

Name : Vishwajit S. Deshmukh

Roll no : 70

Sec : A

Aim : Study of Anaconda IDE and it's Installation

Theroy :

Steps to Install :

1. Download Anaconda Installer

- Go to the Anaconda Downloads page.
- Select **Windows** as your operating system.
- Choose the **Python 3.x** version (recommended) and click **Download**.

2. Launch the Installer

- Once the download is complete, open the **.exe** installer file.
- You will be prompted with the **Anaconda Setup** window.
- Click **Next**.

3. Accept License Agreement

- Read through the **license agreement** and click **I Agree**.

4. Select Installation Type

- You can install Anaconda for **Just Me** or for **All Users** (admin privileges required).
- Select your option and click **Next**.

5. Choose Installation Location

- Choose the folder where you want to install Anaconda. You can leave the default location or select another.
- Click **Next**.

6. Select Advanced Options

- It's recommended to leave **Add Anaconda to my PATH environment variable** unchecked (as this can cause issues with other software).
- Keep the **Register Anaconda as my default Python 3.x** box checked.
- Click **Install**.

7. Installation Progress

- The installation will begin. This can take several minutes.

8. Finish Installation

- After the installation is complete, click **Next**.

9. Optional: Install Microsoft VSCode

- The installer will ask if you want to install **Microsoft VSCode**. This is optional and can be skipped by unchecking the box.

10. Complete

- Click **Finish** to complete the installation process.

Once Anaconda is installed, you can launch **Anaconda Navigator** or use the **Anaconda Prompt** to start working with Python, Jupyter Notebooks, and other tools.

Exp-2-DSS-Data-Acquisition

October 13, 2024

1 Data Aquisition

```
[1]: # Experiment No. 2
```

```
[1]: # Name : Vishwajit S. Deshmukh  
# Roll : 70  
# s e c :
```

```
[ ]: # Aim : Data Aquisition Using Pandas library
```

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

```
[3]: os.getcwd()
```

```
[3]: 'C:\\Users\\Hp'
```

```
[5]: path = os.chdir('E://Downloads')
```

```
[6]: data = pd.read_csv('diabetes.csv')
```

```
[7]: data
```

```
[7]:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	
..	
763	10	101	76	48	180	32.9	
764	2	122	70	27	0	36.8	
765	5	121	72	23	112	26.2	
766	1	126	60	0	0	30.1	
767	1	93	70	31	0	30.4	

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1
..
763	0.171	63	0
764	0.340	27	0
765	0.245	30	0
766	0.349	47	1
767	0.315	23	0
768	rows x 9 columns]		

```
[8]: data.head
```

```
[8]: <bound method NDFrame.head of Pregnancies Glucose
      BloodPressure
```

	SkinThickness	Insulin	BMI \			
0	6	148	72	35	0	33.6
1	1	85	66	29	0	26.6
2	8	183	64	0	0	23.3
3	1	89	66	23	94	28.1
4	0	137	40	35	168	43.1
..
763	10	101	76	48	180	32.9
764	2	122	70	27	0	36.8
765	5	121	72	23	112	26.2
766	1	126	60	0	0	30.1
767	1	93	70	31	0	30.4

```
DiabetesPedigreeFunction Age Outcome
```

0	0.627	50	1
1	0.351	31	0

```

2          0.672 32    1
3          0.167 21    0
4          2.288 33    1
..          ... ..    ...
763        0.171 63    0
764        0.340 27    0
765        0.245 30    0
766        0.349 47    1
767        0.315 23    0

```

```
[768 rows x 9 columns]>
```

```
[9]: data.tail
```

```
[9]: <bound method NDFrame.tail of Pregnancies Glucose
      BloodPressure
```

```

SkinThickness  Insulin  BMI  \
0              6   148   72   35    0 33.6
1              1    85   66   29    0 26.6
2              8   183   64    0    0 23.3
3              1    89   66   23   94 28.1
4              0   137   40   35  168 43.1
..          ...    ...    ...    ...    ...
763           10  101   76   48  180 32.9
764            2  122   70   27    0 36.8
765            5  121   72   23  112 26.2
766            1  126   60    0    0 30.1
767            1   93   70   31    0 30.4

```

