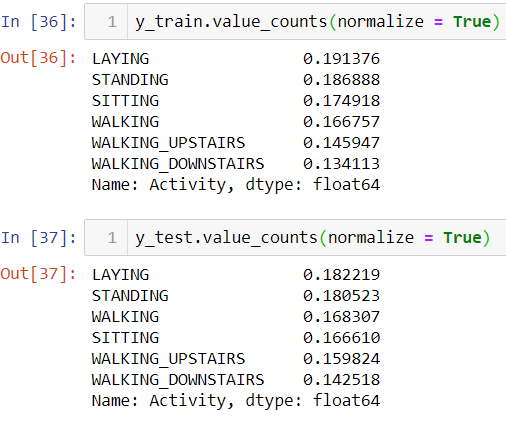
defines the type of activity.

The Activity column acts as the label y and all the rest columns are features X.



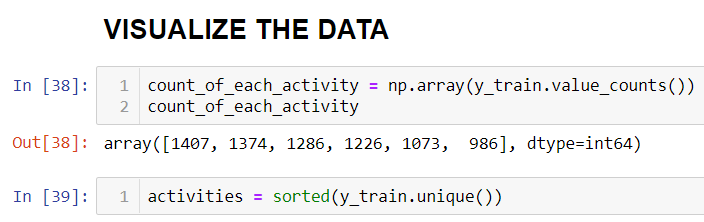
Here we drop the subject column as if we look at the data set,all the values in the subject column are 1. Hence even if we drop it,the dataset will not be affected.



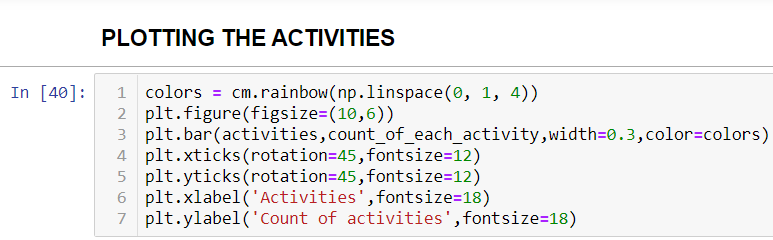


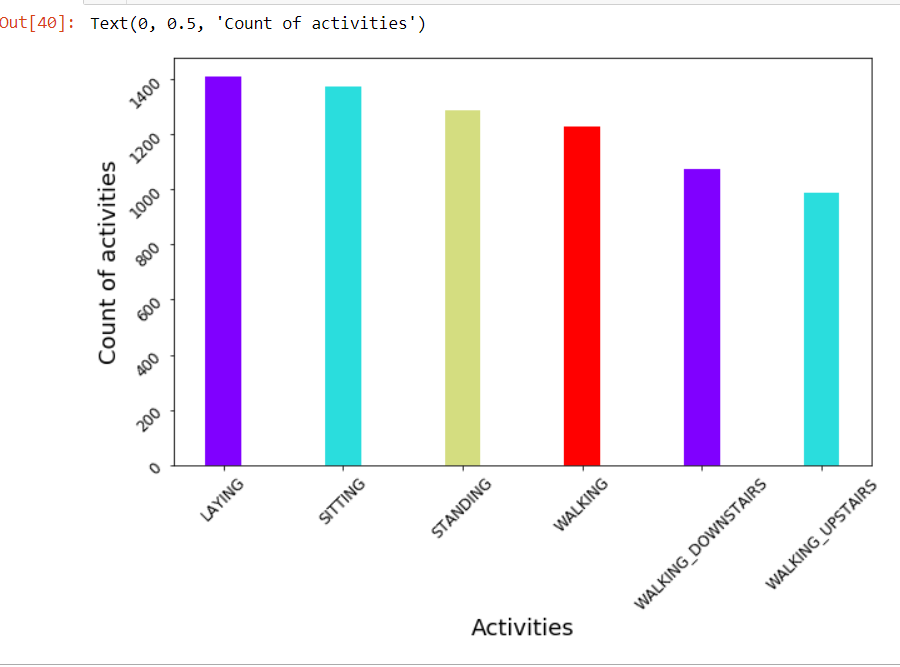
## **Visualize the dataset:**

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



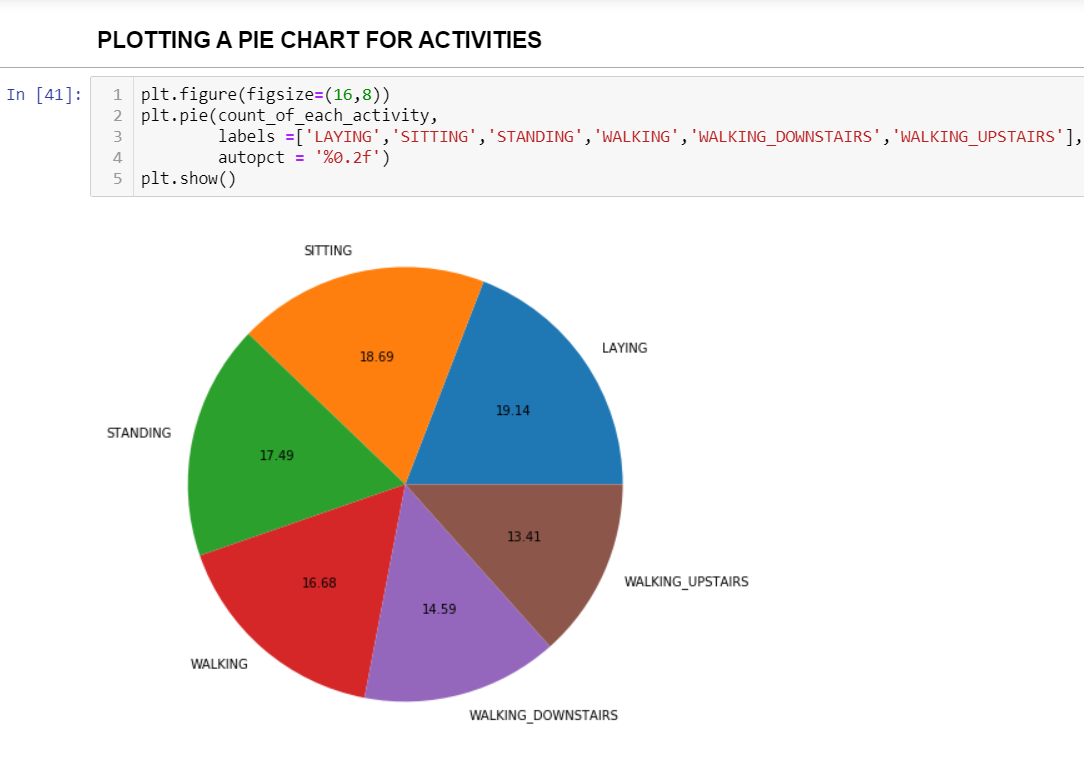
**BAR PLOT**





From the above Bar graph,we can clearly observe that it displays activities performed on the x-axis and count on the y-axis.The laying activity is carried out more,then comes sitting activity followed by standing,walking,walking\_downstairs,walking\_upstairs.

