INTERNSHIP PROGRAM REPORT

[AIML]

Industry Partner – Tevatron Technologies Pvt Ltd

Hexnbit Online Internship
www.hexnbit.com

Intern Detail

Name : 6 Weeks Internship Program in AIML

Name of Intern: Sellamuthu Abishek

Intern's Ph. no.: 9491660050

Intern's Add.: D NO. 20-6-184,ROYAL PAVILLION APTS,FLAT NO. 504, 5TH LANE ,RAMALINGESWARA PETA , VIJAYAWADA, LOTUS LANDMARK, Vijayawada, Andhra Pradesh - 520003

Name of College: SRM University AP

College's Add.: Mangalagiri Neerukonda Tadikonda Rd, Mangalagiri, Mandal, Andhra Pradesh 522502

Branch: BTech 2nd Year CSE

Industry Mentor: Mr. Gagan Preet Singh

Designation: R&D Head

Company: Hexnbit EdTech Pvt. Ltd

Email ID: training@hexnbit.com

Contact Number: +91-9818894299

Table of content

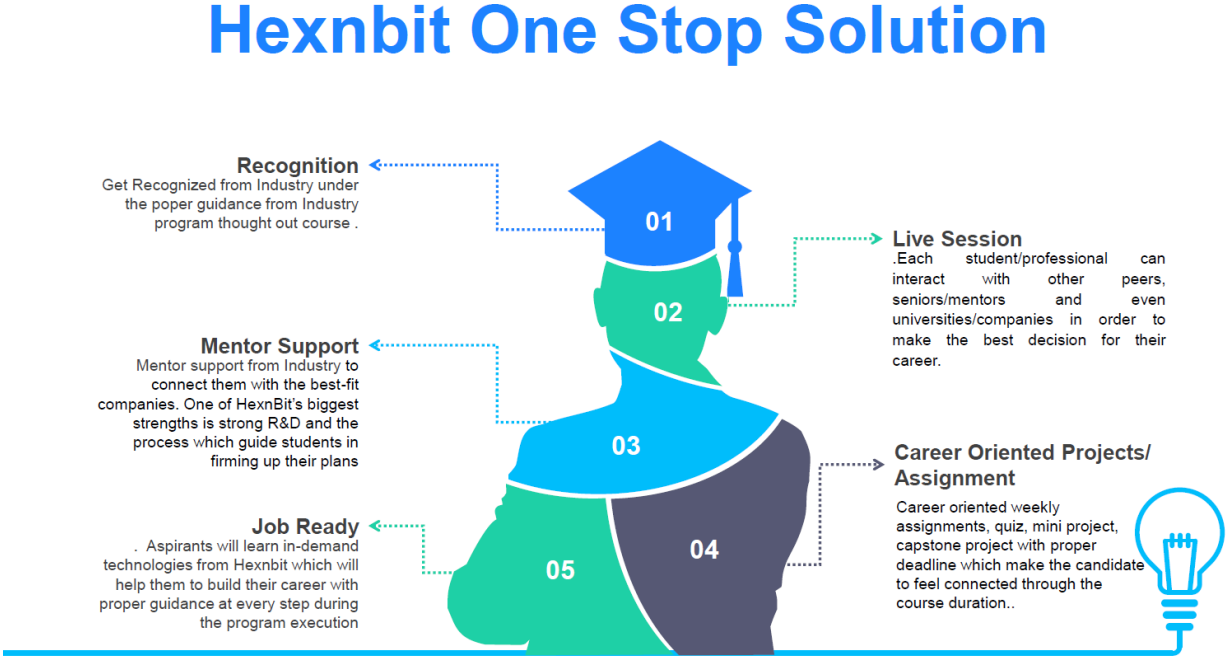
S.No.	Headings	Page No.
1	About Company	1
2	About AIML	3 - 19
3	Computer Vision	20
4	Project Report	21 - 36
5	References	37

About Tevatron Technologies Pvt. Ltd

Tevatron Technologies Pvt Ltd is a **R&D Design & Services** Company focused on **Artificial Intelligence & Machine learning, Internet of Things (IoT), Embedded Hardware and Software Systems, Sensor and MCUs, VLSI Chip Design and PCB Design covering entire ESDM** space from concept to Productization. We are also a member of **IESA(Indian Electronics Semiconductor Association)**. We are actively working towards **#Make in India** as well as **#Design In India** based initiatives

We have in collaboration with our training division **"Hex N Bit"** to provide Online Industrial Internship program. We have added few out of the box features in our programs like: **Project Tracking using Project Management Tools, Access to Faculty for Monitoring, Mentorship during Project, Live Sessions for Understanding of Concepts, Weekly Assignments to keep track of progress, Industry Expert Webinars, eLearning Modules and Assessments**.

- Hexnbit (an ISO certified company) is India's first Ed-tech company which provides one-stop solutions for the students/Professionals in Industry-connect Skill development courses. The platform not only provides subject expertise to the candidates but also, give them industry exposure to apply their learnings analytically in a practical real-world.
- Having registered over 60,000+ candidates, tied up with 250+ universities, 60+ mentors and 10+ Industry, the company aims to bridge the gap between the academics & Industry by providing them with practical knowledge & analytical skills under one umbrella.
- To reach out to the candidates all over the globe, the company has formed Labs in many institutions so any candidate can learn & grow technically with the mentor support (On-Site support as well as virtual support)
- **Hexnbit is now recognised worldwide recognised by STMicroelectronics (https://www.st.com/en/support-and-applications/technical-training-on-stm32.html)**

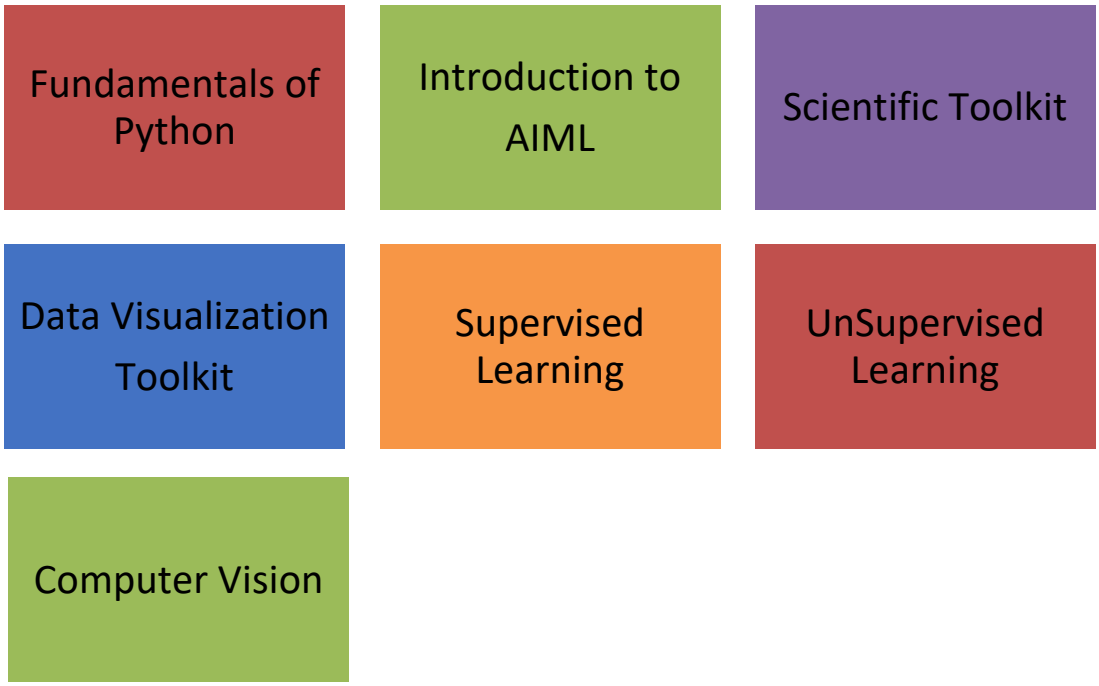


List of Software:

- Anaconda Navigator
- Jupyter Notebook

List of Modules/Libraries:

- Numpy
- Pandas
- Matplotlib
- Seaborn
- Scikit-Learn
- OpenCV
- Os Module


Fundamentals of Python
Introduction to AIML
Scientific Toolkit
Data Visualization Toolkit
Supervised Learning
UnSupervised Learning
Computer Vision

Description:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactic constructions than other languages.