```
DiabetesPedigreeFunction  Age  Outcome
```

```

0          0.627 50    1
1          0.351 31    0
2          0.672 32    1
3          0.167 21    0
4          2.288 33    1

```

```

..
... ..
763      0.171 63 0
764      0.340 27 0
765      0.245 30 0
766      0.349 47 1
767      0.315 23 0

```

```
[768 rows x 9 columns]>
```

```
[13]: data.head(15)
```

```

[13]: Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI \
0           6     148     72      35      0  33.6
1           1     85     66     29      0  26.6
2           8    183     64      0      0  23.3
3           1     89     66     23     94  28.1
4           0    137     40     35    168  43.1
5           5    116     74      0      0  25.6
6           3     78     50     32     88  31.0
7          10    115      0      0      0  35.3
8           2    197     70     45    543  30.5
9           8    125     96      0      0   0.0
10          4    110     92      0      0  37.6
11         10    168     74      0      0  38.0
12         10    139     80      0      0  27.1
13          1    189     60     23    846  30.1
14          5    166     72     19    175  25.8

      DiabetesPedigreeFunction  Age  Outcome
0           0.627      50      1
1           0.351      31      0
2           0.672      32      1
3           0.167      21      0
4           2.288      33      1

```

5	0.201	30	0
6	0.248	26	1
7	0.134	29	0
8	0.158	53	1
9	0.232	54	1
10	0.191	30	0
11	0.537	34	1
12	1.441	57	0
13	0.398	59	1
14	0.587	51	1

```
[14]: data.tail(30)
```

```
[14]: Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin  BMI \
738          2    99      60      17      160 36.6
739          1   102      74       0       0 39.5
740         11   120      80      37      150 42.3
741          3   102      44      20      94 30.8
742          1   109      58      18     116 28.5
743          9   140      94       0       0 32.7
744         13   153      88      37      140 40.6
745         12   100      84      33     105 30.0
746          1   147      94      41       0 49.3
747          1    81      74      41      57 46.3
748          3   187      70      22     200 36.4
749          6   162      62       0       0 24.3
750          4   136      70       0       0 31.2
751          1   121      78      39      74 39.0
752          3   108      62      24       0 26.0
753          0   181      88      44     510 43.3
754          8   154      78      32       0 32.4
```

755	1	128	88	39	110	36.5
756	7	137	90	41	0	32.0
757	0	123	72	0	0	36.3
758	1	106	76	0	0	37.5
759	6	190	92	0	0	35.5
760	2	88	58	26	16	28.4
761	9	170	74	31	0	44.0
762	9	89	62	0	0	22.5
763	10	101	76	48	180	32.9
764	2	122	70	27	0	36.8
765	5	121	72	23	112	26.2
766	1	126	60	0	0	30.1
767	1	93	70	31	0	30.4

DiabetesPedigreeFunction Age Outcome

738	0.453	21	0
739	0.293	42	1
740	0.785	48	1
741	0.400	26	0
742	0.219	22	0
743	0.734	45	1
744	1.174	39	0
745	0.488	46	0
746	0.358	27	1
747	1.096	32	0
748	0.408	36	1
749	0.178	50	1
750	1.182	22	1
751	0.261	28	0
752	0.223	25	0
753	0.222	26	1

754	0.443	45	1
755	1.057	37	1
756	0.391	39	0
757	0.258	52	1
758	0.197	26	0
759	0.278	66	1
760	0.766	22	0
761	0.403	43	1
762	0.142	33	0
763	0.171	63	0
764	0.340	27	0
765	0.245	30	0
766	0.349	47	1
767	0.315	23	0

[

]:

[]:

Exp-3-DSS-Stastical-Description

October 13, 2024

Stastical Description

```
[1]: # Experiment No. 3
```

```
[2]: # Name : Vishwajit S. Deshmukh
      # Roll : 70
      # s e c :
```

```
[17]: # Aim : To perform Stastical Description on data
```

```
[1]: import pandas as pd
import os
```

```
[2]: os.getcwd()
```

```
[2]: 'C:\\Users\\Hp'
```

```
[6]: path = os.chdir('E://Downloads')
```

```
[7]: data = pd.read_csv('framingham.csv')
```

```
[8]: data
```

```
[8]: male age education currentSmoker cigsPerDay BPMeds \
```

0	1	39	4.0	0	0.0	0.0
---	---	----	-----	---	-----	-----

1	0	46	2.0	0	0.0	0.0
---	---	----	-----	---	-----	-----

2	1	48	1.0	1	20.0	0.0
---	---	----	-----	---	------	-----

