Name : Gulam Jawwad Khan

Roll No : 43

Section : A

Date : 05/07/2024

Experiment No : 1

Aim : Study of Anaconda IDE and it's installation

What is Anaconda?

Anaconda is a platform that includes:

- **Python and R distributions:** The core programming languages for data science and machine learning.
- **Pre-installed packages:** A vast number of libraries for data science, machine learning, AI, and data analysis (like NumPy, Pandas, Matplotlib, SciPy, scikit-learn, TensorFlow, etc.).
- **Package management:** With conda, a package manager for handling dependencies and environments.
- **IDE Support:** It integrates with multiple IDEs like Jupyter Notebook, JupyterLab, Spyder, and Visual Studio Code.

Popular Applications to Use in Anaconda

Anaconda includes several applications, with some of the most commonly used ones being:

1. **Jupyter Notebook:** An interactive web-based notebook that allows you to write and run code in real-time.
2. **JupyterLab:** An extension of Jupyter Notebook with additional features like file browsers, terminals, and support for multiple panes.
3. **Spyder:** An open-source scientific IDE specifically designed for data science, with features like code editing, debugging, and interactive execution.
4. **Visual Studio Code (VS Code):** An extensible code editor that supports debugging, task running, and version control, often used for data science and software development.

How to Use Anaconda Navigator

Anaconda Navigator is a graphical user interface (GUI) that makes it easy to manage applications, environments, and packages.

Here's how you can use it:

1. **Launch Anaconda Navigator:**
    - Open the Anaconda Navigator from your start menu (Windows) or applications folder (macOS/Linux).
2. **Home Screen Overview:**
    - You'll see a list of all the available applications like Jupyter Notebook, JupyterLab, Spyder, and more.
    - Each application has a "Launch" button to start the application.
3. **Creating and Managing Environments:**
    - Click on the "Environments" tab to create or manage virtual environments.
    - Virtual environments help isolate different projects with different dependencies.
    - To create a new environment, click the "Create" button, name the environment, and choose the Python version you need.
4. **Installing Packages:**
    - Within the "Environments" tab, select your desired environment.
    - Use the search bar to find packages and click "Apply" to install them.
    - You can also use the conda command in the terminal to manage packages (e.g., conda install numpy).
5. **Launching Applications:**
    - Once your environment is set up, return to the "Home" tab.
    - Click "Launch" next to the application you wish to use (e.g., Jupyter Notebook or Spyder).
    - The application will open, ready for you to start coding.
6. **Updating Anaconda and Packages:**
    - To update Anaconda Navigator itself, click on the "Update Index" button.
    - For individual packages, go to the "Environments" tab, select the package, and click "Update."

Using Anaconda from the Command Line

Although Anaconda Navigator provides a GUI, you can also use the command line for more control:

- conda create -n myenv python=3.9 — Creates a new environment with Python 3.9.

- conda activate myenv — Activates the environment named myenv.
- conda install package_name — Installs a package in the current environment.
- conda list — Lists all packages in the active environment.

Benefits of Using Anaconda

- Easy installation of packages and tools: Anaconda simplifies setting up your Python environment with all the libraries you need for data science.
- Environment management: With conda, you can create multiple isolated environments, avoiding dependency conflicts.
- Access to data science tools: It provides easy access to tools like Jupyter Notebook and Spyder, essential for interactive data analysis.

Anaconda is especially popular among data scientists, researchers, and anyone involved in machine learning or AI projects because of its comprehensive toolkit and ease of use.

Step-by-step guide to installing Anaconda Navigator:

Step 1: Download Anaconda

1. Visit the Anaconda website:
    - Go to the official Anaconda distribution page: https://www.anaconda.com/products/distribution.
2. Choose the installer:
    - Click the "Download" button for your operating system (Windows, macOS, or Linux).
    - Choose the Python version you prefer (Python 3.8 or later is recommended).
    - Download the installer file for your platform.

Step 2: Run the Installer

1. Locate the downloaded file:o Find the Anaconda installer file on your computer (e.g., Anaconda3-2023.x-x-Windows x86_64.exe for Windows).
2. Start the installation:
    - Double-click the installer to start the installation process.

- You might be asked for administrator permissions; click "Yes" if prompted.

Step 3: Follow Installation Instructions

1. Welcome Screen:
   - Click "Next" on the welcome screen.
2. License Agreement:
   - Read and accept the license agreement, then click "Next."
3. Choose Installation Type:
   - Select "Just Me" (recommended) if you want to install it for the current user. o Click "Next."
4. Select Installation Location:
   - Choose the directory where you want to install Anaconda (default location is usually fine).
   - Click "Next."
5. Advanced Installation Options:
   - You will be given two options:
     - **Add Anaconda to my PATH environment variable:** It is generally not recommended to check this box to avoid conflicts with other Python installations.
     - **Register Anaconda as my default Python 3.8 (or later) environment:** It's recommended to check this box to set Anaconda as the default Python.
   - Click "Install" to start the installation.

Step 4: Complete the Installation

1. Wait for Installation to Complete:
   - The installation process might take a few minutes. Let it finish.
2. Finish Installation:
   - Once the installation is complete, you will see the "Completing" screen.
   - Click "Next," then "Finish" to close the installer.

Step 5: Launch Anaconda Navigator

1. Open Anaconda Navigator:
   - Windows: Open the Start menu, search for "Anaconda Navigator," and click to launch it.

- macOS/Linux: Use the Applications folder or the launcher to find and open Anaconda Navigator.

2. Initial Setup:
   - Anaconda Navigator may take a moment to open the first time. It will display a dashboard with various applications like Jupyter Notebook, JupyterLab, Spyder, and others.

Step 6: Verify the Installation

1. Check the Installed Version:
   - Open a terminal or command prompt.
   - Type conda --version and press Enter.
   - If the installation was successful, it should display the version of Conda you installed.

Step 7: Update Anaconda (Optional but Recommended)

1. Update Anaconda Navigator:
   - Open Anaconda Navigator.
   - Click on the "Update Index" button (if visible) to update packages and Navigator.
   - You can also update via the terminal by typing conda update anaconda-navigator.

Data Aquisition

In [ ]:
```
#Aim:- To Perform operaion on 'DATA AQUISITION'
```

In [1]:
```
#Name :- Gulam Jawwad Khan
#Roll No.:- 43
#Sec :- A
#Subject :- DSS(P.E.1)
#Date :- 27/07/2024
```

In [2]:
```
import pandas as pd
```

In [3]:
```
import os
```

In [4]:
```
os.getcwd()
```

Out[4]:
```
'C:\\Users\\Lenovo'
```

In [5]:
```
os.chdir("C:\\Users\\Lenovo\\OneDrive\\Doc\\Desktop")
```

In [6]:
```
df = pd.read_csv("diabetes.csv")
```

In [7]:
```
df.head()
```

Out[7]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [8]:
```
df.head(100)
```

Out[8]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
95	6	144	72	27	228	33.9	0.255	40	0
96	2	92	62	28	0	31.6	0.130	24	0
97	1	71	48	18	76	20.4	0.323	22	0
98	6	93	50	30	64	28.7	0.356	23	0
99	1	122	90	51	220	49.7	0.325	31	1