Characteristics of Python Programming:

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Applications:

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.
-

Topics Covered:

- Getting around Anaconda and Jupyter Notebook
- Python Basics
- Data Types
- Conditional Statements, Loops and Control Statements
- Functions
- Lambda Functions and other built-in functions
- File Handling

Learning Outcome:

In this module, exposure was given around the fundamentals of Python Programming to build up a strong foundation. A strong programming foundation will be helpful in data preparation, data preprocessing, analyzing, etc.

What is AIML?

AIML stands for Artificial Intelligence Markup Language. AIML was developed by the Alicebot free software community and Dr. Richard S. Wallace during 1995-2000. It is an XML based markup language meant to create artificial intelligent applications.

AIML is used to create or customize Alicebot which is a chat-box application based on A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) free software.

Why AIML?

AIML makes it possible to create human interfaces while keeping the implementation simple to program, easy to understand and highly maintainable.

What is Artificial Intelligence?

Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, speech recognition and machine vision.

This aspect of AI programming focuses on acquiring data and creating rules for how to turn the data into actionable information. The rules, which are called algorithms, provide computing devices with step-by-step instructions for how to complete a specific task.

Advantages of AI:

- Good at detail-oriented jobs.
- Reduced time for data-heavy tasks.
- Delivers consistent results; and
- AI-powered virtual agents are always available.

AI used for?

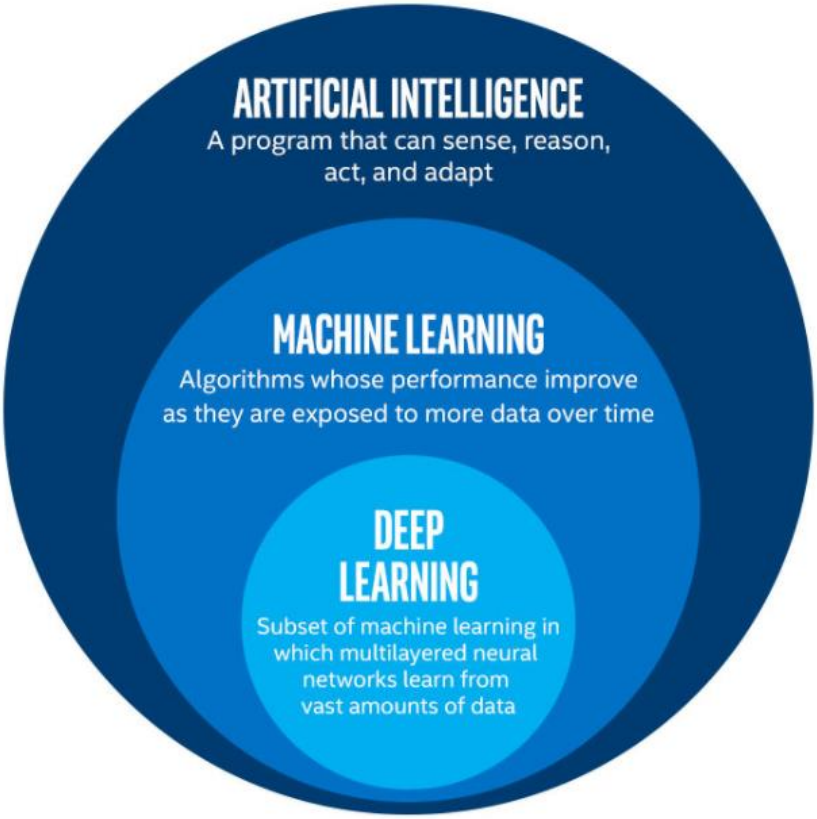
- AI in healthcare.
- AI in business.
- AI in education.
- AI in finance

What is Machine learning?

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Why Machine learning?

Machine learning is an important component of the growing field of data science. Using statistical methods, algorithms are trained to make classifications or predictions, uncovering key insights within data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow.



Learning Outcome:

In this module, Introduction to AIML , basic libraries such as NumPy, Pandas, Data Visualization and known machine learning algorithms were given.

What is Numpy?

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

At the core of the NumPy package, is the Nd array object. This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences:

- NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of an ndarray will create a new array and delete the original.
- The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sized elements.
- NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.
- A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays. In other words, to efficiently use much (perhaps even most) of today's scientific/mathematical Python-based software, just knowing how to use Python's built-in sequence types is insufficient - one also needs to know how to use NumPy arrays.
- The points about sequence size and speed are particularly important in scientific computing. As a simple example, consider the case of multiplying each element in a 1-D sequence with the corresponding element in another sequence of the same length.

What is Pandas?

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like ActiveState's ActivePython.

Library Highlights:

- A fast and efficient DataFrame object for data manipulation with integrated indexing.
- Tools for reading and writing data between in-memory data structures and different formats: CSV and text files, Microsoft Excel, SQL databases, and the fast HDF5 format.
- Intelligent data alignment and integrated handling of missing data: gain automatic label-based alignment in computations and easily manipulate messy data into an orderly form.
- Flexible reshaping and pivoting of data sets.
- Intelligent label-based slicing, fancy indexing, and subsetting of large data sets.
- Columns can be inserted and deleted from data structures for size mutability.
- Aggregating or transforming data with a powerful group by engine allowing split-apply-combine operations on data sets.
- High performance merging and joining of data sets.
- Hierarchical axis indexing provides an intuitive way of working with high-dimensional data in a lower-dimensional data structure.
- Time series-functionality: date range generation and frequency conversion, moving window statistics, date shifting and lagging. Even create domain-specific time offsets and join time series without losing data.
- Python with pandas is in use in a wide variety of academic and commercial domains, including Finance, Neuroscience, Economics, Statistics, Advertising, Web Analytics, and more.

Topics Covered:

- **NumPy**
  NumPy Basics
  Operations
  Indexing, Slicing and Copies

- **Pandas**
  Series
  DataFrames
  Fix Missing Data
  GroupBy
  Merge
  Operations
  File Reading and Writing

Learning Outcome:

In this module, Introduction to Numpy and Pandas library was given that how these libraries can be helpful in preparing and cleaning data, processing the data, bringing in the insights about the data, data analysis, understanding important features related to business domain, etc.

What is Data Visualization?

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

Why is Data Visualization important?

Because of the way the human brain processes information, using charts or graphs to visualize large amounts of complex data is easier than poring over spreadsheets or reports. Data visualization is a quick, easy way to convey concepts in a universal manner – and you can experiment with different scenarios by making slight adjustments.

Data visualization can also:

- Identify areas that need attention or improvement.
- Clarify which factors influence customer behavior.
- Help you understand which products to place where.
- Predict sales volumes.