```
3      0    61    3.0    1    30.0    0.0
```

4	0	46	3.0	1	23.0	0.0
---	---	----	-----	---	------	-----

*** *** *** *** *** ***

4233	1	50	1.0	1	1.0	0.0
------	---	----	-----	---	-----	-----

4234	1	51	3.0	1	43.0	0.0
------	---	----	-----	---	------	-----

4235	0	48	2.0	1	20.0	NaN
------	---	----	-----	---	------	-----

4236	0	44	1.0	1	15.0	0.0
------	---	----	-----	---	------	-----

4237	0	52	2.0	0	0.0	0.0
------	---	----	-----	---	-----	-----

```
prevalentStroke prevalentHyp diabetes totChol sysBP diaBP BMI \
```

0	0 195.0	106.0	70.0
---	---------	-------	------

26.97

1	250.0	121.0	81.0	28.73
---	-------	-------	------	-------

0

0

```

0 0 0

2 0 0 245.0 127.5
80.0 25.34
3 0 1 0 225.0 150.0
95.0 28.58
4 0 0 285.0 130.0
84.0 23.10

...
4233 0 1 0 313.0 179.0 92.0 25.97
4234 0 0 0 207.0 126.5 80.0 19.71
4235 0 0 0 248.0 131.0 72.0 22.00
4236 0 0 0 210.0 126.5 87.0 19.16
4237 0 0 0 269.0 133.5 83.0 21.47

```

```

heartRate glucose TenYearCHD
0 80.0 77.0 0
1 95.0 76.0 0
2 75.0 70.0 0
3 65.0 103.0 1
4 85.0 85.0 0

...
4233 66.0 86.0 1
4234 65.0 68.0 0
4235 84.0 86.0 0
4236 86.0 NaN 0
4237 80.0 107.0 0

```

[4238 rows x 16 columns]

```
[9]: data.head
```

```
[9]: <bound method NDFrame.head of male age education currentSmoker
cigsPerDay BPMeds \
```

```

0 1 39 4.0 0 0.0 0.0
1 0 46 2.0 0 0.0 0.0
2 1 48 1.0 1 20.0 0.0
3 0 61 3.0 1 30.0 0.0
4 0 46 3.0 1 23.0 0.0

...
4233 1 50 1.0 1 1.0 0.0
4234 1 51 3.0 1 43.0 0.0
4235 0 48 2.0 1 20.0 NaN
4236 0 44 1.0 1 15.0 0.0
4237 0 52 2.0 0 0.0 0.0

```

```

prevalentStroke prevalentHyp diabetes totChol sysBP diaBP BMI \
0 0

```

		0		0		0			
0					00	0	195.0	106.0	
					70.0	26.97			
1					00	0	250.0	121.0	
					81.0	28.73			
2					00	0	245.0	127.5	
					80.0	25.34			
3					1225.0	150.0	95.0	28.58	
4					285.0	130.0	84.0	23.10	
...		
4233		0	1	0	313.0	179.0	92.0	25.97	
4234		0	0	0	207.0	126.5	80.0	19.71	
4235		0	0	0	248.0	131.0	72.0	22.00	
4236		0	0	0	210.0	126.5	87.0	19.16	
4237		0	0	0	269.0	133.5	83.0	21.47	

	heartRate	glucose	TenYearCHD
0	80.0	77.0	0
1	95.0	76.0	0
2	75.0	70.0	0
3	65.0	103.0	1
4	85.0	85.0	0
...
4233	66.0	86.0	1
4234	65.0	68.0	0
4235	84.0	86.0	0
4236	86.0	NaN	0
4237	80.0	107.0	0

[4238 rows x 16 columns]>

```
[10]: data.tail
```

```
[10]: <bound method NDFrame.tail of male age education currentSmoker
cigsPerDay BPMeds \
```

0	1	39	4.0	0	0.0	0.0
1	0	46	2.0	0	0.0	0.0
2	1	48	1.0	1	20.0	0.0
3	0	61	3.0	1	30.0	0.0
4	0	46	3.0	1	23.0	0.0
				0		0

0 0 0