**PIE PLOT**



From the plot ,we observe that:

laying-19.14%

sitting-18.69%

standing-17.49%

walking-16.68%

walking\_downstairs-14.59%

walking\_upstairs-13.41%

The percentage values show that the data size for each activity is comparable. The dataset is equally distributed.

On inspecting the dataset, I can see that there are many features.

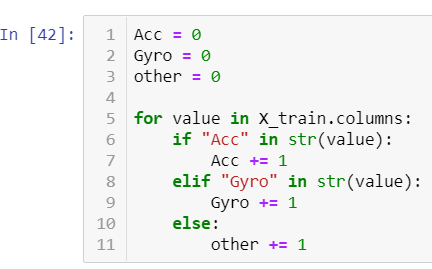
It's easy to identify that there are Accelerometer, Gyroscope and some other values in the dataset.

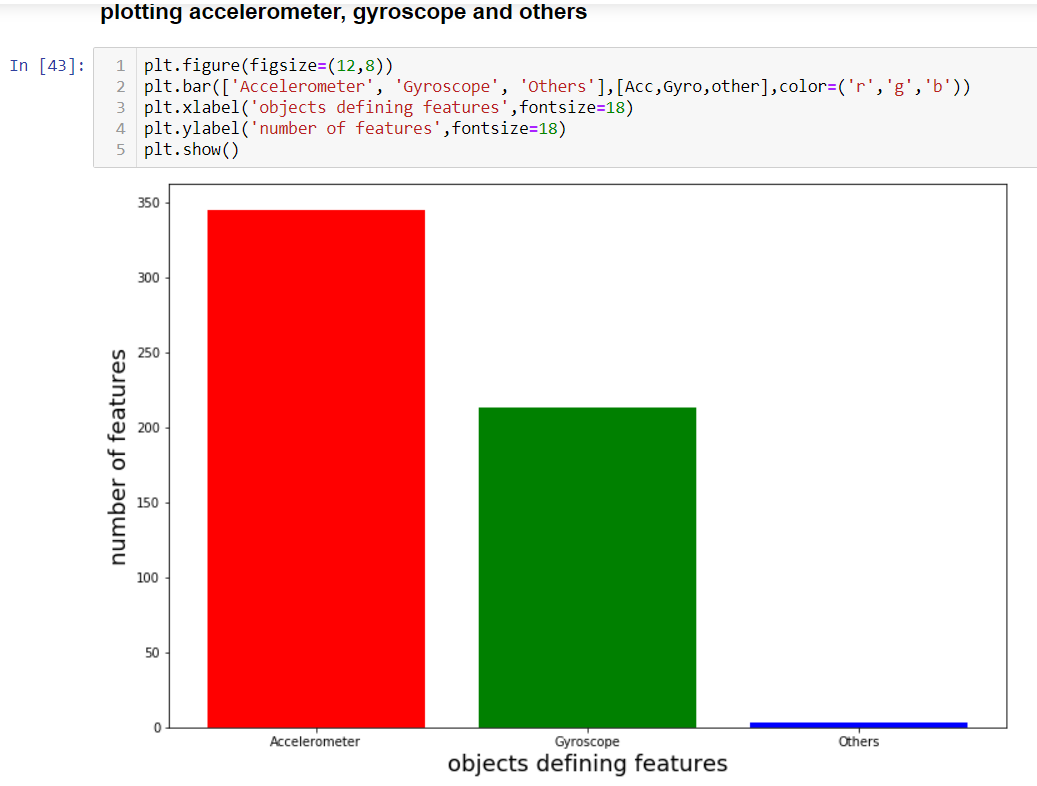
I can check the share of each by plotting a bar graph of each type.

**Accelerometer** values have Acc in them

**Gyroscope** values have Gyro

rest can be considered as others

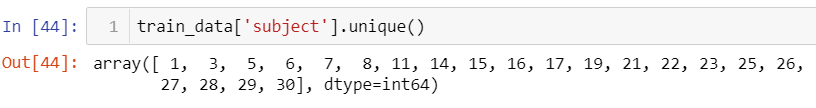




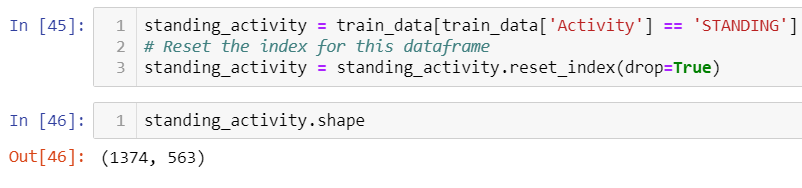
Accelerometer constitutes the maximum features.

Gyroscope comes next.

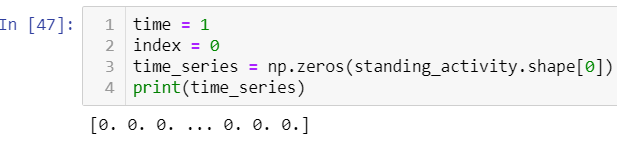
Other features are very less.



Now,I'm selecting rows from the dataset that have the ‘Activity’ label as ‘STANDING’ and store it in standing\_activity.



### Set time series for each subject



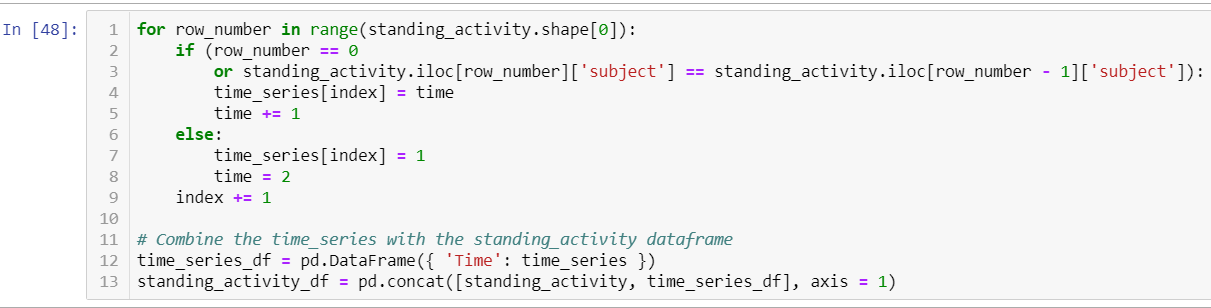
The data collected is in continuous time series for each individual and was recorded at the same rate.

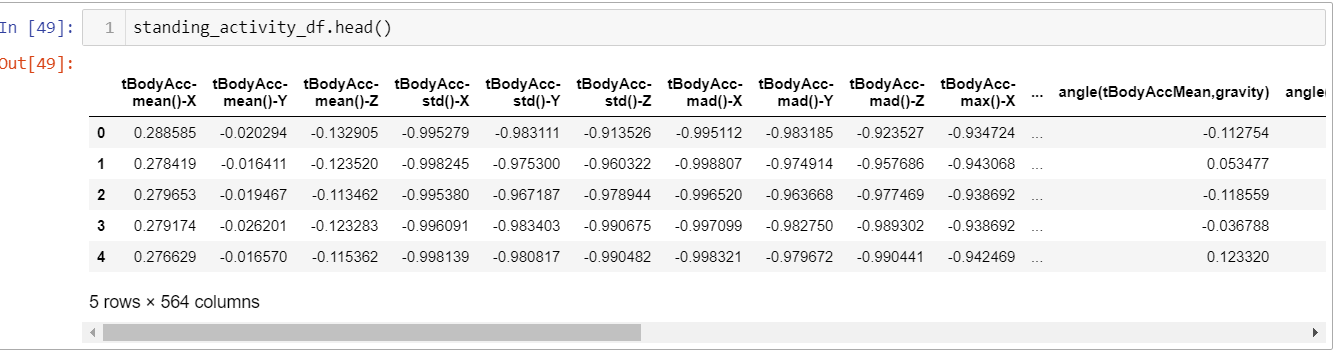
So, I can simply assign time values to each activity starting from 0 each time the subject changes.