100 rows × 9 columns

In [9]:
```
df.tail()
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

In [ ]:
```

```

Data Specialization

In [ ]:
```
#Aim :- To perform operation on 'DATA SPECIALIZATION'
```

In [2]:
```
#Name :- Gulam Jawwad Khan
#Roll No.:- 43
#Sec :- A
#Subject :- DSS(P.E.1)
#Date :- 27/07/2024
```

In [3]:
```
import pandas as pd
```

In [4]:
```
import os
```

In [5]:
```
os.getcwd()
```

Out[5]:
```
'C:\\Users\\Lenovo'
```

In [8]:
```
os.chdir("C:\\Users\\Lenovo\\OneDrive\\Doc\\Desktop")
```

In [10]:
```
df = pd.read_csv("framingham.csv")
```

In [11]:
```
df.head()
```

Out[11]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	80.0	77.0	0
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	95.0	76.0	0
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	75.0	70.0	0
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	65.0	103.0	1
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	85.0	85.0	0

In [13]:
```
df.head(100)
```

Out[13]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	80.0	77.0	0
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	95.0	76.0	0
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	75.0	70.0	0
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	65.0	103.0	1
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	85.0	85.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	0	65	3.0	0	0.0	0.0	0	0	0	193.0	123.0	76.5	29.33	60.0	96.0	0
96	0	63	4.0	1	20.0	0.0	0	0	1	239.0	134.0	80.0	26.64	88.0	126.0	0
97	0	40	2.0	0	0.0	0.0	0	0	0	205.0	100.0	60.0	NaN	60.0	72.0	1
98	0	56	1.0	0	0.0	0.0	0	1	0	296.0	180.0	90.0	23.72	75.0	120.0	0
99	0	56	1.0	1	15.0	0.0	0	0	0	269.0	121.0	75.0	22.36	50.0	66.0	0

100 rows × 16 columns

In [14]:
```
df.tail()
```

Out[14]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
4233	1	50	1.0	1	1.0	0.0	0	1	0	313.0	179.0	92.0	25.97	66.0	86.0	1
4234	1	51	3.0	1	43.0	0.0	0	0	0	207.0	126.5	80.0	19.71	65.0	68.0	0
4235	0	48	2.0	1	20.0	NaN	0	0	0	248.0	131.0	72.0	22.00	84.0	86.0	0
4236	0	44	1.0	1	15.0	0.0	0	0	0	210.0	126.5	87.0	19.16	86.0	NaN	0
4237	0	52	2.0	0	0.0	0.0	0	0	0	269.0	133.5	83.0	21.47	80.0	107.0	0

In [15]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             4238 non-null   int64  
 1   age              4238 non-null   int64  
 2   education        4133 non-null   float64
 3   currentSmoker    4238 non-null   int64  
 4   cigsPerDay       4209 non-null   float64
 5   BPMeds           4185 non-null   float64
 6   prevalentStroke  4238 non-null   int64  
 7   prevalentHyp     4238 non-null   int64  
 8   diabetes         4238 non-null   int64  
 9   totChol          4188 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4219 non-null   float64
 13  heartRate        4237 non-null   float64
 14  glucose          3850 non-null   float64
 15  TenYearCHD       4238 non-null   int64  
dtypes: float64(9), int64(7)
memory usage: 529.9 KB
```

In [16]:
```
df.shape
```

Out[16]:
```
(4238, 16)
```

In [17]:
```
df.size
```

Out[17]:
```
67808
```

In [18]:
```
df.ndim
```

Out[18]:
```
2
```

In [19]:
```
df.tail(10)
```

Out[19]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
4228	0	50	1.0	0	0.0	0.0	0	1	1	260.0	190.0	130.0	43.67	85.0	260.0	0
4229	0	51	3.0	1	20.0	0.0	0	1	0	251.0	140.0	80.0	25.60	75.0	NaN	0
4230	0	56	1.0	1	3.0	0.0	0	1	0	268.0	170.0	102.0	22.89	57.0	NaN	0
4231	1	58	3.0	0	0.0	0.0	0	1	0	187.0	141.0	81.0	24.96	80.0	81.0	0
4232	1	68	1.0	0	0.0	0.0	0	1	0	176.0	168.0	97.0	23.14	60.0	79.0	1
4233	1	50	1.0	1	1.0	0.0	0	1	0	313.0	179.0	92.0	25.97	66.0	86.0	1
4234	1	51	3.0	1	43.0	0.0	0	0	0	207.0	126.5	80.0	19.71	65.0	68.0	0
4235	0	48	2.0	1	20.0	NaN	0	0	0	248.0	131.0	72.0	22.00	84.0	86.0	0
4236	0	44	1.0	1	15.0	0.0	0	0	0	210.0	126.5	87.0	19.16	86.0	NaN	0
4237	0	52	2.0	0	0.0	0.0	0	0	0	269.0	133.5	83.0	21.47	80.0	107.0	0

In [20]:
```
df.describe()
```

Out[20]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
count	4238.000000	4238.000000	4133.000000	4238.000000	4209.000000	4185.000000	4238.000000	4238.000000	4238.000000	4188.000000	4238.000000	4238.000000	4219.000000	4237.000000	3850.000000	4238.000000
mean	0.429212	49.584946	1.978950	0.494101	9.003089	0.029630	0.005899	0.310524	0.025720	236.721585	132.352407	82.893464	25.802008	75.878924	81.966753	0.151958
std	0.495022	8.572160	1.019791	0.500024	11.920094	0.169584	0.076587	0.462763	0.158316	44.590334	22.038097	11.910850	4.080111	12.026596	23.959998	0.359023
min	0.000000	32.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	107.000000	83.500000	48.000000	15.540000	44.000000	40.000000	0.000000
25%	0.000000	42.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	206.000000	117.000000	75.000000	23.070000	68.000000	71.000000	0.000000
50%	0.000000	49.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	234.000000	128.000000	82.000000	25.400000	75.000000	78.000000	0.000000
75%	1.000000	56.000000	3.000000	1.000000	20.000000	0.000000	0.000000	1.000000	0.000000	263.000000	144.000000	89.875000	28.040000	83.000000	87.000000	0.000000
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	1.000000	1.000000	1.000000	696.000000	295.000000	142.500000	56.800000	143.000000	394.000000	1.000000

In [ ]:
```

```

Data Manipulation

In [ ]:
```
#Aim:- To Perform operaion on 'DATA MANIPULATION'
```

In [1]:
```
#Name :- Gulam Jawwad Khan
#Roll No.:- 43
#Sec :- A
#Subject :- DSS(P.E.1)
#Date :- 24/08/2024
```

In [2]:
```
import pandas as pd
```

In [3]:
```
import os
```

In [4]:
```
os.getcwd()
```

Out[4]:
```
'C:\\Users\\Lenovo'
```

In [5]:
```
os.chdir("C:\\Users\\Lenovo\\OneDrive\\Doc\\Desktop")
```

In [6]:
```
df = pd.read_csv("titanic.csv")
```

In [7]:
```
df
```

Out[7]:

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
0	1.0	1.0	Allen, Miss. Elisabeth Walton	female	29.0000	0.0	0.0	24160	211.3375	B5	S	2	NaN	St Louis, MO
1	1.0	1.0	Allison, Master. Hudson Trevor	male	0.9167	1.0	2.0	113781	151.5500	C22 C26	S	11	NaN	Montreal, PQ / Chesterville, ON
2	1.0	0.0	Allison, Miss. Helen Loraine	female	2.0000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON
3	1.0	0.0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	135.0	Montreal, PQ / Chesterville, ON
4	1.0	0.0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1305	3.0	0.0	Zabour, Miss. Thamine	female	NaN	1.0	0.0	2665	14.4542	NaN	C	NaN	NaN	NaN
1306	3.0	0.0	Zakarian, Mr. Mapriededer	male	26.5000	0.0	0.0	2656	7.2250	NaN	C	NaN	304.0	NaN
1307	3.0	0.0	Zakarian, Mr. Ortin	male	27.0000	0.0	0.0	2670	7.2250	NaN	C	NaN	NaN	NaN
1308	3.0	0.0	Zimmerman, Mr. Leo	male	29.0000	0.0	0.0	315082	7.8750	NaN	S	NaN	NaN	NaN
1309	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1310 rows × 14 columns

In [8]:
```
df.head()
```

Out[8]:

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
0	1.0	1.0	Allen, Miss. Elisabeth Walton	female	29.0000	0.0	0.0	24160	211.3375	B5	S	2	NaN	St Louis, MO
1	1.0	1.0	Allison, Master. Hudson Trevor	male	0.9167	1.0	2.0	113781	151.5500	C22 C26	S	11	NaN	Montreal, PQ / Chesterville, ON
2	1.0	0.0	Allison, Miss. Helen Loraine	female	2.0000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON
3	1.0	0.0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	135.0	Montreal, PQ / Chesterville, ON
4	1.0	0.0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON

In [9]:
```
df.tail()
```

Out[9]:

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
1305	3.0	0.0	Zabour, Miss. Thamine	female	NaN	1.0	0.0	2665	14.4542	NaN	C	NaN	NaN	NaN
1306	3.0	0.0	Zakarian, Mr. Mapriededer	male	26.5	0.0	0.0	2656	7.2250	NaN	C	NaN	304.0	NaN
1307	3.0	0.0	Zakarian, Mr. Ortin	male	27.0	0.0	0.0	2670	7.2250	NaN	C	NaN	NaN	NaN
1308	3.0	0.0	Zimmerman, Mr. Leo	male	29.0	0.0	0.0	315082	7.8750	NaN	S	NaN	NaN	NaN
1309	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1310 entries, 0 to 1309
Data columns (total 14 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   pclass      1309 non-null   float64
 1   survived    1309 non-null   float64
 2   name        1309 non-null   object 
 3   sex         1309 non-null   object 
 4   age         1046 non-null   float64
 5   sibsp       1309 non-null   float64
 6   parch       1309 non-null   float64
 7   ticket      1309 non-null   object 
 8   fare        1308 non-null   float64
 9   cabin       295 non-null    object 
 10  embarked    1307 non-null   object 
 11  boat        486 non-null    object 
 12  body        121 non-null    float64
 13  home.dest   745 non-null    object 
dtypes: float64(7), object(7)
memory usage: 143.4+ KB
```

In [11]:
```
df.describe()
```

Out[11]:

	pclass	survived	age	sibsp	parch	fare	body
count	1309.000000	1309.000000	1046.000000	1309.000000	1309.000000	1308.000000	121.000000
mean	2.294882	0.381971	29.881135	0.498854	0.385027	33.295479	160.809917
std	0.837836	0.486055	14.413500	1.041658	0.865560	51.758668	97.696922
min	1.000000	0.000000	0.166700	0.000000	0.000000	0.000000	1.000000
25%	2.000000	0.000000	21.000000	0.000000	0.000000	7.895800	72.000000
50%	3.000000	0.000000	28.000000	0.000000	0.000000	14.454200	155.000000
75%	3.000000	1.000000	39.000000	1.000000	0.000000	31.275000	256.000000
max	3.000000	1.000000	80.000000	8.000000	9.000000	512.329200	328.000000

In [13]:
```
df.shape
```

Out[13]:
```
(1310, 14)
```

In [14]:
```
df.size
```

Out[14]:
```
18340
```

In [15]:
```
df.ndim
```

Out[15]:
```
2
```

In [16]:
```
df.columns
```

Out[16]:
```
Index(['pclass', 'survived', 'name', 'sex', 'age', 'sibsp', 'parch', 'ticket',
       'fare', 'cabin', 'embarked', 'boat', 'body', 'home.dest'],
      dtype='object')
```

In [17]:
```
df.drop(labels='age', axis=1)
```

Out[17]:

	pclass	survived	name	sex	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
0	1.0	1.0	Allen, Miss. Elisabeth Walton	female	0.0	0.0	24160	211.3375	B5	S	2	NaN	St Louis, MO
1	1.0	1.0	Allison, Master. Hudson Trevor	male	1.0	2.0	113781	151.5500	C22 C26	S	11	NaN	Montreal, PQ / Chesterville, ON
2	1.0	0.0	Allison, Miss. Helen Loraine	female	1.0	2.0	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON
3	1.0	0.0	Allison, Mr. Hudson Joshua Creighton	male	1.0	2.0	113781	151.5500	C22 C26	S	NaN	135.0	Montreal, PQ / Chesterville, ON
4	1.0	0.0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	1.0	2.0	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1305	3.0	0.0	Zabour, Miss. Thamine	female	1.0	0.0	2665	14.4542	NaN	C	NaN	NaN	NaN
1306	3.0	0.0	Zakarian, Mr. Mapriededer	male	0.0	0.0	2656	7.2250	NaN	C	NaN	304.0	NaN
1307	3.0	0.0	Zakarian, Mr. Ortin	male	0.0	0.0	2670	7.2250	NaN	C	NaN	NaN	NaN
1308	3.0	0.0	Zimmerman, Mr. Leo	male	0.0	0.0	315082	7.8750	NaN	S	NaN	NaN	NaN
1309	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1310 rows × 13 columns

In [18]:
```
df.drop(labels=2, axis=0)
```

Out[18]:

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
0	1.0	1.0	Allen, Miss. Elisabeth Walton	female	29.0000	0.0	0.0	24160	211.3375	B5	S	2	NaN	St Louis, MO
1	1.0	1.0	Allison, Master. Hudson Trevor	male	0.9167	1.0	2.0	113781	151.5500	C22 C26	S	11	NaN	Montreal, PQ / Chesterville, ON
3	1.0	0.0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	135.0	Montreal, PQ / Chesterville, ON
4	1.0	0.0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON
5	1.0	1.0	Anderson, Mr. Harry	male	48.0000	0.0	0.0	19952	26.5500	E12	S	3	NaN	New York, NY
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1305	3.0	0.0	Zabour, Miss. Thamine	female	NaN	1.0	0.0	2665	14.4542	NaN	C	NaN	NaN	NaN
1306	3.0	0.0	Zakarian, Mr. Mapriededer	male	26.5000	0.0	0.0	2656	7.2250	NaN	C	NaN	304.0	NaN
1307	3.0	0.0	Zakarian, Mr. Ortin	male	27.0000	0.0	0.0	2670	7.2250	NaN	C	NaN	NaN	NaN
1308	3.0	0.0	Zimmerman, Mr. Leo	male	29.0000	0.0	0.0	315082	7.8750	NaN	S	NaN	NaN	NaN
1309	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1309 rows × 14 columns

In [ ]:
```

```

In [1]:
```
#Aim:- Creation of 1D, 2D and multidimensional array (Data cube/OLAP) using numpy.
```

In [2]:
```
#Name :- Gulam Jawwad Khan
#Roll No.:- 43
#Sec :- A
#Subject :- DSS(P.E.1)
#Date :- 24/08/2024
```

In [3]:
```
import numpy as np
```

In [4]:
```
a1 = np.array([10, 20, 30, 40, 50])
```

In [5]:
```
a1
```

Out[5]:
```
array([10, 20, 30, 40, 50])
```

In [8]:
```
a2 = np.array([[10, 20, 30, 40, 50], [60, 70, 80, 90]], dtype=object)
```

In [9]:
```
a2
```

Out[9]:
```
array([list([10, 20, 30, 40, 50]), list([60, 70, 80, 90])], dtype=object)
```

In [10]:
```
a3 = np.array([['R1','R2', 'R3', 'R4'], ['ABC', 'XYZ', 'PQR', 'EFG'], [40, 55, 64, 22]])
```

In [11]:
```
a3
```

Out[11]:
```
array([['R1', 'R2', 'R3', 'R4'],
       ['ABC', 'XYZ', 'PQR', 'EFG'],
       ['40', '55', '64', '22']], dtype='<U11')
```

In [ ]:

Data Cleaning Missing Value Treatment

In [2]:
```
#Aim :- Data preprocessing_Data Cleaning_Missing Value Treatment
```

In [3]:
```
#Name :- Gulam Jawwad Khan
#Roll No.:- 43
#Sec :- A
#Subject :- DSS(P.E.1)
#Date :- 03/08/2024
```

In [4]:
```
import pandas as pd
```

In [5]:
```
import os
```

In [6]:
```
os.getcwd()
```

Out[6]:
```
'C:\\Users\\HP'
```

In [7]:
```
os.chdir("c:\\Users\\HP\\Desktop")