Common general types of data visualization:

- Charts
- Tables
- Graphs
- Maps
- Infographics
- Dashboards
-

Data visualization can be used for:

- Making data engaging and easily digestible.
- Identifying trends and outliers within a set of data.
- Telling a story found within the data.
- Reinforcing an argument or opinion.
- Highlighting the important parts of a set of data.

What is Matplotlib?

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

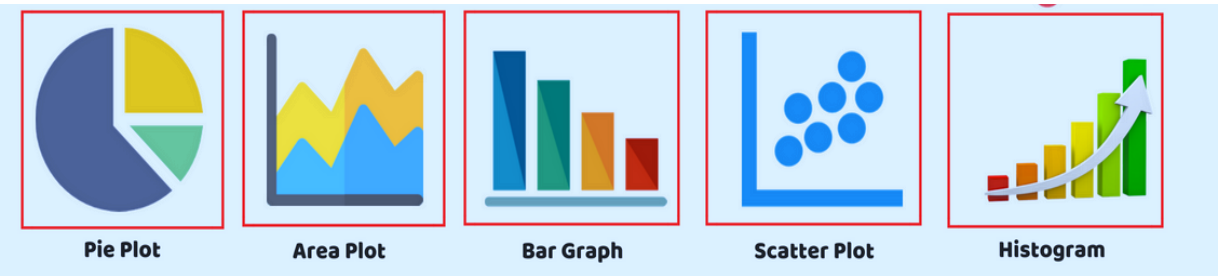
A Python matplotlib script is structured so that a few lines of code are all that is required in most instances to generate a visual data plot. The matplotlib scripting layer overlays two APIs:

- The pyplot API is a hierarchy of Python code objects topped by matplotlib.pyplot
- An OO (Object-Oriented) API collection of objects that can be assembled with greater flexibility than pyplot. This API provides direct access to Matplotlib's backend layers.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Python Matplotlib : Types of Plots

There are various plots which can be created using python matplotlib. Some of them are listed below:



Learning Outcome:

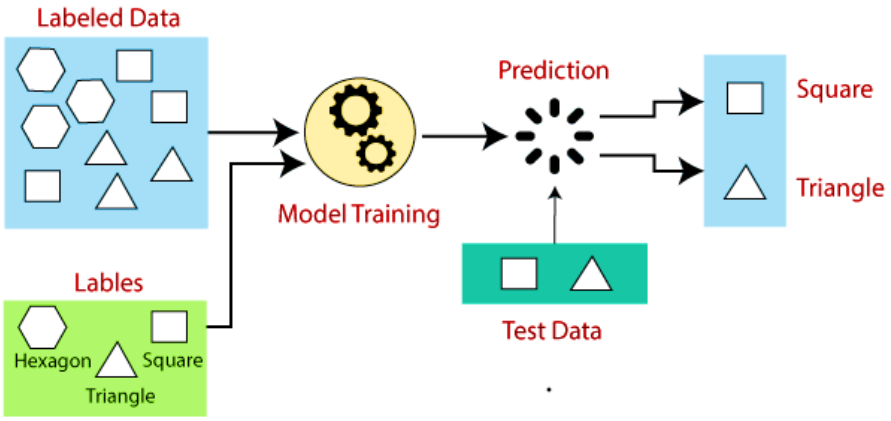
In this module, Introduction to Data Visualization was given that how these libraries can be helpful in better understanding of data, bringing in the insights about the data, understanding important features related to business domain, representing facts in a compact manner to the stakeholders, etc.

What is Supervised Machine Learning?

- Supervised learning is the type of machine learning in which machines are trained using well "labelled" training data, and on the basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.
- In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.
- Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).
- In the real-world, supervised learning can be used for Risk Assessment, Image classification, Fraud Detection, spam filtering, etc.

How Supervised Learning Works?

- In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.



Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.

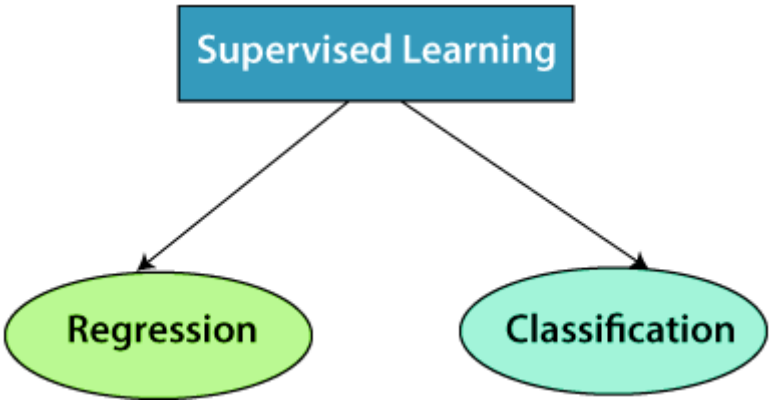
- If the given shape has four sides, and all the sides are equal, then it will be labelled as a Square.
- If the given shape has three sides, then it will be labelled as a triangle.
- If the given shape has six equal sides then it will be labelled as hexagon.

Now, after training, we test our model using the test set, and the task of the model is to identify the shape.

The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output.

Types of supervised Machine learning Algorithms:

Supervised learning can be further divided into two types of problems:



1. Regression

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

- Linear Regression
- Regression Trees
- Non-Linear Regression
- Bayesian Linear Regression
- Polynomial Regression

2. Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

- Spam Filtering,
- Random Forest
- Decision Trees
- Logistic Regression
- Support vector Machines

Advantages of Supervised learning:

- With the help of supervised learning, the model can predict the output on the basis of prior experiences.
- In supervised learning, we can have an exact idea about the classes of objects.
- Supervised learning model helps us to solve various real-world problems such as fraud detection, spam filtering, etc.

Disadvantages of supervised learning:

- Supervised learning models are not suitable for handling complex tasks.
- Supervised learning cannot predict the correct output if the test data is different from the training dataset.
- Training required lots of computation times.
- In supervised learning, we need enough knowledge about the classes of objects.

Topics Covered:

- Machine Learning with Python
- Supervised- Linear Regression
- Supervised- Logistic Regression
- Supervised- Decision Tree
- Supervised- Support Vector Machine
- Supervised– K-Nearest Neighbours

Learning Outcome:

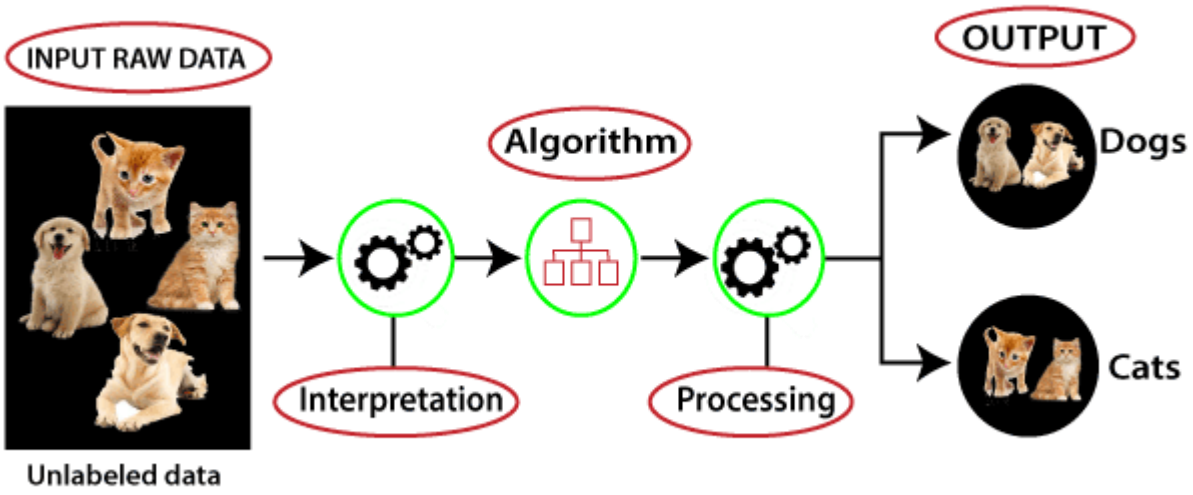
In this module, Introduction to supervised learning was given on how these libraries can be helpful in building up the prediction models using different algorithms for different business use cases, improving the models, etc.