For each subject, the Standing activity records will start with a time value of 0 and increment by 1 till the previous row’s subject matches the present row’s subject

. I store all the time series in a variable time\_series and convert it into a dataframe using pandas method DataFrame() and store it in a variable time\_series\_df.

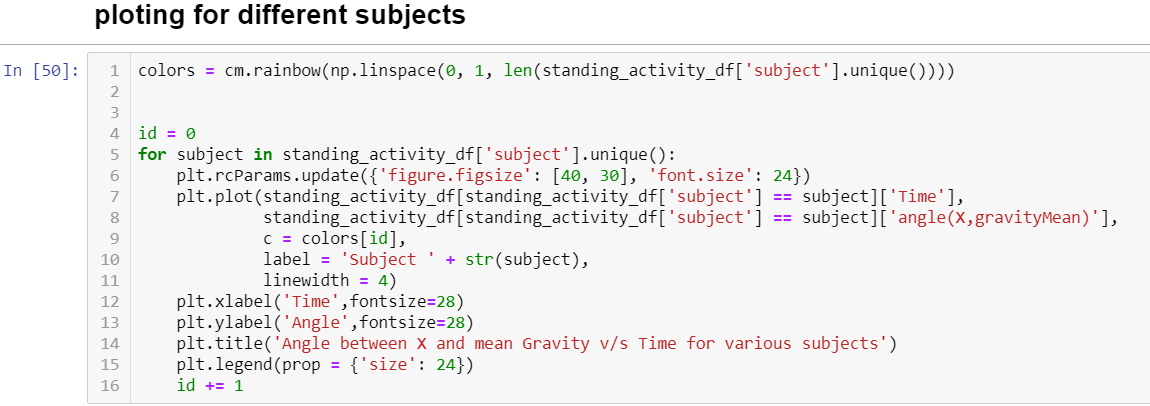
Lastly, I combine the records and the time series variable together in standing\_activity\_df using pandas concatenate() method.

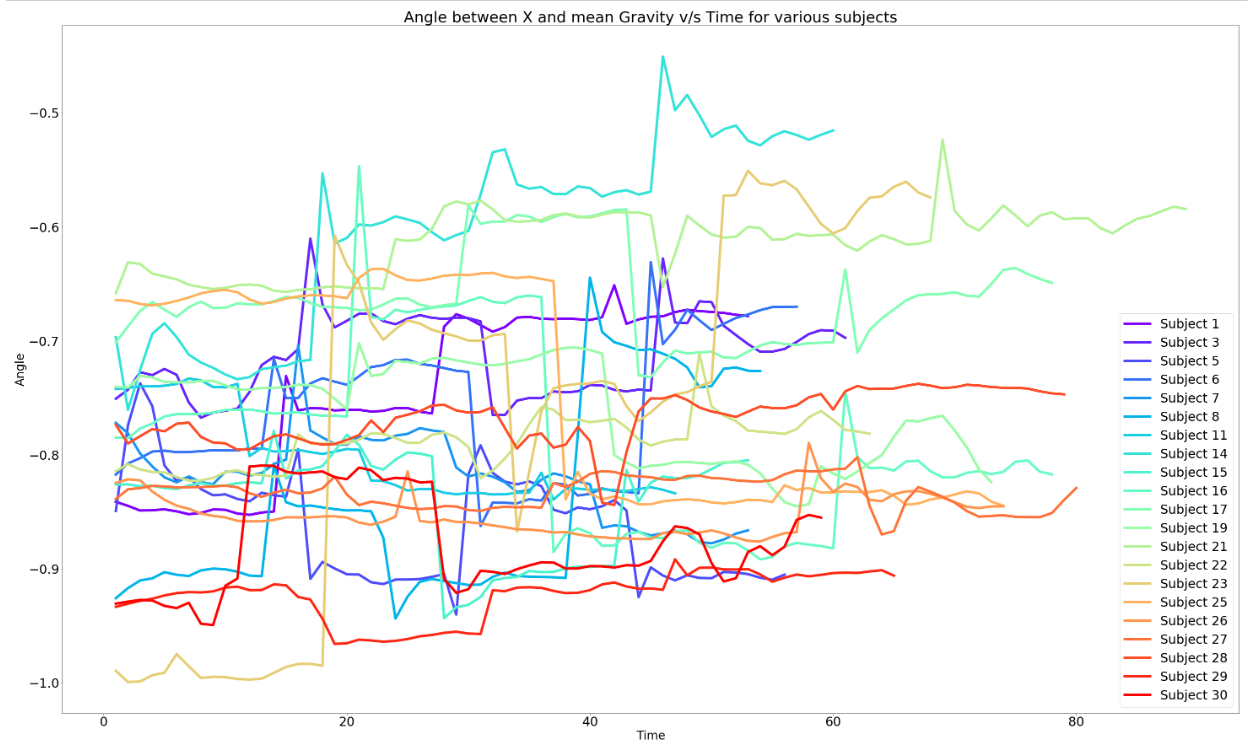




For each subject, I can now plot the graph of their angles with time. I use the subpackage of matplotlib to get a set of colors which shall be used for differentiating the 30 subjects.

We create plot for each subject, which will all be displayed overlapping on one plot.





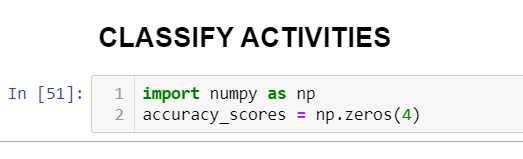
If I take a closer look at the graph,each colour represents one subject and we can see that each line on an average, transitions between a maximum range of 0.2–0.3 values. This is indeed the expected behaviour as slight variations can be attributed to minor human errors.