```

In [8]:
```
data = pd.read_csv("Titanic.csv")
```

In [9]:
```
data.head()
```

Out[9]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [10]:
```
data.tail()
```

Out[10]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

In [11]:
```
data.head(25)
```

Out[11]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q

6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S
12	13	0	3	Saundercock, Mr. William Henry	male	20.0	0	0	A/5. 2151	8.0500	NaN	S
13	14	0	3	Andersson, Mr. Anders Johan	male	39.0	1	5	347082	31.2750	NaN	S
14	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0	0	0	350406	7.8542	NaN	S
15	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55.0	0	0	248706	16.0000	NaN	S
16	17	0	3	Rice, Master. Eugene	male	2.0	4	1	382652	29.1250	NaN	Q
17	18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	13.0000	NaN	S
18	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female	31.0	1	0	345763	18.0000	NaN	S
19	20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	7.2250	NaN	C
20	21	0	2	Fynney, Mr. Joseph J	male	35.0	0	0	239865	26.0000	NaN	S
21	22	1	2	Beesley, Mr. Lawrence	male	34.0	0	0	248698	13.0000	D56	S
22	23	1	3	McGowan, Miss. Anna "Annie"	female	15.0	0	0	330923	8.0292	NaN	Q
23	24	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788	35.5000	A6	S
24	25	0	3	Palsson, Miss. Torborg Danira	female	8.0	3	1	349909	21.0750	NaN	S

In [12]:
```
data.tail(43)
```

Out[12]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
848	849	0	2	Harper, Rev. John	male	28.0	0	1	248727	33.0000	NaN	S
849	850	1	1	Goldenberg, Mrs. Samuel L (Edwiga Grabowska)	female	NaN	1	0	17453	89.1042	C92	C
850	851	0	3	Andersson, Master. Sigvard Harald Elias	male	4.0	4	2	347082	31.2750	NaN	S
851	852	0	3	Svensson, Mr. Johan	male	74.0	0	0	347060	7.7750	NaN	S
852	853	0	3	Boulos, Miss. Nourelain	female	9.0	1	1	2678	15.2458	NaN	C
853	854	1	1	Lines, Miss. Mary Conover	female	16.0	0	1	PC 17592	39.4000	D28	S
854	855	0	2	Carter, Mrs. Ernest Courtenay (Lilian Hughes)	female	44.0	1	0	244252	26.0000	NaN	S
855	856	1	3	Aks, Mrs. Sam (Leah Rosen)	female	18.0	0	1	392091	9.3500	NaN	S
856	857	1	1	Wick, Mrs. George Dennick (Mary Hitchcock)	female	45.0	1	1	36928	164.8667	NaN	S
857	858	1	1	Daly, Mr. Peter Denis	male	51.0	0	0	113055	26.5500	E17	S
858	859	1	3	Baclini, Mrs. Solomon (Latifa Qurban)	female	24.0	0	3	2666	19.2583	NaN	C
859	860	0	3	Razi, Mr. Raihed	male	NaN	0	0	2629	7.2292	NaN	C
860	861	0	3	Hansen, Mr. Claus Peter	male	41.0	2	0	350026	14.1083	NaN	S
861	862	0	2	Giles, Mr. Frederick Edward	male	21.0	1	0	28134	11.5000	NaN	S
862	863	1	1	Swift, Mrs. Frederick Joel (Margaret Welles Ba...	female	48.0	0	0	17466	25.9292	D17	S
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.5500	NaN	S
864	865	0	2	Gill, Mr. John William	male	24.0	0	0	233866	13.0000	NaN	S
865	866	1	2	Bystrom, Mrs. (Karolina)	female	42.0	0	0	236852	13.0000	NaN	S
866	867	1	2	Duran y More, Miss. Asuncion	female	27.0	1	0	SC/PARIS 2149	13.8583	NaN	C
867	868	0	1	Roebling, Mr. Washington Augustus II	male	31.0	0	0	PC 17590	50.4958	A24	S
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777	9.5000	NaN	S
869	870	1	3	Johnson, Master. Harold Theodor	male	4.0	1	1	347742	11.1333	NaN	S
870	871	0	3	Balkic, Mr. Cerin	male	26.0	0	0	349248	7.8958	NaN	S
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542	D35	S

872	873	0	1	Carlsson, Mr. Frans Olof	male	33.0	0	0	695	5.0000	B51 B53 B55	S
873	874	0	3	Vander Cruyssen, Mr. Victor	male	47.0	0	0	345765	9.0000	NaN	S
874	875	1	2	Abelson, Mrs. Samuel (Hannah Wizosky)	female	28.0	1	0	P/PP 3381	24.0000	NaN	C
875	876	1	3	Najib, Miss. Adele Kiamie "Jane"	female	15.0	0	0	2667	7.2250	NaN	C
876	877	0	3	Gustafsson, Mr. Alfred Ossian	male	20.0	0	0	7534	9.8458	NaN	S
877	878	0	3	Petroff, Mr. Nedelio	male	19.0	0	0	349212	7.8958	NaN	S
878	879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217	7.8958	NaN	S
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C
880	881	1	2	Shelley, Mrs. William (Imanita Parrish Hall)	female	25.0	0	1	230433	26.0000	NaN	S
881	882	0	3	Markun, Mr. Johann	male	33.0	0	0	349257	7.8958	NaN	S
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22.0	0	0	7552	10.5167	NaN	S
883	884	0	2	Banfield, Mr. Frederick James	male	28.0	0	0	C.A./SOTON 34068	10.5000	NaN	S
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.0500	NaN	S
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1250	NaN	Q
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

In [13]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [14]:
```
data.describe()
```

Out[14]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [15]:

In [15]:
```
data.ndim
```

Out[15]:
```
2
```

In [16]:
```
data.shape
```

Out[16]:
```
(891, 12)
```

In [17]:
```
data.size
```

Out[17]:
```
10692
```

In [18]:
```
data.isna()
```

Out[18]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	True	False
...	...	...	...	...	...	...	...	...	...	...	...	...
886	False	False	False	False	False	False	False	False	False	False	True	False
887	False	False	False	False	False	False	False	False	False	False	False	False
888	False	False	False	False	False	True	False	False	False	False	True	False
889	False	False	False	False	False	False	False	False	False	False	False	False
890	False	False	False	False	False	False	False	False	False	False	True	False

891 rows × 12 columns

In [19]:
```
data.isna().any()
```

Out[19]:
```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age             True
SibSp          False
Parch          False
Ticket         False
Fare           False
Cabin           True
Embarked        True
dtype: bool
```

In [20]:
```
data.isna().sum()
```

Out[20]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [21]:
```
data["Age"].fillna(29.699118)
```