What is Un-Supervised Machine Learning?

- As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things. It can be defined as:
- Unsupervised learning is a type of machine learning in which models are trained using unlabeled datasets and are allowed to act on that data without any supervision.
- Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

How Un-Supervised Learning Works?

Working of unsupervised learning can be understood by the below diagram:

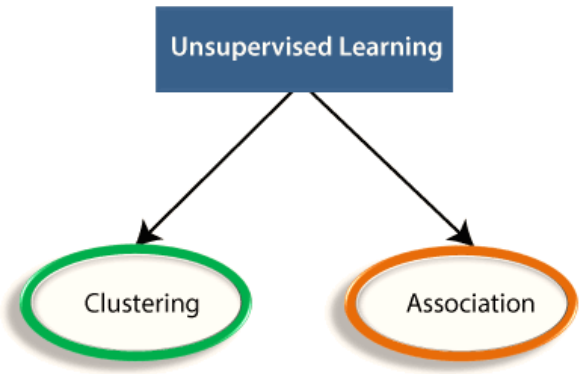


Here, we have taken an unlabeled input data, which means it is not categorized and corresponding outputs are also not given. Now, this unlabeled input data is fed to the machine learning model in order to train it. Firstly, it will interpret the raw data to find the hidden patterns from the data and then will apply suitable algorithms such as k-means clustering, Decision tree, etc.

Once it applies the suitable algorithm, the algorithm divides the data objects into groups according to the similarities and differences between the objects.

Types of Unsupervised Learning Algorithm:

The unsupervised learning algorithm can be further categorized into two types of problems:



1. **Clustering:**
   Clustering is a method of grouping the objects into clusters such that objects with most similarities remain into a group and have less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

2. **Association:**
   An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occur together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

Advantages of Unsupervised Learning

- Unsupervised learning is used for more complex tasks as compared to supervised learning because, in unsupervised learning, we don't have labeled input data.
- Unsupervised learning is preferable as it is easy to get unlabeled data in comparison to labeled data.

Disadvantages of Unsupervised Learning

- Unsupervised learning is intrinsically more difficult than supervised learning as it does not have corresponding output.
- The result of the unsupervised learning algorithm might be less accurate as input data is not labeled, and algorithms do not know the exact output in advance.

Supervised vs Un-Supervised Learning

Supervised Learning	Un-Supervised Learning
Supervised learning algorithms are trained using labeled data.	Unsupervised learning algorithms are trained using unlabeled data.
Supervised learning model takes direct feedback to check if it is predicting correct output or not.	Unsupervised learning model does not take any feedback.
Supervised learning model predicts the output.	Unsupervised learning model find the hidden patterns in data.
In supervised learning, input data is provided to the model along with the output.	In unsupervised learning, only input data is provided to the model.
The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset
Supervised learning can be categorized in Classification and Regression problems.	Unsupervised Learning can be classified in Clustering and Associations problems.
Supervised learning can be used for those cases where we know the input as well as corresponding outputs.	Unsupervised learning can be used for those cases where we have only input data and no corresponding output data.
It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc.	It includes various algorithms such as Clustering, KNN, and Apriori algorithm.

What is Computer Vision?

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos, and other visual inputs — and take actions or make recommendations based on that information. If AI enables computers to think, computer vision enables them to see, observe and understand.

Computer vision works much the same as human vision, except humans have a head start. Human sight has the advantage of lifetimes of context to train how to tell objects apart, how far away they are, whether they are moving and whether there is something wrong in an image.

How does computer vision work?

Computer vision needs lots of data. It runs analyses of data over and over until it discerns distinctions and ultimately recognizes images. For example, to train a computer to recognize automobile tires, it needs to be fed vast quantities of tire images and tire-related items to learn the differences and recognize a tire, especially one with no defects.

- **Image classification** sees an image and can classify it (a dog, an apple, a person's face). More precisely, it can accurately predict that a given image belongs to a certain class. For example, a social media company might want to use it to automatically identify, and segregate objectionable images uploaded by users.
- **Object detection** can use image classification to identify a certain class of image and then detect and tabulate their appearance in an image or video. Examples include detecting damages on an assembly line or identifying machinery that requires maintenance.
- **Object tracking** follows or tracks an object once it is detected. This task is often executed with images captured in sequence or real-time video feeds. Autonomous vehicles, for example, need to not only classify and detect objects such as pedestrians, other cars, and road infrastructure, they need to track them in motion to avoid collisions and obey traffic laws.
- **Content-based image retrieval** uses computer vision to browse, search and retrieve images from large data stores, based on the content of the images rather than metadata tags associated with them. This task can incorporate automatic image annotation that replaces manual image tagging. These tasks can be used for digital asset management systems and can increase the accuracy of search and retrieval.

Parkinson's disease

What is Parkinson's disease?

Parkinson's disease is a progressive nervous system disorder that affects movement. Symptoms start gradually, sometimes starting with a barely noticeable tremor in just one hand. It has 5 stages to it and affects more than 1 million individuals every year in India. This is chronic and has no cure yet. It is a neurodegenerative disorder affecting dopamine-producing neurons in the brain.

Objective:

To build a machine learning model that accurately detects the presence of Parkinson's disease in an individual. This project deals with a real-world medical issue that can help doctors determine if a person has Parkinson's and how the disease is likely to progress.

Dataset for the Project:

A dataset file for the project is - "parkinsons.data" and the attributes file for the dataset is "parkinsons.names". We converted the dataset from .data into a .csv file to load in the project.

Prerequisites

- Python
- Jupyter Lab - Notebook
- Installed Python libraries(Using command prompt)

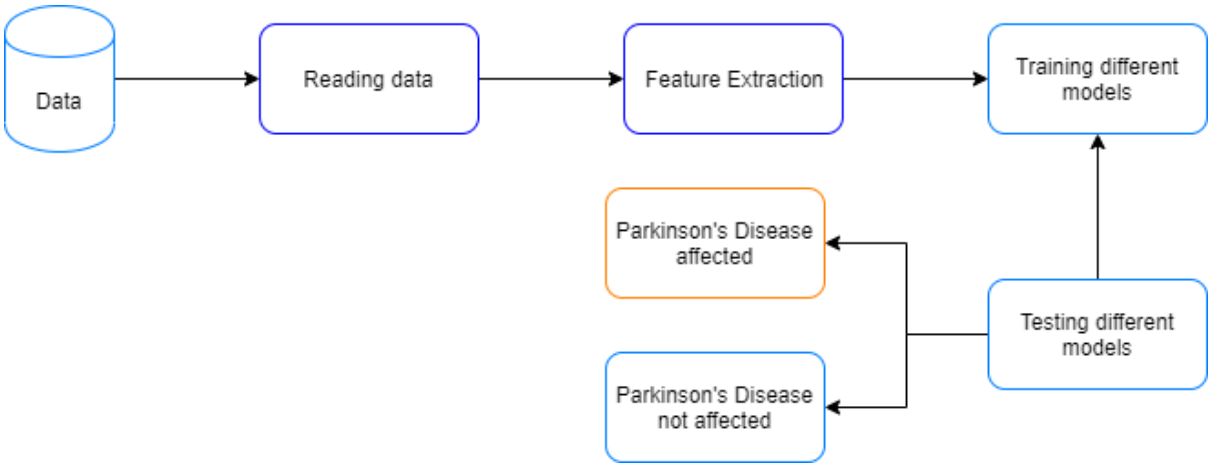
Python Libraries used:

- Numpy
- Pandas
- Matplotlib
- Seaborn
- Scikit-Learn
- Os Module

Algorithm:

1. Load the dataset into a dataframe using pandas
2. Check for any missing data.
3. Analyze the data to make observations like correlation, etc and visualize them
4. Drop unwanted data
5. Standardise the data to improve its quality
6. Split the dataset into training and testing data (80% training 20% testing)
7. Train different models with training data using each classifier's object and check their score
8. Store the names of models and their predictions in different variables
9. Create another variable to store the accuracy of all the models
10. Append the accuracy scores of the models into variable created in Step 9 using a for loop
11. Create a dataframe and store the names and accuracy scores of all models and display it
12. Print the classification report of all models
13. Create a plot and plot the confusion matrix of all the models
14. Create a barplot of all the model's names and their accuracy

Block Diagram:



1.Importing Libraries and Reading Dataset:

We need to import the libraries required to build the model and execute in the first cell of the jupyter notebook.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sn
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report,confusion_matrix
import warnings
warnings.filterwarnings('ignore')
```

After Importing the libraries we are loading the previously converted parksions.csv into the program with the help of pandas and read it as 'df'.

```
df = pd.read_csv("parkinsons.csv")
```

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	...	Shimm
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	...	
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	...	
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	...	
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	...	
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	...	
...	...	...	...	...	...	...	...	...	...	...	...	...
190	phon_R01_S50_2	174.188	230.978	94.261	0.00459	0.00003	0.00263	0.00259	0.00790	0.04087	...	
191	phon_R01_S50_3	209.516	253.017	89.488	0.00564	0.00003	0.00331	0.00292	0.00994	0.02751	...	
192	phon_R01_S50_4	174.688	240.005	74.287	0.01360	0.00008	0.00624	0.00564	0.01873	0.02308	...	
193	phon_R01_S50_5	198.764	396.961	74.904	0.00740	0.00004	0.00370	0.00390	0.01109	0.02296	...	
194	phon_R01_S50_6	214.289	260.277	77.973	0.00567	0.00003	0.00295	0.00317	0.00885	0.01884	...	

195 rows × 24 columns

After executing the cell the dataset df is displayed and the shape is – 195 rows & 24 Columns.

`df.head()` is used to print the first five rows of data.

2.Checking for Missing Data:

We need to check for any NaN values inside the dataset so that there is no error while performing the training of the model.

`df.isnull().sum()` is used to check null values in Pandas DataFrame and sum is the total number of NaN values.

-- There are no null values in the dataset as the sum of NaN values is 0.

3.Handling Categorical Values and Standardization

The unnecessary columns for the training of the model are (names ,status – target) .

The columns of the variables (names and status) were dropped from the dataframe because they are not needed to make predictions.

```
# Dropping name column as it is not required
x = df.drop(columns=['name','status'], axis=1)
#target values
y = df['status']
```

The dependent and independent variables are defined, being y and x respectively. The dependent variable, y is the target and the independent variable, x is the dataframe with the target and names dropped from it.

Data standardization is about making sure that data is internally consistent; that is, each data type has the same content and format. By using sklearn.preprocessing library and

importing StandardScaler we standardize the data helps improve the quality of your data by transforming and standardizing it.

```
from sklearn.preprocessing import StandardScaler
stdscaler = StandardScaler()
X = np.array(stdscaler.fit_transform(x))
```

	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	MDVP:Shimmer(dB)	...	MD
0	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	0.426	...	
1	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	0.626	...	
2	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	0.482	...	
3	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	0.517	...	
4	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	0.584	...	
...	...	...	...	...	...	...	...	...	...	...	...	...
190	174.188	230.978	94.261	0.00459	0.00003	0.00263	0.00259	0.00790	0.04087	0.405	...	
191	209.516	253.017	89.488	0.00564	0.00003	0.00331	0.00292	0.00994	0.02751	0.263	...	
192	174.688	240.005	74.287	0.01360	0.00008	0.00624	0.00564	0.01873	0.02308	0.256	...	
193	198.764	396.961	74.904	0.00740	0.00004	0.00370	0.00390	0.01109	0.02296	0.241	...	
194	214.289	260.277	77.973	0.00567	0.00003	0.00295	0.00317	0.00885	0.01884	0.190	...	

195 rows × 22 columns

X data frame has 22 columns after categorizing values and standardization.

4.Exploratory Data Analytics:

Exploratory Data Analysis refers to the critical process of performing initial investigations on data to discover patterns, to spot anomalies, to test hypotheses and to check assumptions with the help of summary statistics and graphical representations.

EDA is a practice to understand the data first and try to gather as many insights from it.

Shape of the Data:

`df.shape` returns a tuple representing the dimensionality of the Dataset.
The total number of rows and columns in the Dataset. Here in this dataset, it is

```
(195, 24)
```

Columns of the Data:

`df.columns` returns the columns of the dataset.

The columns associated in the dataset are:

```
'name', 'MDVP:Fo(Hz)', 'MDVP:Fhi(Hz)', 'MDVP:Flo(Hz)', 'MDVP:Jitter(%)',


'MDVP:Jitter(Abs)', 'MDVP:RAP', 'MDVP:PPQ', 'Jitter:DDP',
       'MDVP:Shimmer', 'MDVP:Shimmer(dB)', 'Shimmer:APQ3', 'Shimmer:APQ5',
       'MDVP:APQ', 'Shimmer:DDA', 'NHR', 'HNR', 'status', 'RPDE', 'DFA',
       'spread1', 'spread2', 'D2', 'PPE'
```

Describe():

`df.describe().transpose()` returns the count, mean, standard deviation, minimum and maximum values, and the quantiles of the data. Here we transposed the dataset for better view.

Target Variable:

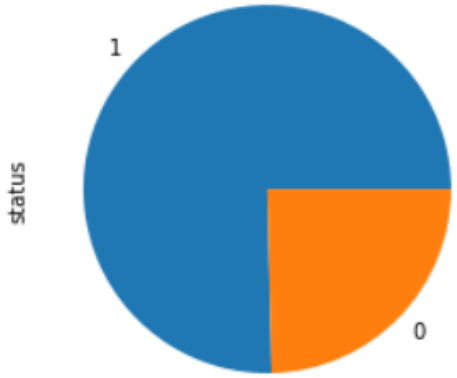
Then we analyzed the column called status, which is the target variable. I found that approximately 25% of the examples were healthy and 75% were for Parkinson's:

```
# Distribution of target Variable. 1 for Parkinson, 0 for No Parkinson
target_count = df['status'].value_counts()
```

```
output :

