**Importing The Libraries**

**Project Name:** **statistical machine learning Team ID :** PNT2022TMID04163

**approaches to liver disease prediction**

**ANACONDA:**

Anaconda is the most used distribution platform for python & R programming languages in the data science & machine learning community as it simplifies the installation of packages like [pandas](https://pandas.pydata.org/), [NumPy](https://numpy.org/), [SciPy](https://scipy.org/), and many more. [Conda](https://conda.pydata.org/docs/) is the package manager that the Anaconda distribution is built upon. It is a package manager that is both cross-platform and language agnostic. We can use conda to install any third-party packages.

**PANDAS:**

Pandas is an open-source framework in Python to works with tabular data (rows and columns). pandas have DataFrame which is a two-dimensional data table and Series one dimensional. pandas will help you to explore, clean, and process your data in easy steps

**Program:**

import pandas as pd

df = pd.DataFrame({'X':[78,85,96,80,86], 'Y':[84,94,89,83,86],'Z':[86,97,96,72,83]});

print(df)

**Sample Output:**

X Y Z

0 78 84 86

1 85 94 97

2 96 89 96

3 80 83 72

4 86 86 83

**NUMPY:**

**Before you can import numpy, you first need to install it. There are two ways to install numpy:**

* Install the binary (pre-compiled) version using pip
* Compile it from source code, and then install it

**Program:**

# Python program to demonstrate

# basic array characteristics

import numpy as np

# Creating array object

arr = np.array( [[ 1, 2, 3],

[ 4, 2, 5]] )

# Printing type of arr object

print("Array is of type: ", type(arr))

# Printing array dimensions (axes)

print("No. of dimensions: ", arr.ndim)

# Printing shape of array

print("Shape of array: ", arr.shape)

# Printing size (total number of elements) of array

print("Size of array: ", arr.size)

# Printing type of elements in array

print("Array stores elements of type: ", arr.dtype)

**Output :**

Array is of type:

No. of dimensions: 2

Shape of array: (2, 3)

Size of array: 6

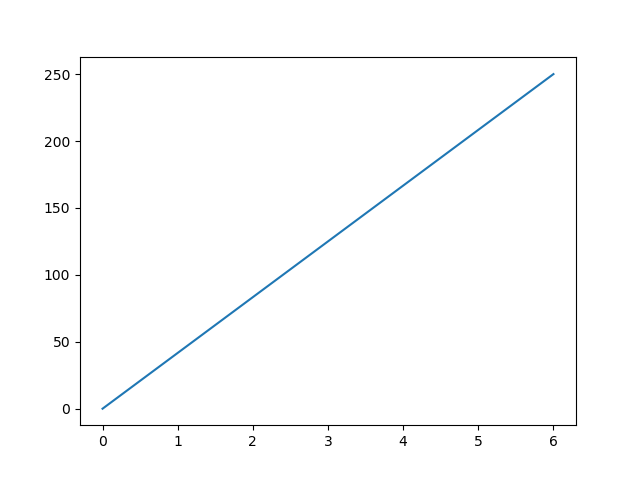
Array stores elements of type: int6

**Matplotlib :**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

import matplotlib.pyplot as plt  
import numpy as np  
  
xpoints = np.array([0, 6])  
ypoints = np.array([0, 250])  
  
plt.plot(xpoints, ypoints)  
plt.show()

Result:



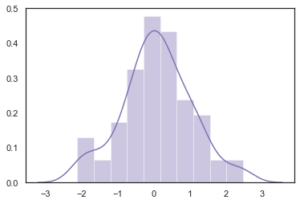
**Seaborn:**

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of [matplotlib](https://www.geeksforgeeks.org/python-introduction-matplotlib/) library and also closely integrated to the data structures from [pandas](https://www.geeksforgeeks.org/introduction-to-pandas-in-python/).

**program:**

|  |
| --- |
| # Importing libraries  import numpy as np  import seaborn as sns      # Selecting style as white,  # dark, whitegrid, darkgrid  # or ticks  sns.set(style="white")    # Generate a random univariate  # dataset  rs = np.random.RandomState(10)  d = rs.normal(size=100)    # Plot a simple histogram and kde  # with binsize determined automatically  sns.distplot(d, kde=True, color="m") |

**Output:**



**Pickle:**

“Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream (from a [binary file](https://docs.python.org/3/glossary.html#term-binary-file) or [bytes-like object](https://docs.python.org/3/glossary.html#term-bytes-like-object)) is converted back into an object hierarchy.

**program:**

|  |
| --- |
| # Python3 program to illustrate store  # efficiently using pickle module  # Module translates an in-memory Python object  # into a serialized byte stream—a string of  # bytes that can be written to any file-like object.    import pickle    def storeData():      # initializing data to be stored in db      Omkar = {'key' : 'Omkar', 'name' : 'Omkar Pathak',      'age' : 21, 'pay' : 40000}      Jagdish = {'key' : 'Jagdish', 'name' : 'Jagdish Pathak',      'age' : 50, 'pay' : 50000}        # database      db = {}      db['Omkar'] = Omkar      db['Jagdish'] = Jagdish        # Its important to use binary mode      dbfile = open('examplePickle', 'ab')        # source, destination      pickle.dump(db, dbfile)      dbfile.close()    def loadData():      # for reading also binary mode is important      dbfile = open('examplePickle', 'rb')      db = pickle.load(dbfile)      for keys in db:          print(keys, '=>', db[keys])      dbfile.close()    if \_\_name\_\_ == '\_\_main\_\_':      storeData()      loadData() |

**Output:**

omkarpathak-Inspiron-3542:~/Documents/Python-Programs$ python P60\_PickleModule.py

Omkar => {'age': 21,  'name': 'Omkar Pathak',  'key': 'Omkar',  'pay': 40000}

Jagdish => {'age': 50,  'name': 'Jagdish Pathak',  'key': 'Jagdish',  'pay': 50000}