```
data['Age'].fillna(29.699118)
```

Out[21]:
```
0      22.000000
1      38.000000
2      26.000000
3      35.000000
4      35.000000
         ...
886    27.000000
887    19.000000
888    29.699118
889    26.000000
890    32.000000
Name: Age, Length: 891, dtype: float64
```

In [22]:
```
data.isna().sum()
```

Out[22]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [23]:
```
data["Age"].fillna(29.699118)
```

Out[23]:
```
0      22.000000
1      38.000000
2      26.000000
3      35.000000
4      35.000000
         ...
886    27.000000
887    19.000000
888    29.699118
889    26.000000
890    32.000000
Name: Age, Length: 891, dtype: float64
```

In [24]:
```
data.isna().sum()
```

Out[24]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [27]:
```
data = data.dropna()
```

In [28]:
```
data.isna().any()
```

Out[28]:
```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age            False
SibSp          False
Parch          False
```

```
Ticket         False
Fare           False
Cabin          False
Embarked       False
dtype: bool
```

In [31]:
```
data.isna().sum()
```

Out[31]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

In [ ]:
```

```

Data Visualization

In [1]:
```
#Aim:- To Perform 'DATA VISUALIZAION'
```

In [2]:
```
#Name :- Gulam Jawwad Khan
#Roll No.:- 43
#Sec :- A
#Subject :- DSS(P.E.1)
#Date :- 27/07/2024
```

In [4]:
```
import numpy as np
from matplotlib import pyplot as plt
```

In [5]:
```
x = np.arange(1, 11)
```

In [6]:
```
x
```

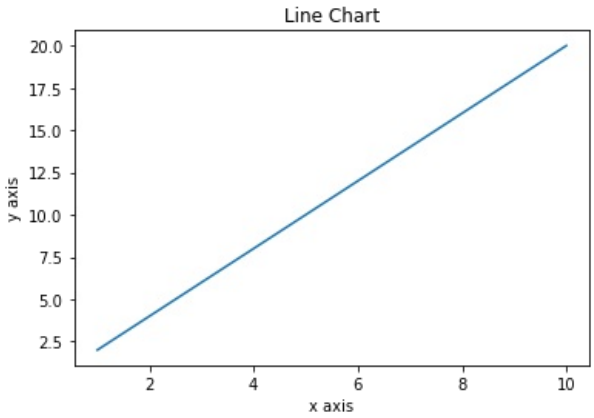
Out[6]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

In [7]:
```
y = 2*x
```

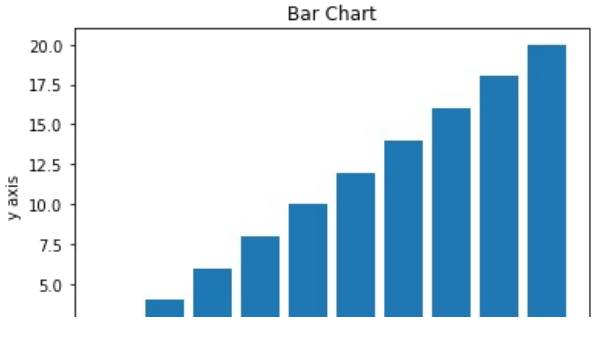
In [8]:
```
y
```

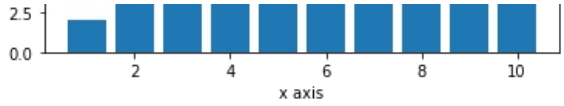
Out[8]:
```
array([ 2,  4,  6,  8, 10, 12, 14, 16, 18, 20])
```

In [9]:
```
plt.plot(x, y)
plt.title("Line Chart")
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.show()
```



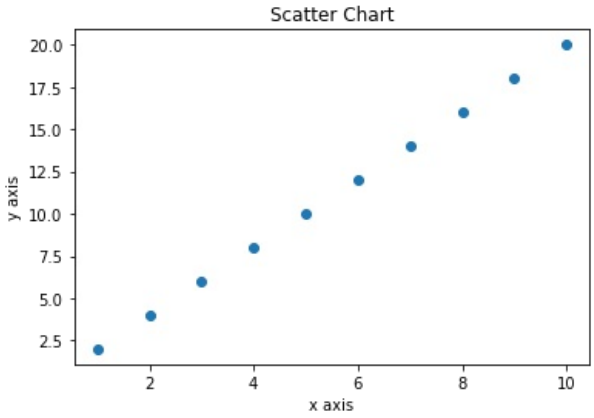
In [11]:
```
plt.bar(x, y)
plt.title("Bar Chart")
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.show()
```





In [12]:

```
plt.scatter(x, y)
plt.title("Scatter Chart")
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.show()
```

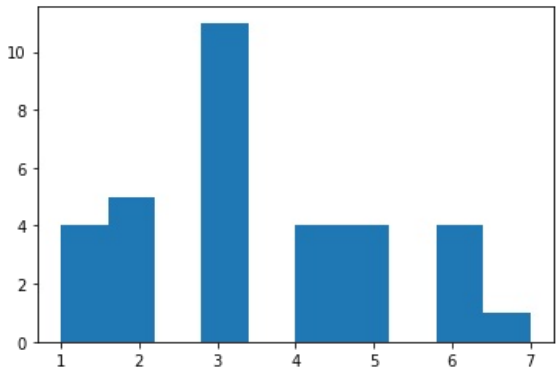


In [13]:

```
H=1, 2, 3, 3, 4, 6, 7, 4, 3, 2, 1, 2, 3, 4, 5, 5, 6, 6, 5, 4, 3, 3, 3, 3, 3, 3, 3, 5, 6, 2, 1, 1, 2
```

In [14]:

```
plt.hist(H)
plt.show()
```



In [ ]:

Central Tendency of Measures

Mean Median Mode

In [ ]:
```
#Experiment no.: 3
```

In [ ]:
```
#Aim:- To perform
```

In [3]:
```
#Name :- Gulam Jawwad Khan
#Roll No.:- 43
#Section:- A
#Subject:- P.E.1 (DSS)
#Date:- 27/07/2024
```

In [24]:
```
age = [20, 21, 22, 21, 20, 21, 22, 20, 21, 21, 20, 22, 21, 21, 21]
```

In [26]:
```
age
```

Out[26]:
```
[20, 21, 22, 21, 20, 21, 22, 20, 21, 21, 20, 22, 21, 21, 21]
```

In [27]:
```
import statistics
```

In [28]:
```
Mean = statistics.mean(age)
```

In [29]:
```
Mean
```

Out[29]:
```
20.933333333333334
```

In [30]:
```
Median = statistics.median(age)
```

In [31]:
```
Median
```

Out[31]:
```
21
```

In [32]:
```
Mode = statistics.mode(age)
```

In [33]:
```
Mode
```

Out[33]:
```
21
```

In [ ]:
```

```

Simple Linear Regression

In [ ]:
```
#Aim:- To Perform operaion on 'SIMPLE LINEAR REGRESSION'
```

In [ ]:
```
#Name :- Gulam Jawwad Khan
#Roll No.:- 43
#Sec :- A
#Subject :- DSS(P.E.1)
#Date :- 05/10/2024
```

In [6]:
```
import pandas as pd
```

In [7]:
```
import os
```

In [8]:
```
os.getcwd()
```

Out[8]:
```
'C:\\Users\\Lenovo'
```

In [9]:
```
df = pd.read_csv("C:\\Users\\Lenovo\\Salary.csv")
```

In [10]:
```
df
```

Out[10]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891
5	2.9	56642
6	3.0	60150
7	3.2	54445
8	3.2	64445
9	3.7	57189
10	3.9	63218
11	4.0	55794
12	4.0	56957
13	4.1	57081
14	4.5	61111
15	4.9	67938
16	5.1	66029
17	5.3	83088
18	5.9	81363
19	6.0	93940
20	6.8	91738
21	7.1	98273
22	7.9	101302
23	8.2	113812
24	8.7	109431
25	9.0	105582
26	9.5	116969
27	9.6	112635
28	10.3	122391
29	10.5	121872
30	11.2	127345

31	11.5	126756
32	12.3	128765
33	12.9	135675
34	13.5	139465

In [11]:
```
df.head()
```

Out[11]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

In [12]:
```
df.tail()
```

Out[12]:

	YearsExperience	Salary
30	11.2	127345
31	11.5	126756
32	12.3	128765
33	12.9	135675
34	13.5	139465

In [13]:
```
df.shape
```

Out[13]:
```
(35, 2)
```

In [14]:
```
df.size
```

Out[14]:
```
70
```

In [15]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35 entries, 0 to 34
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  35 non-null     float64
 1   Salary           35 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 688.0 bytes
```

In [16]:
```
df.describe()
```

Out[16]:

	YearsExperience	Salary
count	35.000000	35.000000
mean	6.308571	83945.600000
std	3.618610	32162.673003
min	1.100000	37731.000000
25%	3.450000	57019.000000
50%	5.300000	81363.000000
75%	9.250000	113223.500000
max	13.500000	139465.000000

In [17]:
```
df.isnull()
```

Out[17]:

	YearsExperience	Salary
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
5	False	False
6	False	False
7	False	False
8	False	False
9	False	False
10	False	False
11	False	False
12	False	False
13	False	False
14	False	False
15	False	False
16	False	False
17	False	False
18	False	False
19	False	False
20	False	False
21	False	False
22	False	False
23	False	False
24	False	False
25	False	False
26	False	False
27	False	False
28	False	False
29	False	False
30	False	False
31	False	False
32	False	False
33	False	False
34	False	False