## **Classify activities**

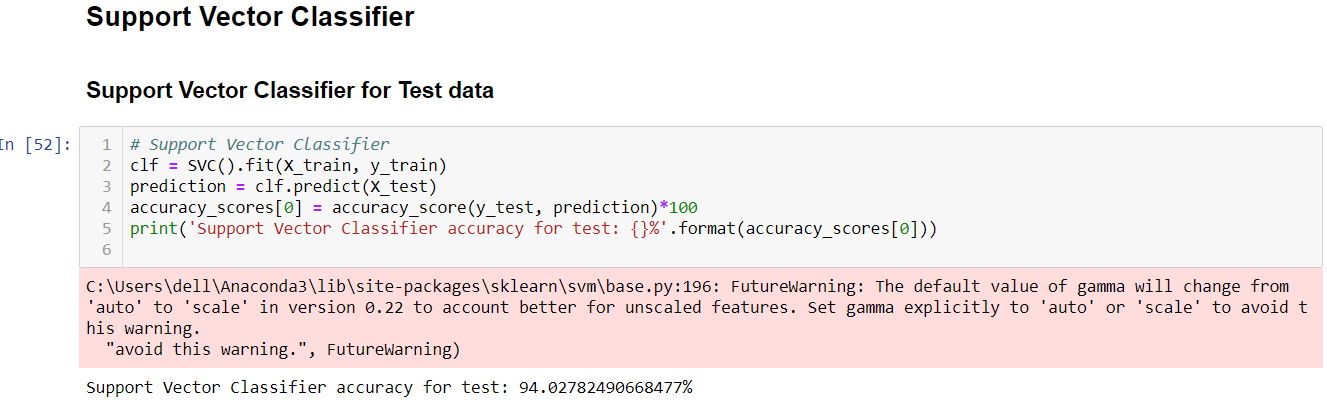
To begin, I'll use various machine learning algorithms available inside the sklearn package that I have already imported.

For each algorithm, I'll calculate the accuracy of prediction and identify the most accurate algorithm.

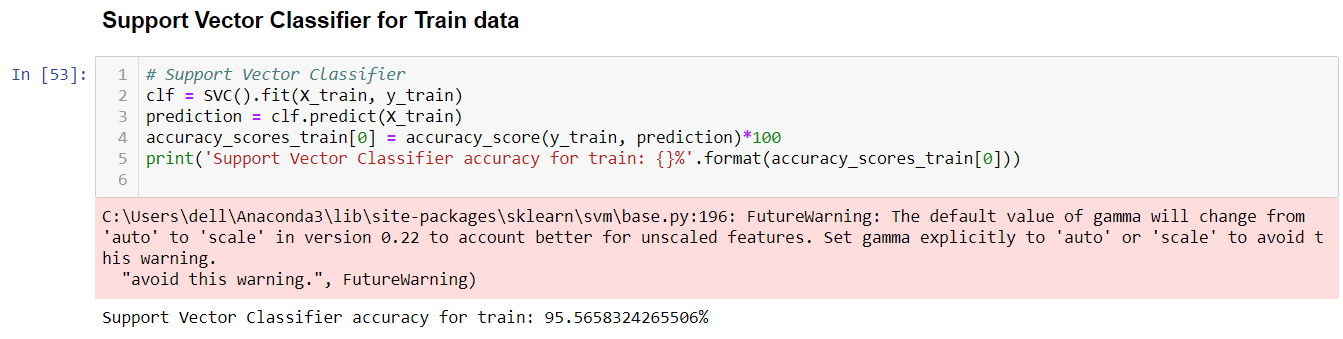
For now, I will keep the default values of parameters as defined in sklearn for each classifier.



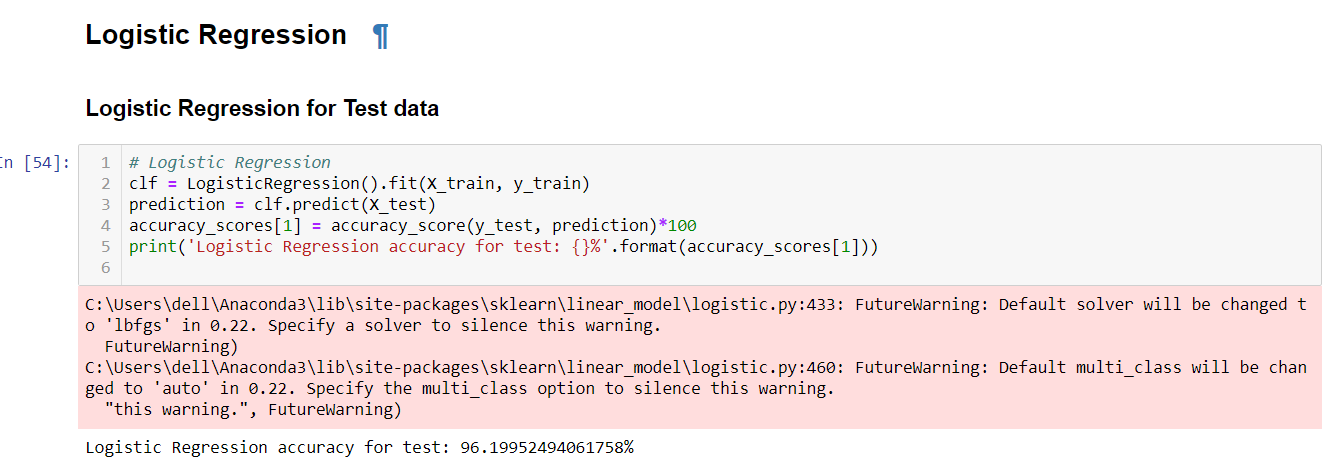
I am using 4 algorithms to predict the accuracy of my train and test data



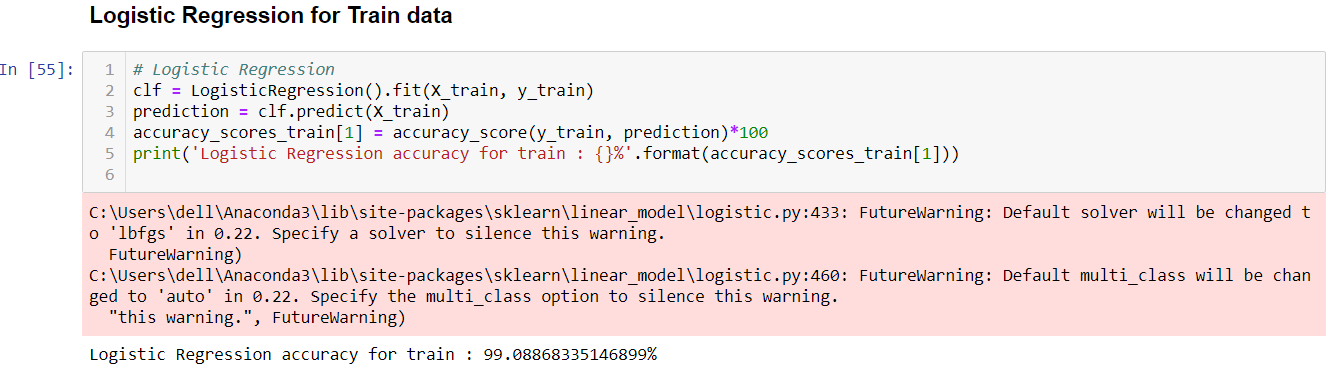
Using Support Vector Classifier the accuracy for test data is 94%