```

...      ...      ...      ...      ...      ...
4233    1  50    1.0  1    1.0  0.0
4234    1  51    3.0  1    43.0  0.0
4235    0  48    2.0  1    20.0 NaN
4236    0  44    1.0  1    15.0  0.0
4237    0  52    2.0  0    0.0  0.0

```

```

prevalentStroke prevalentHyp diabetes totChol sysBP diaBP BMI \
0          0      0      0      195.0 106.0      70.0 26.97
1          0      0      0      250.0 121.0      81.0 28.73
2          0      0      0      245.0 127.5      80.0 25.34
3          0      1      0      225.0 150.0      95.0 28.58
4          0      0      0      285.0 130.0      84.0 23.10

```

```

...      ...      ...      ...      ...      ...      ...
4233          1 313.0 179.0      92.0
          25.97
4234          207.0 126.5 80.0 19.71
4235          0 0      0      248.0 131.0
          72.0 22.00
4236          0 0      0      210.0 126.5
          87.0 19.16
4237          0 0      0      269.0 133.5
          83.0 21.47

```

```

heartRate glucose TenYearCHD
0          80.0  77.0  0
1          95.0  76.0  0
2          75.0  70.0  0
3          65.0 103.0  1
4          85.0  85.0  0
...      ...      ...      ...
4233          66.0  86.0  1
4234          65.0  68.0  0
4235          84.0  86.0  0
4236          86.0  NaN   0
4237          80.0 107.0  0

```

[4238 rows x 16 columns]>

0 0

0 0 0

```
[11]: data.info
```

```
[11]: <bound method DataFrame.info of male age education currentSmoker
```

```
cigsPerDay BPMeds \
```

```
0 1 39 4.0 0 0.0 0.0
```

```
1 0 46 2.0 0 0.0 0.0 2 1 48 1.0 1
```

```
20.0 0.0
```

```
3 0 61 3.0 1 30.0 0.0
```

```
4 0 46 3.0 1 23.0 0.0
```

```
... ..
```

```
4233 1 50 1.0 1 1.0 0.0 4234 1 51
```

```
3.0 1 43.0 0.0
```

```
4235 0 48 2.0 1 20.0 NaN
```

```
4236 0 44 1.0 1 15.0 0.0
```

```
4237 0 52 2.0 0 0.0 0.0
```

```
prevalentStroke prevalentHyp diabetes totChol sysBP diaBP BMI \
```

```
0 0 0 0 195.0 106.0 70.0 26.97
```

```
1 0 0 0 250.0 121.0 81.0 28.73
```

```
2 0 0 0 245.0 127.5 80.0 25.34
```

```
3 0 1 0 225.0 150.0 95.0 28.58
```

```
4 0 0 0 285.0 130.0 84.0 23.10
```

```
... ..
```

```
4233 01 0 313.0 179.0
```

```
92.0 25.97
```

```
4234 00 0 207.0 126.5
```

```
80.0 19.71
```

```
4235 00 0 248.0 131.0
```

```
72.0 22.00
```

```
4236 0210.0 126.5 87.0 19.16
```

0 0

0 0 0

4237 269.0 133.5 83.0 21.47

	heartRate	glucose	TenYearCHD
0	80.0	77.0	0
1	95.0	76.0	0
2	75.0	70.0	0
3	65.0	103.0	1
4	85.0	85.0	0

...

4233	66.0	86.0	1
4234	65.0	68.0	0
4235	84.0	86.0	0
4236	86.0	NaN	0
4237	80.0	107.0	0

[4238 rows x 16 columns]>

[12]: data.count

[12]: <bound method DataFrame.count of male age education currentSmoker
cigsPerDay BPMeds \

0	1	39	4.0	0	0.0	0.0
1	0	46	2.0	0	0.0	0.0
		20.0	0.0			
3	0	61	3.0	1	30.0	0.0
4	0	46	3.0	1	23.0	0.0
...
4233	1	50	1.0	1	1.0	0.0
	3.0	1	43.0	0.0		
4235	0	48	2.0	1	20.0	NaN
4236	0	44	1.0	1	15.0	0.0
4237	0	52	2.0	0	0.0	0.0

	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI \
0	0	0	0	195.0	106.0	70.0	26.97
1	0	0	0	250.0	121.0	81.0	28.73
2	0	0	0	245.0	127.5	80.0	25.34
3	0	1	0	225.0	150.0	95.0	28.58
4	0	0	0	285.0	130.0	84.0	23.10