In [18]:
```
df.isnull().any()
```

Out[18]:
```
YearsExperience    False
Salary             False
dtype: bool
```

In [19]:
```
df.isnull().sum()
```

Out[19]:
```
YearsExperience    0
Salary             0
dtype: int64
```

In [22]:
```
a = "ashish"
```

In [25]:
```
print(a)
```

```
ashish
```

In [26]:
```
a[0]
```

Out[26]:
```
'a'
```

In [27]:
```
a[-1]
```

Out[27]:
```
'h'
```

In [25]:
```
a[1:3]
```

Out[25]:
```
'sh'
```

In [27]:
```
a[1:4]
```

Out[27]:
```
'shi'
```

In [29]:
```
#Assiging values in X & Y

X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

#X = df['YearsExperience']
#y = df['Salary']
```

In [30]:
```
import matplotlib.pyplot as plt
```

In [31]:
```
import seaborn as sns
import numpy as np
```

In [30]:
```
print(X)
```

```
[[ 1.1]
 [ 1.3]
 [ 1.5]
 [ 2. ]
 [ 2.2]
 [ 2.9]
 [ 3. ]
 [ 3.2]
 [ 3.2]
 [ 3.7]
 [ 3.9]
 [ 4. ]
 [ 4. ]
 [ 4.1]
 [ 4.5]
 [ 4.9]
 [ 5.1]
 [ 5.3]
 [ 5.9]
 [ 6. ]
 [ 6.8]
 [ 7.1]
 [ 7.9]
 [ 8.2]
 [ 8.7]
 [ 9. ]
 [ 9.5]
 [ 9.6]
 [10.3]
 [10.5]
 [11.2]
```

```
 [11.5]
 [12.3]
 [12.9]
 [13.5]]
```

In [31]:
```
print(y)
```

```
[ 39343  46205  37731  43525  39891  56642  60150  54445  64445  57189
  63218  55794  56957  57081  61111  67938  66029  83088  81363  93940
  91738  98273 101302 113812 109431 105582 116969 112635 122391 121872
 127345 126756 128765 135675 139465]
```

In [32]:
```
#Splitting testdata into x_train,x_test,y_train,y_test
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=.3,random_state=42)
```

In [33]:
```
print(X_train)
```

```
[[12.9]
 [ 1.1]
 [ 2.2]
 [ 5.3]
 [ 9.6]
 [ 2.9]
 [ 4. ]
 [ 1.3]
 [ 1.5]
 [12.3]
 [ 2. ]
 [11.2]
 [ 8.2]
 [11.5]
 [ 3.9]
 [ 7.9]
 [ 5.9]
 [ 9. ]
 [ 3. ]
 [ 6.8]
 [13.5]
 [ 3.2]
 [ 4.5]
 [10.3]]
```

In [35]:
```
print(X_test)
```

```
[[ 9.5]
 [ 4.1]
 [ 8.7]
 [ 7.1]
 [ 4.9]
 [10.5]
 [ 6. ]
 [ 4. ]
 [ 3.2]
 [ 5.1]
 [ 3.7]]
```

In [36]:
```
print(y_train)
```

```
[135675  39343  39891  83088 112635  56642  55794  46205  37731 128765
  43525 127345 113812 126756  63218 101302  81363 105582  60150  91738
 139465  54445  61111 122391]
```

In [37]:
```
print(y_test)
```

```
[116969  57081 109431  98273  67938 121872  93940  56957  64445  66029
  57189]
```

In [40]:

In [40]:
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train, y_train)
```

Out[40]:
```
LinearRegression()
```

In [41]:
```
#Assigning Coefficient (slope) to m
m = lr.coef_
```

In [42]:
```
print("Coefficient :", m)
```

```
Coefficient : [8555.33918938]
```

In [44]:
```
#Assigning Y-intercept to a
c = lr.intercept_
```

In [45]:
```
 print("Intercept : ", c)
```

```
Intercept :  29602.07353482095
```

In [46]:
```
 lr.score(X_test,y_test) * 100
```

Out[46]:
```
91.71426108885098
```

In [ ]:
```

```

In [1]:
```
#Aim : To perform operation on logistic regression algorithm
```

In [2]:
```
#Name :- Gulam Jawwad Khan
#Roll No.:- 43
#Sec :- A
#Subject :- DSS(P.E.1)
#Date :- 05/10/2024
```

In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
import os
```

In [5]:
```
os.getcwd()
```

Out[5]:
```
'C:\\Users\\Lenovo'
```

In [7]:
```
os.chdir("C:\\Users\\Lenovo\\OneDrive\\Doc\\Desktop")
```

In [8]:
```
df = pd.read_csv("framingham.csv")
```

In [9]:
```
df.head()
```

Out[9]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRat
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	80.
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	95.
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	75.
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	65.
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	85.

In [10]:
```
df.describe()
```

Out[10]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totCh
count	4238.000000	4238.000000	4133.000000	4238.000000	4209.000000	4185.000000	4238.000000	4238.000000	4238.000000	4188.00000
mean	0.429212	49.584946	1.978950	0.494101	9.003089	0.029630	0.005899	0.310524	0.025720	236.72158
std	0.495022	8.572160	1.019791	0.500024	11.920094	0.169584	0.076587	0.462763	0.158316	44.59033
min	0.000000	32.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	107.00000
25%	0.000000	42.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	206.00000
50%	0.000000	49.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	234.00000
75%	1.000000	56.000000	3.000000	1.000000	20.000000	0.000000	0.000000	1.000000	0.000000	263.00000
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	1.000000	1.000000	1.000000	696.00000

In [11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             4238 non-null   int64  
 1   age              4238 non-null   int64  
 2   education        4133 non-null   float64
 3   currentSmoker    4238 non-null   int64  
 4   cigsPerDay       4209 non-null   float64
```

```
 5   BPMeds           4185 non-null   float64
 6   prevalentStroke  4238 non-null   int64
 7   prevalentHyp     4238 non-null   int64
 8   diabetes         4238 non-null   int64
 9   totChol          4188 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4219 non-null   float64
 13  heartRate        4237 non-null   float64
 14  glucose          3850 non-null   float64
 15  TenYearCHD       4238 non-null   int64
dtypes: float64(9), int64(7)
memory usage: 529.9 KB
```

In [12]:
```
df.isna().sum()
```

Out[12]:
```
male                0
age                 0
education         105
currentSmoker       0
cigsPerDay         29
BPMeds             53
prevalentStroke     0
prevalentHyp        0
diabetes            0
totChol            50
sysBP               0
diaBP               0
BMI                19
heartRate           1
glucose           388
TenYearCHD          0
dtype: int64
```

In [13]:
```
df
```

Out[13]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heart
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4233	1	50	1.0	1	1.0	0.0	0	1	0	313.0	179.0	92.0	25.97	
4234	1	51	3.0	1	43.0	0.0	0	0	0	207.0	126.5	80.0	19.71	
4235	0	48	2.0	1	20.0	NaN	0	0	0	248.0	131.0	72.0	22.00	
4236	0	44	1.0	1	15.0	0.0	0	0	0	210.0	126.5	87.0	19.16	
4237	0	52	2.0	0	0.0	0.0	0	0	0	269.0	133.5	83.0	21.47	

4238 rows × 16 columns

Missing Value Treatment

Since,'glucose' and 'education' columns had a significant amount of all nul values,so we replaced them with the mean of values for their respective columns

In [14]:
```
df['glucose'].fillna(value = df['glucose'].mean(),inplace=True)
```

In [15]:
```
df['education'].fillna(value = df['education'].mean(),inplace=True)
```

In [16]:
```
df['heartRate'].fillna(value = df['heartRate'].mean(),inplace=True)
```

In [17]:
```
df['BMI'].fillna(value = df['BMI'].mean(),inplace=True)
```

In [18]:
```
df['cigsPerDay'].fillna(value = df['cigsPerDay'].mean(),inplace=True)
```

In [19]:
```
df['totChol'].fillna(value = df['totChol'].mean(),inplace=True)
```

In [20]:
```
df['BPMeds'].fillna(value = df['BPMeds'].mean(),inplace=True)
```

In [21]:
```
df.isna().sum()
```

Out[21]:
```
male               0
age                0
education          0
currentSmoker      0
cigsPerDay         0
BPMeds             0
prevalentStroke    0
prevalentHyp       0
diabetes           0
totChol            0
sysBP              0
diaBP              0
BMI                0
heartRate          0
glucose            0
TenYearCHD         0
dtype: int64
```

In [22]:
```
#Splitting the dependent and independent variables.
x = df.drop("TenYearCHD",axis=1)
y = df['TenYearCHD']
```

In [23]:
```
x #checking the features
```

Out[23]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heart
0	1	39	4.0	0	0.0	0.00000	0	0	0	195.0	106.0	70.0	26.97	
1	0	46	2.0	0	0.0	0.00000	0	0	0	250.0	121.0	81.0	28.73	
2	1	48	1.0	1	20.0	0.00000	0	0	0	245.0	127.5	80.0	25.34	
3	0	61	3.0	1	30.0	0.00000	0	1	0	225.0	150.0	95.0	28.58	
4	0	46	3.0	1	23.0	0.00000	0	0	0	285.0	130.0	84.0	23.10	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4233	1	50	1.0	1	1.0	0.00000	0	1	0	313.0	179.0	92.0	25.97	
4234	1	51	3.0	1	43.0	0.00000	0	0	0	207.0	126.5	80.0	19.71	
4235	0	48	2.0	1	20.0	0.02963	0	0	0	248.0	131.0	72.0	22.00	
4236	0	44	1.0	1	15.0	0.00000	0	0	0	210.0	126.5	87.0	19.16	
4237	0	52	2.0	0	0.0	0.00000	0	0	0	269.0	133.5	83.0	21.47	

4238 rows × 15 columns

Train Test Split

In [26]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

In [27]:
```
y_train
```

Out[27]:
```
3252    0
3946    0
1261    0
2536    0
4089    0
       ..
3444    0
466     0
3092    0
3772    0
```