1    147
0     48
```

```
target_count.plot.bar() [plotting the pie diagram for the target_count]
```



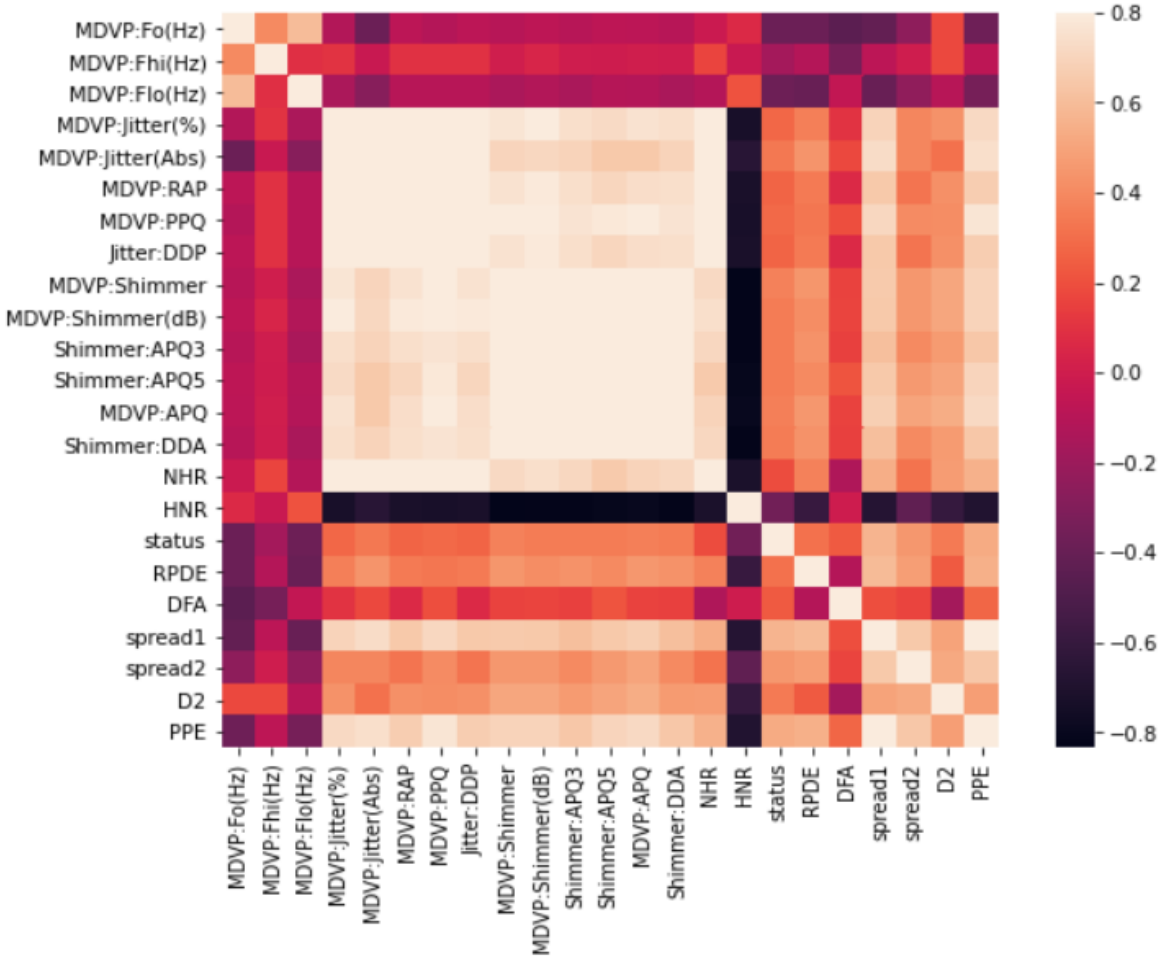

```
#finding the percentage of the column status - target variable
percent = (target_count / len(df))*100
percent
output :

1    75.384615
0    24.615385
```

Heatmap:

A heatmap is created to see how the independent variables correlated with the target.

df.corr() – sn.heatmap() used to create a heatmap for the variables correlated.



5.Split Data for Training and Testing

The X dataframe is then split for training and validating.

The well-known library sklearn.model_selection importing the train_test_split function is used in this process.

Splitting the dataset into training and testing sets keeps 20% of the data for testing and 80% for training .

```
X_train,X_test,Y_train,Y_test = train_test_split(x,y,test_size=0.2,random_state=10
)
```

- Shape of X_train : (156, 22)
- Shape of X_test : (39,22)
- Shape of Y_train : (156,)
- Shape of Y_test : (39,)

The length of the Y_test is 39.

6.Apply different Classification Algorithms and tune them

- **Linear Regression model:**

  Linear regression is perhaps one of the most well-known and well understood algorithms in statistics and machine learning. Generally Linear Regression is used for continuous data but it is used here(classification problem) just for testing purposes. By training the linear regression model with the help of

  ```
  from sklearn.linear_model import LinearRegression
  ```

  Model 1 is taken as Linear Regression model

```
[0.51464502 0.92291805 1.08895196 0.93072429 0.39722818 0.96450257
 0.51665718 0.73428536 0.96425774 0.89648295 0.54933081 1.04009737
 1.02683814 1.02090193 0.52986194 0.21476581 1.07171961 0.93419406
 0.55144217 0.87538823 1.17667136 0.53241189 0.40486539 0.77032915
 0.94973256 0.79196182 0.95604616 1.05056541 0.02141046 0.08635714
 0.86404109 0.73424635 1.07673007 0.16402396 0.26651789 0.70248981
 0.5602687  0.96492844 0.09659312]
```

  Output of Y_predmod 1 is the predicted model with an array of decimal values.

  Our objective is to find the maximum accuracy in training the model. So, we need to convert the above values to binary to find accuracy.
  We use enumerate function,

```
for i,j in enumerate(Y_predmod1):
if(j<0.5):
Y_predmod1[i]=0
else:
Y_predmod1[i]=1
print(Y_predmod1)
print(Y_test)
```

Output for Y_predmod1 and Y_test:

```
[1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 0. 1. 1.
1. 1. 1. 0. 0. 1. 1. 1. 0. 0. 1. 1. 1. 0.]

[1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 0 0 1 0
 1 0]
```

Accuracy on training data in Linear Regression model1 is:

```
model1.score(X_test,Y_test)
```

Output : 0.6634994862742847

Accuracy of the 1st model is 66% which is very low as expected since Linear Regression is best used for continuous values.

- **Logistic Regression model:**

  Logistic Regression was used in the biological sciences in the early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical.

  By training the logistic regression model with the help of

  ```
  from sklearn.linear_model import LogisticRegression
  ```

  Output for Y-predmod2 and Y_test:

```
[1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1 0 0 1 1 1 1 1 0 0 1 1 1 0 0 1 0
 1 0]
[1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 0 0 1 0
 1 0]
```

Accuracy on training data in Logistic Regression model2 is:

```
model2.score(X_test,Y_test)
```

Output : 0.9743589743589743

Accuracy of the 2nd model is 97% which is better than the Linear regression model.

- **Decision Tree model:**
  A decision tree is a machine learning algorithm that partitions the data into subsets. The partitioning process starts with a binary split and continues until no further splits can be made. Various branches of variable length are formed. The goal of a decision tree is to encapsulate the training data in the smallest possible tree.
  Decision trees can handle both categorical and numerical data.

By training the decision tree model with the help of

```
from sklearn import tree
```

Output for Y-predmod3 and Y_test:

```
[1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 1 1 1 0 0 1 0
 1 0]
[1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 0 0 1 0
 1 0]
```

Accuracy on training data in Decision tree model3 is:

```
model3.score(X_test,Y_test)
```

Output : `0.9230769230769231`

Accuracy of the 3rd model is 92% which is better than linear regression model and average when compared to logistic regression.

- **Support Vector Machine model:**

  Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The algorithm classifies the output from the given set of training data into a hyperplane categorizing the data. The C parameter tells the SVM optimization how much you want to avoid misclassifying each training example. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points. Here C value is set as 8 after trial and error.

  By training the support vector model with the help of

  ```
  from sklearn.svm import SVC
  ```

  Output for Y-predmod4 and Y_test:

```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0 1 1
 1 0]
[1 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 0 0 1 0
 1 0]
```

Accuracy on training data in Support vector model4 is:

```
model4.score(X_test,Y_test)
```

```
Output : 1.0
```

Accuracy of the 4th model is 100%

- **Random Forest model:**

  Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML.
  Random forest is one of the accurate learning algorithms. The basic concept of the algorithm is to build many small decision-tree and then merge them to form a forest. It is a computationally easy and cheap process to build many such small and weak decision trees. So, such decision trees can be formed in parallel and then it can be combined to form a single and strong forest.

  By training the random forest model with the help of

  ```
  from sklearn.ensemble import RandomForestClassifier
  ```

  Output for Y-predmod5 and Y_test:

```
[1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 0 0 1 0
 1 0]
[1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 0 0 1 0
 1 0]
```

Accuracy on training data in Random Forest model5 is:

```
model5.score(X_test,Y_test)
Output : 0.9487179487179487
```

Accuracy of the 5th model is 94%

- **XGBoost model:**

  XGBoost is a new Machine Learning algorithm designed with speed and performance in mind. XGBoost stands for eXtreme Gradient Boosting and is based on decision trees.

  By training the XGBoost model with the help of

```
from xgboost import XGBClassifier
```

  Output for Y-predmod6 and Y_test:

```
[1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 0 0 1 0
 1 0]
[1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 0 0 1 0
 1 0]
```

Accuracy on training data in XGBoost model6 is:

```
model6.score(X_test,Y_test)
```

```
Output : 1.0
```

Accuracy of the 6th model is 100%

7.Get performance metrics for all the applied classifiers:

```
from sklearn.metrics import classification_report,confusion_matrix
```

The above function is used to build a text report showing the main classification metrics.

The report shows the main classification metrics **precision, recall** and **f1-score** on a per-class basis. The metrics are calculated by using true and false positives, true and false negatives. Positive and negative in this case are generic names for the predicted classes. There are four ways to check if the predictions are right or wrong:

1. **TN / True Negative:** when a case was negative and predicted negative.
2. **TP / True Positive:** when a case was positive and predicted positive.
3. **FN / False Negative:** when a case was positive but predicted negative.
4. **FP / False Positive:** when a case was negative but predicted positive.

Classification Report of the Performance Metrics

Linear Regression Performance metrics

	precision	recall	f1-score	support
0	1.00	0.73	0.84	11
1	0.90	1.00	0.95	28
accuracy			0.92	39
macro avg	0.95	0.86	0.90	39
weighted avg	0.93	0.92	0.92	39

Logistic Regression Performance metrics

	precision	recall	f1-score	support
0	0.92	1.00	0.96	11
1	1.00	0.96	0.98	28
accuracy			0.97	39
macro avg	0.96	0.98	0.97	39
weighted avg	0.98	0.97	0.97	39

Decision Tree Performance metrics

	precision	recall	f1-score	support
0	0.90	0.82	0.86	11
1	0.93	0.96	0.95	28
accuracy			0.92	39
macro avg	0.92	0.89	0.90	39
weighted avg	0.92	0.92	0.92	39

Support Vector Machines Performance metrics

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	1.00	1.00	1.00	28
accuracy			1.00	39
macro avg	1.00	1.00	1.00	39
weighted avg	1.00	1.00	1.00	39

Random Forest Performance metrics

	precision	recall	f1-score	support
0	0.91	0.91	0.91	11
1	0.96	0.96	0.96	28
accuracy			0.95	39
macro avg	0.94	0.94	0.94	39
weighted avg	0.95	0.95	0.95	39

XGBoost Performance metrics

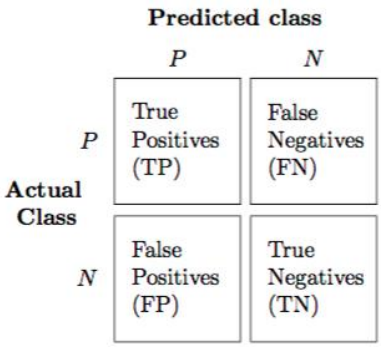
	precision	recall	f1-score	support

```
  0       1.00      1.00      1.00        11
          1       1.00      1.00      1.00        28