```

...
...
...
...
...
...
...
4233      0      1      0      313.0 179.0      92.0 25.97
4234      0      0      0      207.0 126.5      80.0 19.71
4235      0      0      0      248.0 131.0      72.0 22.00
4236      0      0      0      210.0 126.5      87.0 19.16
4237      0      0      0      269.0 133.5      83.0 21.47

```

```

      heartRate glucose TenYearCHD
0      80.0      77.0 0
1      95.0      76.0 0
2      75.0      70.0 0
3      65.0     103.0 1
4      85.0      85.0 0

```

```

...
...
...
...
4233      66.0      86.0 1
4234      65.0      68.0 0
4235      84.0      86.0 0
4236      86.0      NaN  0
4237      80.0     107.0 0

```

```
[4238 rows x 16 columns]>
```

```
[16]: data.describe()
```

```

[16]:
      count  male      age  education  currentSmoker  cigsPerDay \
count  4238.000000  4238.000000  4133.000000  4238.000000  4209.000000
mean    0.429212   49.584946    1.978950    0.494101    9.003089
std     0.495022    8.572160    1.019791    0.500024   11.920094
min     0.000000   32.000000    1.000000    0.000000    0.000000
25%     0.000000   42.000000    1.000000    0.000000    0.000000
50%     0.000000   49.000000    2.000000    0.000000    0.000000
75%     1.000000   56.000000    3.000000    1.000000   20.000000
max     1.000000   70.000000    4.000000    1.000000   70.000000

      BPMeds prevalentStroke prevalentHyp  diabetes  totChol \

```


count	4185.000000	4238.000000	4238.000000	4238.000000
		4188.000000		
mean	0.029630	0.005899	0.310524	0.025720 236.721585
std	0.169584	0.076587	0.462763	0.158316 44.590334
min	0.000000	0.000000	0.000000	0.000000 107.000000
25%	0.000000	0.000000	0.000000	0.000000 206.000000
50%	0.000000	0.000000	0.000000	0.000000 234.000000
75%	0.000000	0.000000	1.000000	0.000000 263.000000
max	1.000000	1.000000	1.000000	1.000000 696.000000

	sysBP	diaBP	BMI	heartRate	glucose \
count	4238.000000	4238.000000	4219.000000	4237.000000	3850.000000
mean	132.352407	82.893464	25.802008	75.878924	81.966753
std	22.038097	11.910850	4.080111	12.026596	23.959998
min	83.500000	48.000000	15.540000	44.000000	40.000000
25%	117.000000	75.000000	23.070000	68.000000	71.000000
50%	128.000000	82.000000	25.400000	75.000000	78.000000
75%	144.000000	89.875000	28.040000	83.000000	87.000000
max	295.000000	142.500000	56.800000	143.000000	394.000000

	TenYearCHD
count	4238.000000
mean	0.151958
std	0.359023
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

```
[13]: data.shape
```

```
[ ]:
```

```
[13]: (4238, 16)
```

```
[14]: data.size
```

```
[14]: 67808
```

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

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

Exp-4-DSS-Data-Manipulation

October 13, 2024

1 Data Manipulation

```
[1]: # Experiment No. 4
```

```
[2]: # Name : Vishwajit S. Deshmukh  
# Roll : 70  
# s e c :
```

```
[ ]: # Aim : To perform Data Manipulation using Pandas
```

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

```
[1]: import os
```

```
[2]: os.getcwd()
```

```
[2]: 'C:\\\\Users\\Hp'
```

```
[6]: os.chdir("E:\\Downloads")
```

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

```
[8]: df
```

```
[8]:
```

	PassengerId	Survived	Pclass	\
0	892	0	3	
1	893	1	3	
2	894	0	2	
3	895	0	3	
4	896	1	3	
..	
413	1305	0	3	
414	1306	1	1	
415	1307	0	3	
416	1308	0	3	

417 1309 0 3