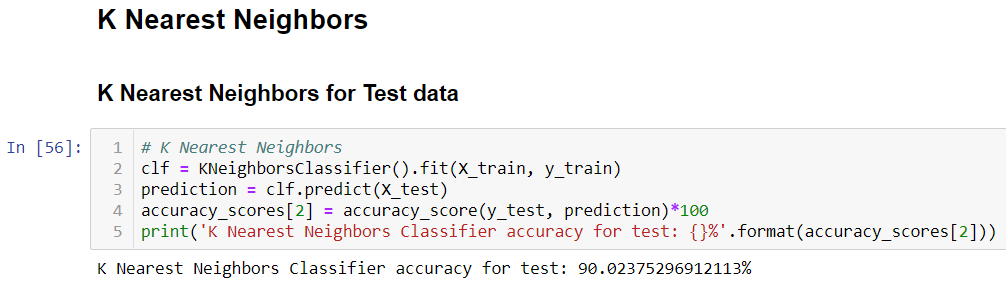
Using Support Vector Classifier the accuracy for train data is 95%



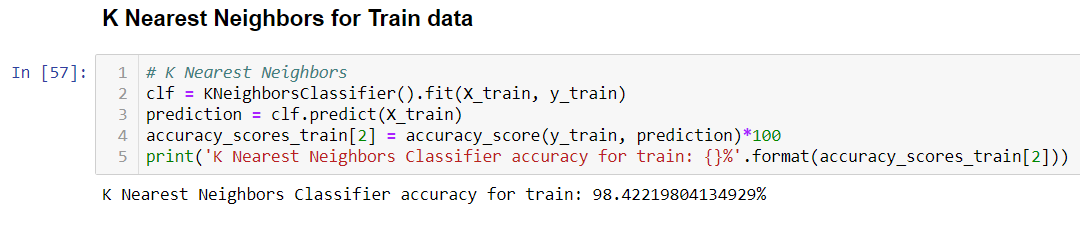
Using Logistic Regression the accuracy for test data is 96%



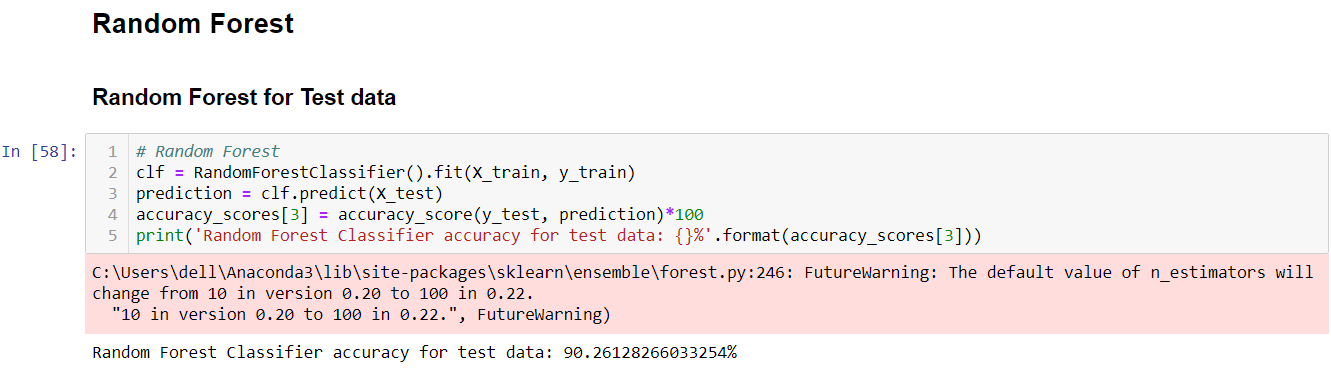
Using Logistic Regression the accuracy for train is 99%



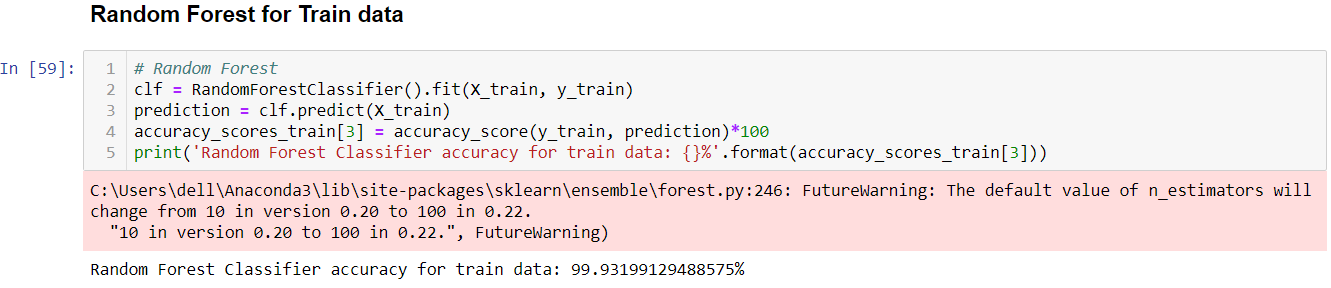
Using K Nearest Neighbours the accuracy of test was 90%



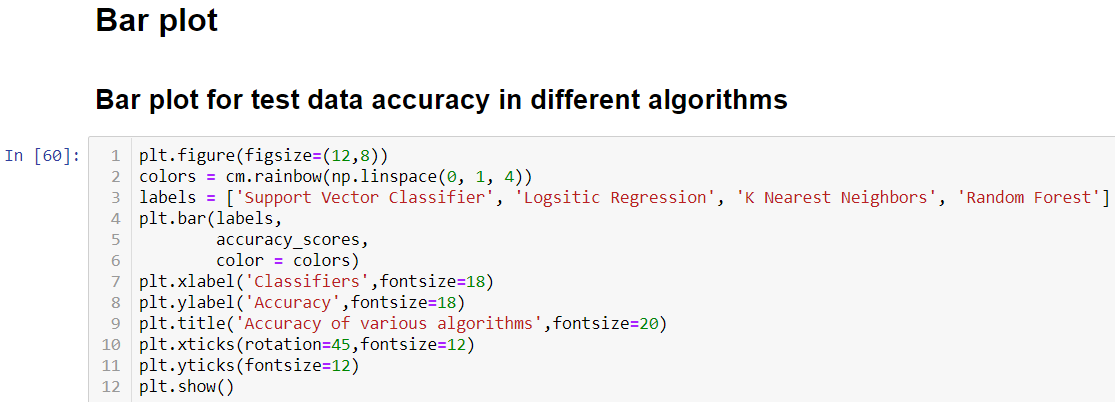
Using K Nearest Neighbors the accuracy of train was 98%