    accuracy                            1.00        39
   macro avg       1.00      1.00      1.00        39
weighted avg       1.00      1.00      1.00        39
```

8.Visually compare the performance of all classifiers:

Now for comparing all the performance of the classifiers visually we need a confusion matrix of the tested and predicted value in the form of



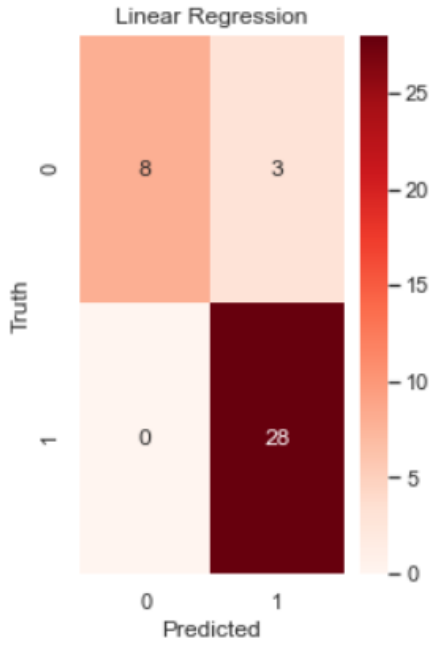
The Confusion matrix is one of the easiest metrics used for finding the correctness and accuracy of the model. It is used for classification problems where the output can be of two or more types of classes. The Confusion matrix is not a performance measure as such, but almost all the performance metrics are based on Confusion Matrix and the numbers inside it.

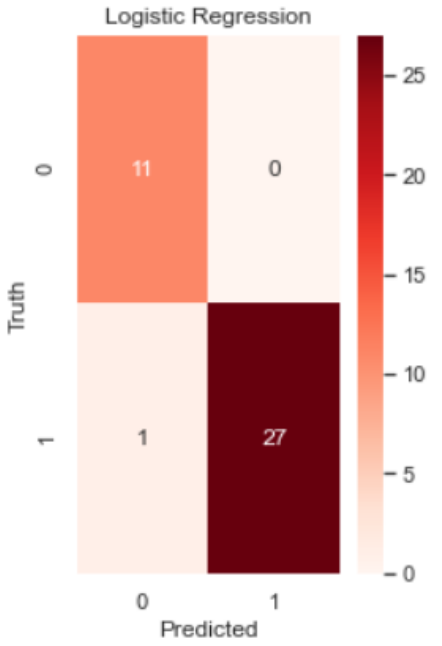
We use heatmaps for classification of algorithms on the acquired data set and then drawing out a comparison of the results to one another and which classifier is most accurate among all the Classifiers.

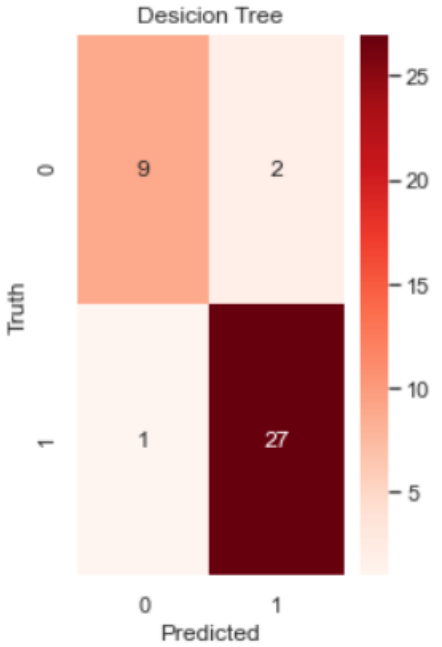
ACCURACY VALUES:

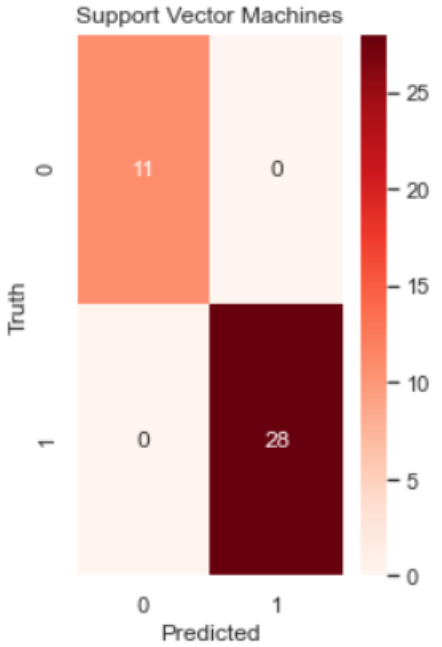
Linear Regression Accuracy:	0.6634994862742847
Logistic Regression Accuracy:	0.9743589743589743
Decision Tree Accuracy:	0.9230769230769231
Support Vector Machines Accuracy:	1.0000000000000000
Random Forest Accuracy:	0.9743589743589743

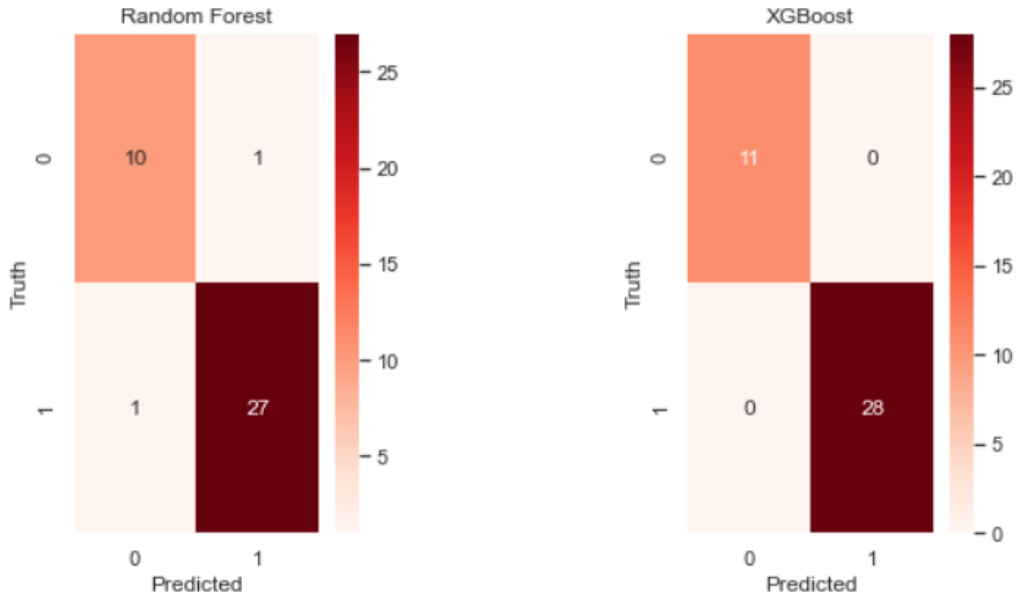
XGBoost:	1.0000000000000000



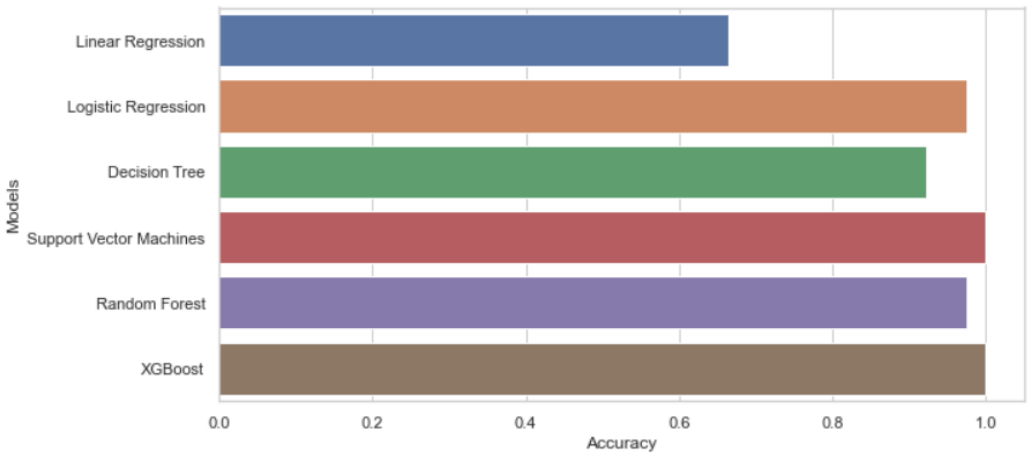








Graphically represents the performance of the six classification techniques based on their accuracy measures.



According to the outputs derived with the help of python, implementing Scikit Libraries,in order to calculate the accuracy and find out the performance, a confusion matrix was constructed at first. From that matrix, the true positives, true negatives along with the false positives and false negatives were used to calculate the support, recall, f1-score, and precision were calculated by implementing specified modules. Final accuracy was calculated using these parameters.

It has been noticed in the result analysis that the opted algorithms under the classification technique show some well-to-do accuracy percentages, especially the Support Vector Machines and XGBoost. Also the models Logistic Regression and Random forest are so close and accurate.

Conclusion and Future Scope:

In this Python machine learning project, we discussed the various possibilities of using machine learning algorithms to detect the presence of Parkinson's Disease in individuals using various factors. Machine Learning classification algorithms were chosen to evaluate their performance in terms of classification performance measures which are accuracy, precision, recall, f1 score and support to classify if the individual is healthy or Parkinson diseased based on the voice input parameters.

These algorithms can be preferred over the others to classify the dependent variable. With the application of six classification algorithms on the acquired data set and then drawing out a comparison of the results to one another and also predicting the outcome whether the person is healthy or Parkinson disease affected from the given data.

The accuracy of the algorithms can be further improved by Feature Selection and Dimensionality Reduction algorithms. The algorithm can be also improved by using Opencv with different imaging and deep learning techniques. UPDRS data can be used with LSTMs to give the progression paths for the disease. Using this as the foundation, we can train different CNNs and LSTMs to show how the deep learning algorithms can be used to analyze Parkinson's data. It is also possible to automate the process of classifying the dataset by creating a simple interface.

References:

- https://matplotlib.org/stable/gallery/index.html#
- https://www.tableau.com/learn/articles/data-visualization
- https://pandas.pydata.org/
- https://numpy.org/
- https://www.javatpoint.com/machine-learning
- https://www.ibm.com/topics/computer-vision
- https://www.python.org/
- https://www.hexnbit.com/
- https://www.tevatrontech.com/
- https://scikit-learn.org/stable/