```

                                Name      Sex   Age SibSp Parch \
0                               Kelly, Mr. James      male 34.5   0     0
1                               Wilkes, Mrs. James (Ellen Needs) female 47.0
                                1     0
2                               Myles, Mr. Thomas Francis male 62.0   0     0
3                               Wirz, Mr. Albert      male 27.0   0     0
4                               Hirvonen, Mrs. Alexander (Helga E Lindqvist)
                                female 22.0   1     1
..                               ...      ...   ...   ...   ...
413                              Spector, Mr. Woolf      male NaN    0     0
414                              Oliva y Ocana, Dona. Fermina female 39.0     0
                                0
415                              Saether, Mr. Simon Sivertsen      male 38.5   0
                                0
416                              Ware, Mr. Frederick      male NaN    0     0
417                              Peter, Master. Michael J      male NaN    1     1
```

```

                        Ticket      Fare Cabin Embarked
0                        330911  7.8292     NaN    Q
1                        363272  7.0000     NaN    S
2                        240276  9.6875     NaN    Q
3                        315154  8.6625     NaN    S
4                        3101298 12.2875     NaN    S
..                        ...      ...   ...   ...
413                       A.5. 3236  8.0500     NaN    S
414                       PC 17758 108.9000 C105     C
415                       SOTON/O.Q. 3101262  7.2500     NaN    S
416                       359309  8.0500     NaN    S
417                       2668  22.3583     NaN    C
```

418 rows x 12 columns]

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

```
[9]: PassengerId Survived Pclass \
```

```
0      892      0      3
1      893      1      3
2      894      0      2
3      895      0      3
4      896      1      3
```

```

                                Name      Sex      Age SibSp Parch \
0                                Kelly, Mr. James male 34.5  0      0
1                                Wilkes, Mrs. James (Ellen Needs) female
                                47.0  1      0
2                                Myles, Mr. Thomas Francis      male 62.0  0
                                0
3                                Wirz, Mr. Albert male 27.0  0      0
4                                Hirvonen, Mrs. Alexander (Helga E
                                Lindqvist) female 22.0 1      1
```

```

      Ticket   Fare Cabin Embarked
0   330911    7.8292   NaN      Q
1   363272    7.0000   NaN      S
2   240276    9.6875   NaN      Q
3   315154    8.6625   NaN      S
4   3101298  12.2875   NaN      S
```

```
[19]: df.info()
```

```
<class
'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to
417 Data columns (total 12
columns):
# Column Non-Null Count Dtype --- -
-----
0   PassengerId 418 non-null int64
1   Survived    418 non-null  int64
2   Pclass      418 non-null  int64
3   Name        418 non-null  object
4   Sex         418 non-null  object
5   Age         332 non-null  float64
6   SibSp       418 non-null  int64
7   Parch       418 non-null  int64
8   Ticket      418 non-null  object
9   Fare        417 non-null  float64
10  Cabin        91 non-null   object
11  Embarked     418 non-null  object
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

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

```
[11]:      PassengerId  Survived  Pclass     Age  SibSp \
count  418.000000  418.000000  418.000000  332.000000  418.000000
mean    1100.500000    0.363636    2.265550   30.272590    0.447368
std      120.810458    0.481622    0.841838   14.181209    0.896760
min       892.000000    0.000000    1.000000    0.170000    0.000000
25%      996.250000    0.000000    1.000000   21.000000    0.000000
50%     1100.500000    0.000000    3.000000   27.000000    0.000000
75%     1204.750000    1.000000    3.000000   39.000000    1.000000
max     1309.000000    1.000000    3.000000   76.000000    8.000000

      Parch     Fare
count  418.000000  417.000000
mean    0.392344   35.627188
std     0.981429   55.907576
min     0.000000    0.000000
25%     0.000000    7.895800
50%     0.000000   14.454200
75%     0.000000   31.500000
max      9.000000  512.329200
```

```
[12]: df.shape
```

```
[12]: (418, 12)
```

```
[13]: df.size
```

```
[13]: 5016
```

```
[14]: df.ndim
```

```
[14]: 2
```