```
860     0
Name: TenYearCHD, Length: 3390, dtype: int64
```

Logistic Regression Algorithm

In [28]:
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression().fit(x_train,y_train)
model.score(x_train, y_train)
```

Out[28]:
```
0.8495575221238938
```

In [ ]:
```

```

In [1]:
```
#Aim : To perform operation on KNN (K Nearest Neighbor)
```

In [2]:
```
#Name :- Gulam Jawwad Khan
#Roll No.:- 42
#Sec :- A
#Subject :- DSS(P.E.1)
#Date :- 05/10/2024
```

importing libraries

In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
import os
```

In [5]:
```
os.getcwd()
```

Out[5]:
```
'C:\\Users\\Lenovo'
```

In [6]:
```
os.chdir("C:\\Users\\Lenovo\\OneDrive\\Doc\\Desktop")
```

In [7]:
```
df = pd.read_csv("framingham.csv")
```

In [8]:
```
df.head()
```

Out[8]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRat
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	80.
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	95.
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	75.
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	65.
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	85.

In [9]:
```
df.describe()
```

Out[9]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totCh
count	4238.000000	4238.000000	4133.000000	4238.000000	4209.000000	4185.000000	4238.000000	4238.000000	4238.000000	4188.0000
mean	0.429212	49.584946	1.978950	0.494101	9.003089	0.029630	0.005899	0.310524	0.025720	236.7215
std	0.495022	8.572160	1.019791	0.500024	11.920094	0.169584	0.076587	0.462763	0.158316	44.5903
min	0.000000	32.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	107.0000
25%	0.000000	42.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	206.0000
50%	0.000000	49.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	234.0000
75%	1.000000	56.000000	3.000000	1.000000	20.000000	0.000000	0.000000	1.000000	0.000000	263.0000
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	1.000000	1.000000	1.000000	696.0000

In [10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             4238 non-null   int64  
 1   age              4238 non-null   int64  
 2   education        4133 non-null   float64
```

```
 3   currentSmoker    4238 non-null   int64
 4   cigsPerDay       4209 non-null   float64
 5   BPMeds           4185 non-null   float64
 6   prevalentStroke  4238 non-null   int64
 7   prevalentHyp     4238 non-null   int64
 8   diabetes         4238 non-null   int64
 9   totChol          4188 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4219 non-null   float64
 13  heartRate        4237 non-null   float64
 14  glucose          3850 non-null   float64
 15  TenYearCHD       4238 non-null   int64
dtypes: float64(9), int64(7)
memory usage: 529.9 KB
```

In [11]:
```
df.isna().sum()
```

Out[11]:
```
male                0
age                 0
education         105
currentSmoker       0
cigsPerDay         29
BPMeds             53
prevalentStroke     0
prevalentHyp        0
diabetes            0
totChol            50
sysBP               0
diaBP               0
BMI                19
heartRate           1
glucose           388
TenYearCHD          0
dtype: int64
```

In [12]:
```
df
```

Out[12]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heart
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4233	1	50	1.0	1	1.0	0.0	0	1	0	313.0	179.0	92.0	25.97	
4234	1	51	3.0	1	43.0	0.0	0	0	0	207.0	126.5	80.0	19.71	
4235	0	48	2.0	1	20.0	NaN	0	0	0	248.0	131.0	72.0	22.00	
4236	0	44	1.0	1	15.0	0.0	0	0	0	210.0	126.5	87.0	19.16	
4237	0	52	2.0	0	0.0	0.0	0	0	0	269.0	133.5	83.0	21.47	

4238 rows × 16 columns

Missing Value Treatment

Since, 'glucose' and 'education' columns had a significant amount of null values, so we replaced them with the mean of values for their respective columns

In [13]:
```
df['glucose'].fillna(value = df['glucose'].mean(),inplace=True)
```

In [14]:
```
df['education'].fillna(value = df['education'].mean(),inplace=True)
```

In [15]:
```
df['heartRate'].fillna(value = df['heartRate'].mean(),inplace=True)
```

In [16]:

In [16]:
```
df['BMI'].fillna(value = df['BMI'].mean(),inplace=True)
```

In [17]:
```
df['cigsPerDay'].fillna(value = df['cigsPerDay'].mean(),inplace=True)
```

In [18]:
```
df['totChol'].fillna(value = df['totChol'].mean(),inplace=True)
```

In [19]:
```
df['BPMeds'].fillna(value = df['BPMeds'].mean(),inplace=True)
```

In [20]:
```
df.isna().sum()
```

Out[20]:
```
male               0
age                0
education          0
currentSmoker      0
cigsPerDay         0
BPMeds             0
prevalentStroke    0
prevalentHyp       0
diabetes           0
totChol            0
sysBP              0
diaBP              0
BMI                0
heartRate          0
glucose            0
TenYearCHD         0
dtype: int64
```

In [21]:
```
#Splitting the dependent and independent variables.
x = df.drop("TenYearCHD",axis=1)
y = df['TenYearCHD']
```

In [22]:
```
x #checking the features
```

Out[22]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heart
0	1	39	4.0	0	0.0	0.00000	0	0	0	195.0	106.0	70.0	26.97	
1	0	46	2.0	0	0.0	0.00000	0	0	0	250.0	121.0	81.0	28.73	
2	1	48	1.0	1	20.0	0.00000	0	0	0	245.0	127.5	80.0	25.34	
3	0	61	3.0	1	30.0	0.00000	0	1	0	225.0	150.0	95.0	28.58	
4	0	46	3.0	1	23.0	0.00000	0	0	0	285.0	130.0	84.0	23.10	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4233	1	50	1.0	1	1.0	0.00000	0	1	0	313.0	179.0	92.0	25.97	
4234	1	51	3.0	1	43.0	0.00000	0	0	0	207.0	126.5	80.0	19.71	
4235	0	48	2.0	1	20.0	0.02963	0	0	0	248.0	131.0	72.0	22.00	
4236	0	44	1.0	1	15.0	0.00000	0	0	0	210.0	126.5	87.0	19.16	
4237	0	52	2.0	0	0.0	0.00000	0	0	0	269.0	133.5	83.0	21.47	