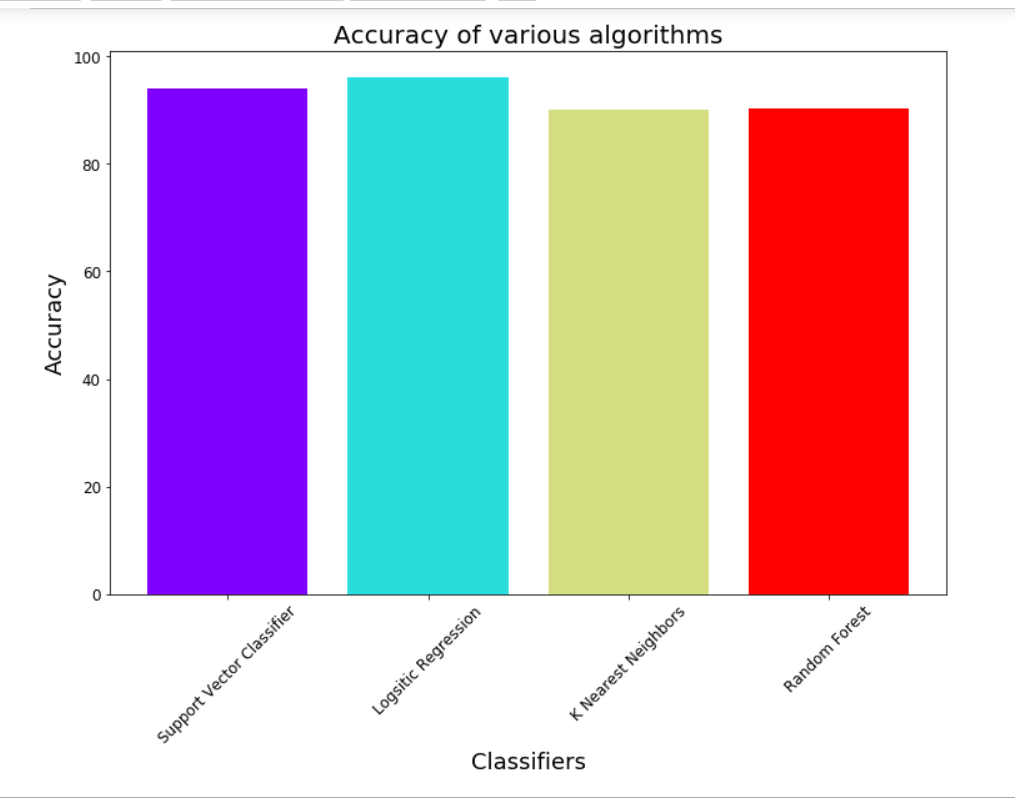


Using Random Forest the accuracy of test was 90%

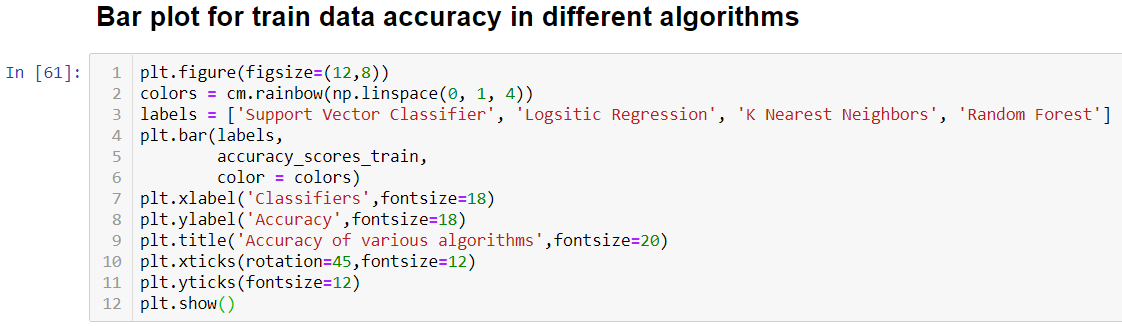


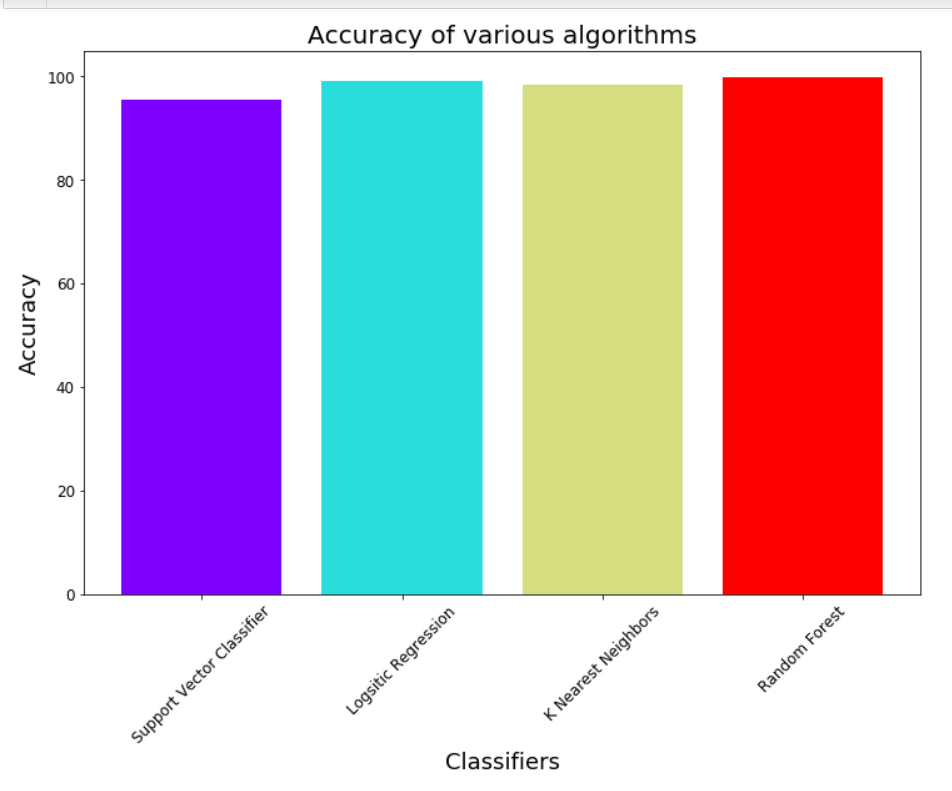
Using Random Forest the accuracy of train was 90%





In the above bar plot i've plotted the accuracy prediction of test data using all the four algorithms where logistic regression predicted the highest accuracy.





In the above bar plot i've plotted the accuracy prediction of train data using all the four algorithms where logistic regression predicted the highest accuracy.

Clearly seen that Logistic Regression performed the best with the highest accuracy of about 99%.

● If the model doesn’t meet the accuracy we can perform mathematical operations on input and output columns, so that it meets the accuracy

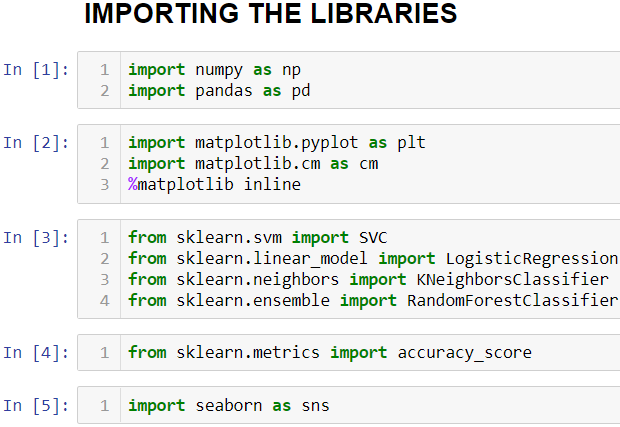
As the accuracy is already high we don't need to perform grid search cv on this.