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

```
[16]: Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age',
'SibSp',
        'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
dtype='object')
```

```
[17]: df.drop(labels="Age",axis=1)
```

```
[17]: PassengerId Survived Pclass \
0      892    0      3
1      893    1      3
2      894    0      2
3      895    0      3
4      896    1      3
..      ...    ...    ...
413    1305    0      3
414    1306    1      1
415    1307    0      3
416    1308    0      3
417    1309    0      3
```

```
                                Name      Sex SibSp Parch \
0      Kelly, Mr. James      male  0      0
1      Wilkes, Mrs. James (Ellen Needs) female  1
      0
2      Myles, Mr. Thomas Francis male  0      0
3      Wirz, Mr. Albert      male  0      0
4      Hirvonen, Mrs. Alexander (Helga E Lindqvist)
      female  1      1
```

```
..      ...      ...      ...      ...
413      Spector, Mr. Woolf      male  0      0
414      Oliva y Ocana, Dona. Fermina female  0      0
415      Saether, Mr. Simon Sivertsen      male  0      0
416      Ware, Mr. Frederick      male  0      0
417      Peter, Master. Michael J      male  1      1
```

```
                                Ticket      Fare Cabin Embarked
0      330911  7.8292      NaN  Q
1      363272  7.0000      NaN  S
2      240276  9.6875      NaN  Q
```

```

3          315154 8.6625      NaN  S
4          310129812.2875      NaN  S
..          ...      ... ..      ...
413        A.5. 3236  8.0500      NaN  S
414        PC 17758 108.9000 C105      C
415        SOTON/O.Q. 3101262  7.2500      NaN  S
416        359309  8.0500      NaN  S
417        2668 22.3583      NaN  C

```

[418 rows x 11 columns]

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

```

[18]: PassengerId Survived Pclass \
0      892  0      3
1      893  1      3
3      895  0      3
4      896  1      3
5      897  0      3
..      ...      ...      ...
413    1305  0      3
414    1306  1      1
415    1307  0      3
416    1308  0      3
417    1309  0      3

```

```

                                Name      Sex  Age SibSp Parch \
0      Kelly, Mr. James      male  34.5  0      0
1      Wilkes, Mrs. James (Ellen Needs) female  47.0
      1      0
3      Wirz, Mr. Albert      male  27.0  0      0
4      Hirvonen, Mrs. Alexander (Helga E Lindqvist)
      female  22.0  1      1
5      Svensson, Mr. Johan Cervin      male  14.0  0
      0

```

```

..
413      Spector, Mr. Woolf      male NaN  0    0
414      Oliva y Ocana, Dona. Fermina female 39.0    0
415      Saether, Mr. Simon Sivertsen      male 38.5  0
416      Ware, Mr. Frederick      male NaN  0    0
417      Peter, Master. Michael J      male NaN  1    1

```

```

      Ticket      Fare Cabin Embarked
0      330911  7.8292      NaN    Q
1      363272  7.0000      NaN    S
3      315154      8.6625      NaN    S
4      3101298      12.2875      NaN    S
5      7538  9.2250      NaN    S
..
      ...      ... ..      ...
413      A.5. 3236  8.0500      NaN    S
414      PC 17758 108.9000 C105      C
415      SOTON/O.Q. 3101262      7.2500      NaN    S
416      359309      8.0500      NaN    S
417      2668 22.3583      NaN    C

```

[417 rows x 12 columns] []:



Exp-5-DSS-Array-using-numpy

October 13, 2024

[14]:

```
[9]: # Practical no . 5
```

```
[1]: #Name : Vishwajit S. Deshmukh  
#Roll No: 70  
#Sec : A
```

```
[ ]: #Aim : Creation of Array Using Numpy
```

```
[2]: import random
```

```
[13]: rm=[random.randint(1,50) for _ in range(15)]
```

rm

```
[14]: [49, 30, 43, 12, 37, 28, 13, 36, 9, 21, 39, 32,  
42, 41, 46]
```

```
[1]: import numpy as np  
  
arr = np.array([1, 2, 3, 4, 5])  
  
print(arr)  
  
print(type(arr))
```

```
[ 1 2 3 4 5 ]  
<class 'numpy.ndarray'>
```

```
[2]: # 1-D Array
```

```
[3]: arr = np.array([1, 2, 3, 4, 5])
```

```
[4]: arr
```

```
[4]: array([1, 2, 3, 4, 5])
```

```
[5]: # 2-D Array
```