4238 rows × 15 columns

Train Test Split

In [23]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

In [24]:
```
y_train
```

Out[24]:
```
3252    0
3946    0
1261    0
2536    0
4089    0
       ..
3444    0
466     0
```

```
3092    0
3772    0
860     0
Name: TenYearCHD, Length: 3390, dtype: int64
```

KNN Classifier

In [25]:

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=5, p=2, metric='minkowski')
knn.fit(x_train, y_train)
acc = knn.score(x_test,y_test)*100
print(acc)
```

```
83.13679245283019
```

In [ ]:

In [1]:
```
#Aim : To perform operation on SVM (Support Vector Machine)
```

In [2]:
```
#Name :- Gulam Jawwad Khan
#Roll No.:- 42
#Sec :- A
#Subject :- DSS(P.E.1)
#Date :- 05/10/2024
```

In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
import os
```

In [5]:
```
os.getcwd()
```

Out[5]:
```
'C:\\Users\\Lenovo'
```

In [6]:
```
os.chdir("C:\\Users\\Lenovo\\OneDrive\\Doc\\Desktop")
```

In [7]:
```
df = pd.read_csv("framingham.csv")
```

In [8]:
```
df.head()
```

Out[8]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	80.0	77.0	0
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	95.0	76.0	0
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	75.0	70.0	0
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	65.0	103.0	1
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	85.0	85.0	0

In [9]:
```
df.describe()
```

Out[9]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
count	4238.000000	4238.000000	4133.000000	4238.000000	4209.000000	4185.000000	4238.000000	4238.000000	4238.000000	4188.000000	4238.000000	4238.000000	4219.000000	4237.000000	3850.000000	4238.000000
mean	0.429212	49.584946	1.978950	0.494101	9.003089	0.029630	0.005899	0.310524	0.025720	236.721585	132.352407	82.893464	25.802008	75.878924	81.966753	0.151958
std	0.495022	8.572160	1.019791	0.500024	11.920094	0.169584	0.076587	0.462763	0.158316	44.590334	22.038097	11.910850	4.080111	12.026596	23.959998	0.359023
min	0.000000	32.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	107.000000	83.500000	48.000000	15.540000	44.000000	40.000000	0.000000
25%	0.000000	42.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	206.000000	117.000000	75.000000	23.070000	68.000000	71.000000	0.000000
50%	0.000000	49.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	234.000000	128.000000	82.000000	25.400000	75.000000	78.000000	0.000000
75%	1.000000	56.000000	3.000000	1.000000	20.000000	0.000000	0.000000	1.000000	0.000000	263.000000	144.000000	89.875000	28.040000	83.000000	87.000000	0.000000
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	1.000000	1.000000	1.000000	696.000000	295.000000	142.500000	56.800000	143.000000	394.000000	1.000000

In [10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             4238 non-null   int64  
 1   age              4238 non-null   int64  
 2   education        4133 non-null   float64
 3   currentSmoker    4238 non-null   int64  
 4   cigsPerDay       4209 non-null   float64
 5   BPMeds           4185 non-null   float64
 6   prevalentStroke  4238 non-null   int64  
 7   prevalentHyp     4238 non-null   int64  
 8   diabetes         4238 non-null   int64  
 9   totChol          4188 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4219 non-null   float64
 13  heartRate        4237 non-null   float64
 14  glucose          3850 non-null   float64
 15  TenYearCHD       4238 non-null   int64  
dtypes: float64(9), int64(7)
memory usage: 529.9 KB
```

In [11]:
```
df.isna().sum()
```

Out[11]:
```
male                0
age                 0
education         105
currentSmoker       0
cigsPerDay         29
BPMeds             53
prevalentStroke     0
prevalentHyp        0
diabetes            0
totChol            50
sysBP               0
diaBP               0
BMI                19
heartRate           1
glucose           388
TenYearCHD          0
dtype: int64
```

In [12]:
```
df
```

Out[12]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	80.0	77.0	0
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	95.0	76.0	0
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	75.0	70.0	0
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	65.0	103.0	1
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	85.0	85.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4233	1	50	1.0	1	1.0	0.0	0	1	0	313.0	179.0	92.0	25.97	66.0	86.0	1
4234	1	51	3.0	1	43.0	0.0	0	0	0	207.0	126.5	80.0	19.71	65.0	68.0	0
4235	0	48	2.0	1	20.0	NaN	0	0	0	248.0	131.0	72.0	22.00	84.0	86.0	0
4236	0	44	1.0	1	15.0	0.0	0	0	0	210.0	126.5	87.0	19.16	86.0	NaN	0
4237	0	52	2.0	0	0.0	0.0	0	0	0	269.0	133.5	83.0	21.47	80.0	107.0	0

4238 rows × 16 columns

Missing Value Treatment

Since, 'glucose' and 'education' columns had a significant amount of null values, so we replaced them with the mean of values for their respective columns

In [13]:
```
df['glucose'].fillna(value = df['glucose'].mean(),inplace=True)
```

In [14]:
```
df['education'].fillna(value = df['education'].mean(),inplace=True)
```

In [15]:
```
df['heartRate'].fillna(value = df['heartRate'].mean(),inplace=True)
```

In [16]:
```
df['BMI'].fillna(value = df['BMI'].mean(),inplace=True)
```

In [17]:
```
df['cigsPerDay'].fillna(value = df['cigsPerDay'].mean(),inplace=True)
```

In [18]:
```
df['totChol'].fillna(value = df['totChol'].mean(),inplace=True)
```

In [19]:
```
df['BPMeds'].fillna(value = df['BPMeds'].mean(),inplace=True)
```

In [20]:
```
df.isna().sum()
```

Out[20]:
```
male               0
age                0
education          0
currentSmoker      0
cigsPerDay         0
BPMeds             0
prevalentStroke    0
prevalentHyp       0
diabetes           0
totChol            0
sysBP              0
diaBP              0
BMI                0
heartRate          0
glucose            0
TenYearCHD         0
dtype: int64
```

In [21]:
```
#Splitting the dependent and independent variables.
x = df.drop("TenYearCHD",axis=1)
y = df['TenYearCHD']
```

In [22]:
```
x #checking the features
```

Out[22]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose
0	1	39	4.0	0	0.0	0.00000	0	0	0	195.0	106.0	70.0	26.97	80.0	77.000000
1	0	46	2.0	0	0.0	0.00000	0	0	0	250.0	121.0	81.0	28.73	95.0	76.000000
2	1	48	1.0	1	20.0	0.00000	0	0	0	245.0	127.5	80.0	25.34	75.0	70.000000
3	0	61	3.0	1	30.0	0.00000	0	1	0	225.0	150.0	95.0	28.58	65.0	103.000000
4	0	46	3.0	1	23.0	0.00000	0	0	0	285.0	130.0	84.0	23.10	85.0	85.000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4233	1	50	1.0	1	1.0	0.00000	0	1	0	313.0	179.0	92.0	25.97	66.0	86.000000
4234	1	51	3.0	1	43.0	0.00000	0	0	0	207.0	126.5	80.0	19.71	65.0	68.000000
4235	0	48	2.0	1	20.0	0.02963	0	0	0	248.0	131.0	72.0	22.00	84.0	86.000000
4236	0	44	1.0	1	15.0	0.00000	0	0	0	210.0	126.5	87.0	19.16	86.0	81.966753
4237	0	52	2.0	0	0.0	0.00000	0	0	0	269.0	133.5	83.0	21.47	80.0	107.000000

4238 rows × 15 columns

Test Train Split

In [23]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

In [24]:
```
y_train
```

Out[24]:
```
3252    0
3946    0
1261    0
2536    0
4089    0
       ..
3444    0
466     0
3092    0
3772    0
860     0
Name: TenYearCHD, Length: 3390, dtype: int64
```

SVM Classifier

In [25]:
```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
svc=SVC()
svc.fit(x_test,y_test)
acc = svc.score(x_test,y_test)*100
print(acc)
```

```
85.37735849056604
```

In [ ]:
```

```