**4.3 EVALUATING THE CASE STUDY:**

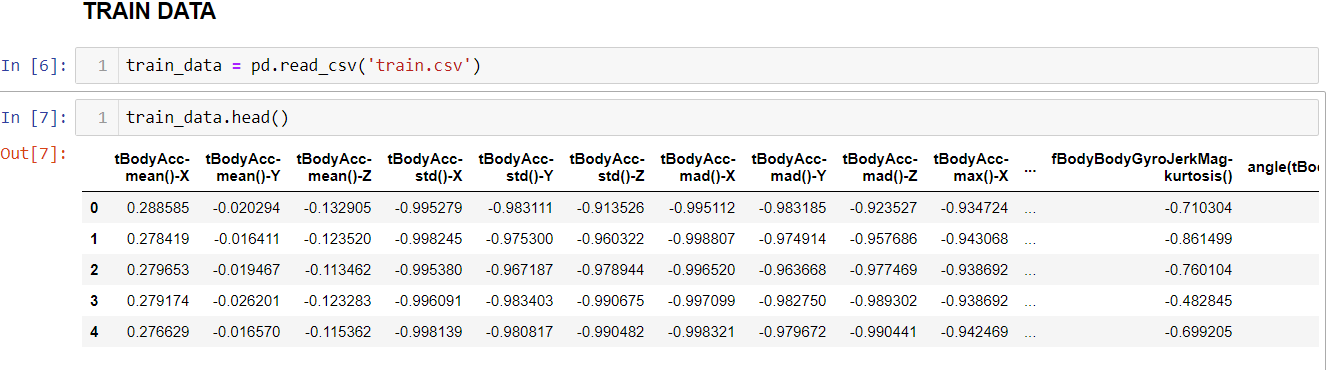
**● LOGISTIC REGRESSION:**

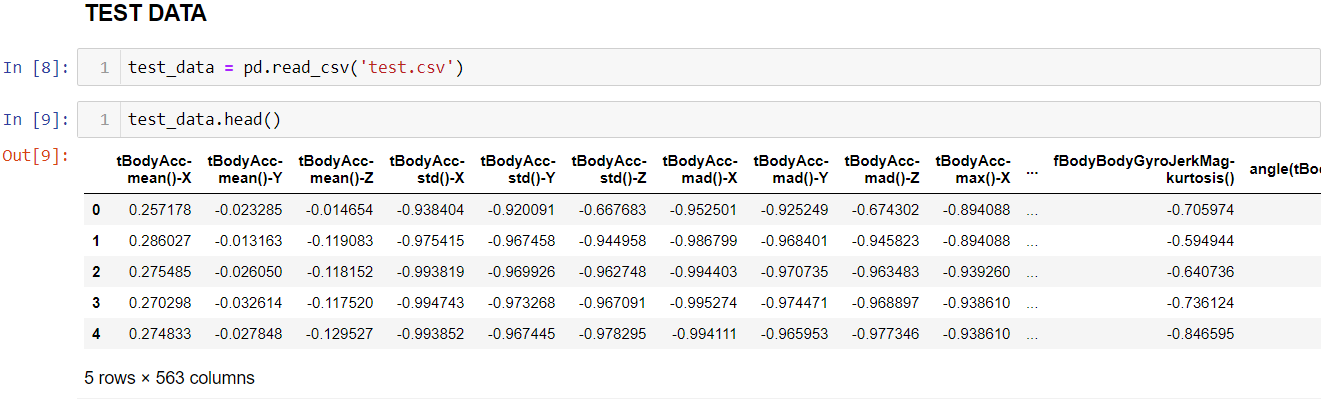
## Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable , although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is [e](https://en.wikipedia.org/wiki/Estimation_theory)stimating the parameters of a logistic model . Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1".

● Importing the required libraries



● Reading the Data-Set





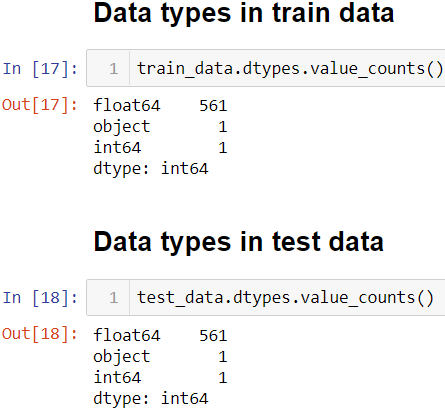
● Handling the missing values

o There is a method called isnull() which gives the number of missing values in each and every column.

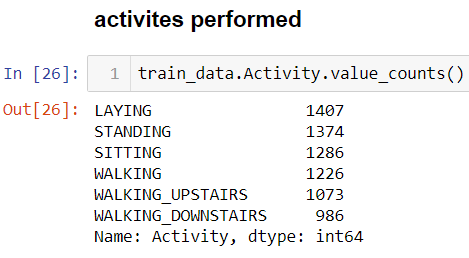
o Using fillna() method each and every missing value is replaced by 0.

## 

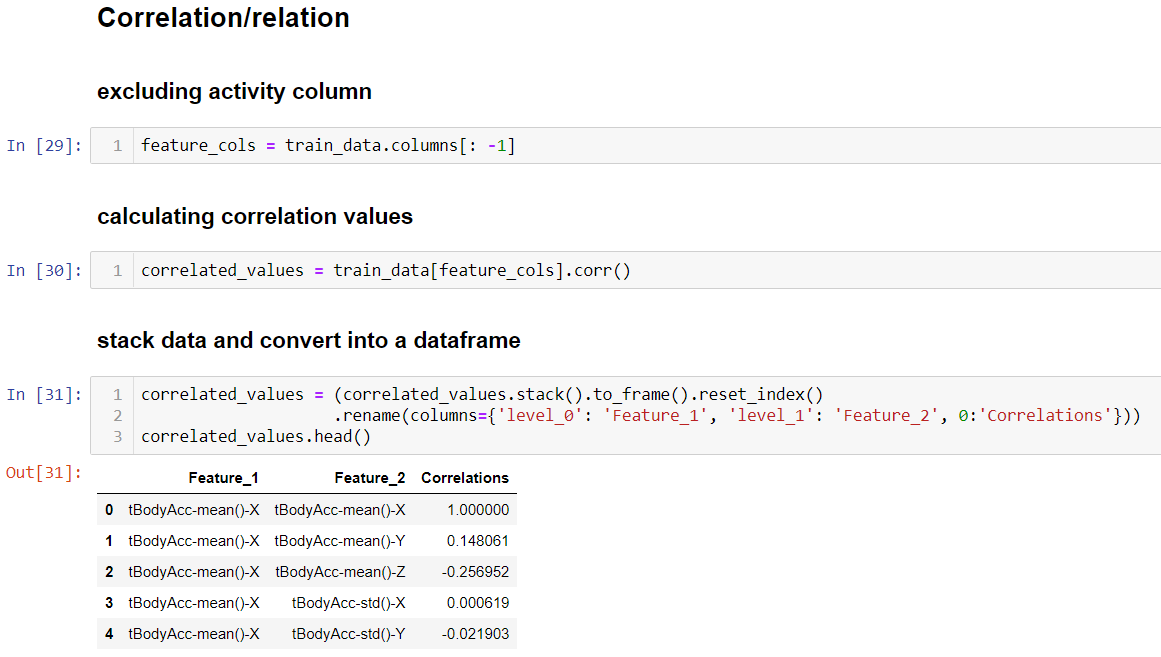
Data types



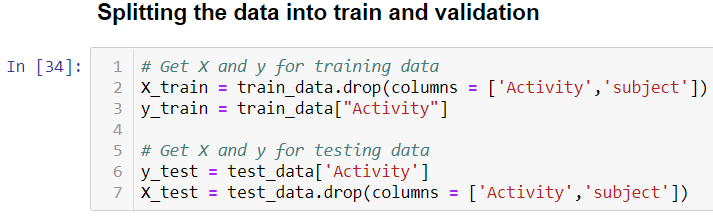
Knowing about the Activity column



Calculating Correlation between the data

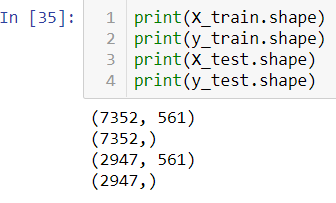


Splitting data

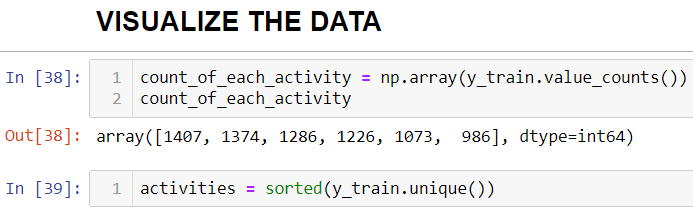


In the above splitting we drop subject column,this is because the values of the subject are all 1 in the dataset.Hence,even if we drop it there would be no change to the dataset

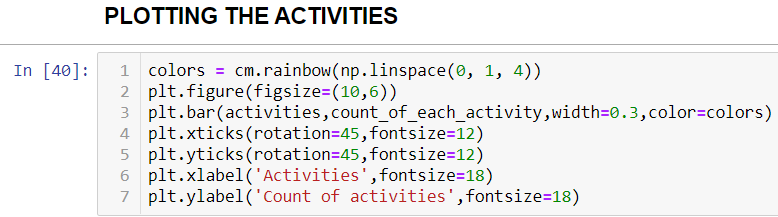
We find the no of rows and columns of the split data using shape

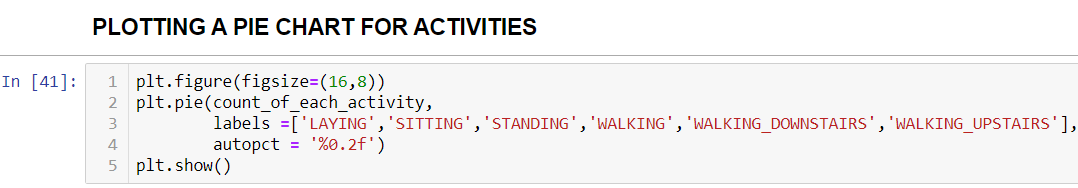


## Visualize the data

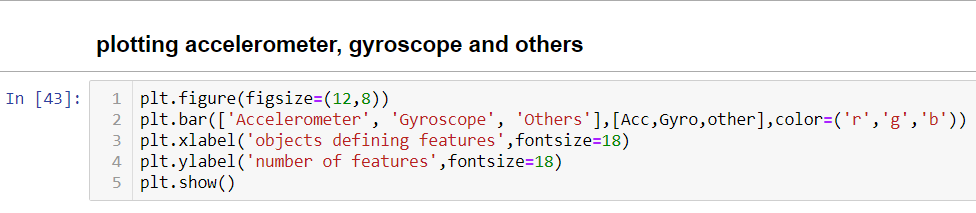


Plot the data

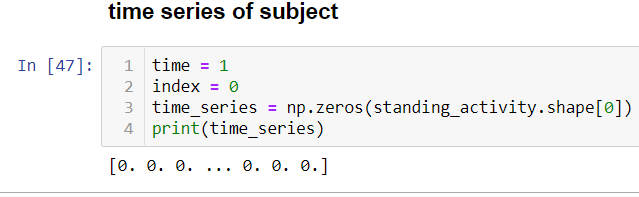




Plotting the accelerometer and gyroscope



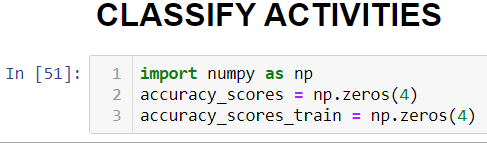
Checking time series



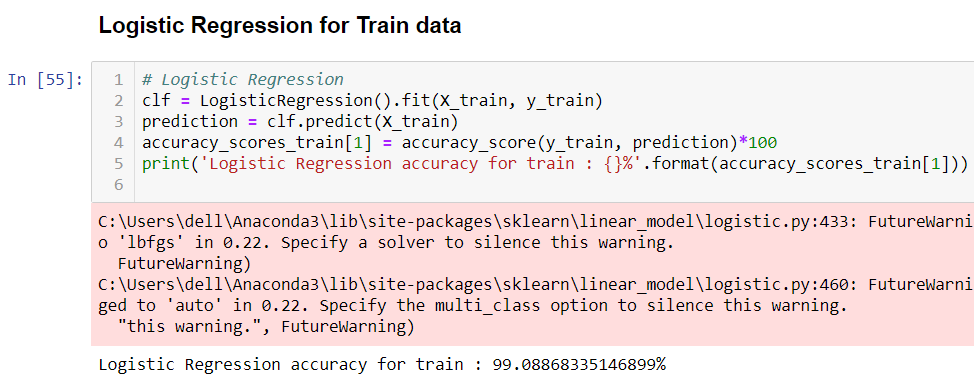
Calculating and printing standing\_activity

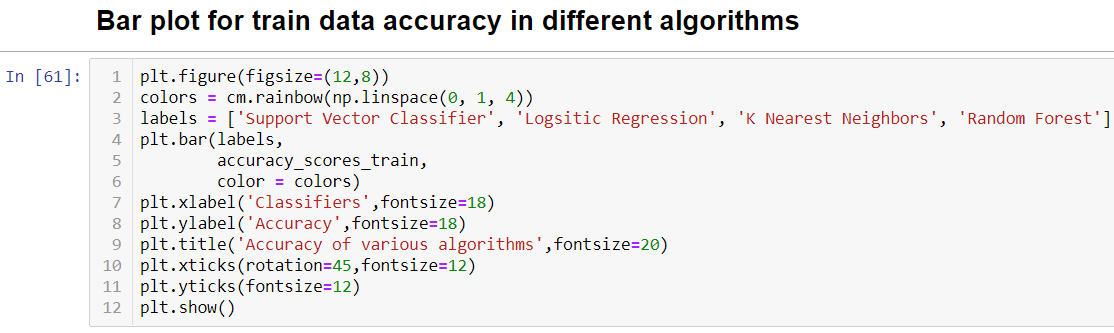
## 

Classifying Activities



## 





Once the bar plot is plotted it shows that logistic regression has highest accuracy (99%) .

If the accuracy doesn't meet our requirements we need to perform certain mathematical operations to get the best accuracy,but in this case because the accuracy is already high we need not perform any operations.

## **Conclusion:**

In this particular project, I explored the activity recognition dataset. I visualized the data using matplotlib. Then, I applied numerous machine learning algorithms and found out that Logistic Regression performed the best in classifying different activities with accuracy of almost 99%