[7]:

```
[6]: arr = np.array([[1, 2, 3], [4, 5, 6]])
```

arr

```
[7]: array([[1, 2, 3],  
          [4, 5, 6]])
```

```
[8]: a = np.array(42)  
b = np.array([1, 2, 3, 4, 5])  
c = np.array([[1, 2, 3], [4, 5, 6]])  
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])  
  
print(a.ndim)  
print(b.ndim)  
print(c.ndim)  
print(d.ndim)
```

0

1

2

3

[]:

Exp-6-DSS-Missing-value-treatment

October 13, 2024

1 Data Cleaning, Missing Value Treatment

```
[1]: # Experiment No. 6
```

```
[2]: # Name : Vishwajit S. Deshmukh  
# Roll : 70  
# s e c :
```

```
[1]: # Aim : To perform Data Cleaning And Missing Value Treatment
```

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

```
[1]: import os
```

```
[2]: os.getcwd()
```

```
[2]: 'C:\\\\Users\\Hp'
```

```
[7]: os.chdir("E:\\Downloads")
```

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

```
[9]: df
```

```
[9]:
```

	PassengerId	Survived	Pclass	\
0	892	0	3	
1	893	1	3	
2	894	0	2	
3	895	0	3	
4	896	1	3	
..	
413	1305	0	3	
414	1306	1	1	
415	1307	0	3	
416	1308	0	3	

```
417      1309  0      3
```

```
                                Name      Sex   Age SibSp Parch \
0                               Kelly, Mr. James      male 34.5  0      0
1                               Wilkes, Mrs. James (Ellen Needs) female 47.0
                                1      0
2                               Myles, Mr. Thomas Francis male 62.0  0      0
3                               Wirz, Mr. Albert      male 27.0  0      0
4                               Hirvonen, Mrs. Alexander (Helga E Lindqvist)
                                female 22.0      1      1
```

```
..
```

```
... ..
```

```
413      Spector, Mr. Woolf      male  NaN  0      0
414      Oliva y Ocana, Dona. Fermina female 39.0      0
                                0
415      Saether, Mr. Simon Sivertsen      male 38.5  0
                                0
416      Ware, Mr. Frederick      male  NaN  0      0
417      Peter, Master. Michael J      male  NaN  1      1
```

```
      Ticket      Fare Cabin Embarked
```

```
0      330911  7.8292      NaN  Q
1      363272  7.0000      NaN  S
2      240276  9.6875      NaN  Q
3      315154  8.6625      NaN  S
4      3101298 12.2875      NaN  S
..
                                ... ..
413      A.5. 3236  8.0500      NaN  S
414      PC 17758 108.9000 C105      C
415      SOTON/O.Q. 3101262      7.2500      NaN  S
416      359309      8.0500      NaN  S
417      2668  22.3583      NaN  C
```

```
418      rows x 12 columns]
```

```
[10]: df.head(15)
```

```
[10]: PassengerId Survived Pclass \
0      892      0      3
1      893      1      3
2      894      0      2
3      895      0      3
4      896      1      3
5      897      0      3
6      898      1      3
7      899      0      2
8      900      1      3
9      901      0      3
10     902      0      3
11     903      0      1
12     904      1      1
13     905      0      2
14     906      1      1
```

```

                                Name      Sex   Age SibSp \
0      Kelly, Mr. James  male  34.5   0
1      Wilkes, Mrs. James (Ellen Needs) female 47.0 1
2      Myles, Mr. Thomas Francis      male  62.0
   0
3      Wirz, Mr. Albert  male  27.0   0
4      Hirvonen, Mrs. Alexander (Helga E Lindqvist) female 22.0 1
5      Svensson, Mr. Johan Cervin      male  14.0
   0
6      Connolly, Miss. Kate  female  30.0   0
7      Caldwell, Mr. Albert Francis male  26.0
   1
8      Abraham, Mrs. Joseph (Sophie Halaut Easu) female 18.0 0
9      Davies, Mr. John Samuel      male  21.0
   2
10     Ilieff, Mr. Ylio  male   NaN   0
11     Jones, Mr. Charles Cresson      male  46.0
   0
```

```

12          Snyder, Mrs. John Pillsbury (Nelle
          Stevenson) female 23.0 1
13          Howard, Mr. Benjamin male 63.0 1
14          Chaffee, Mrs. Herbert Fuller (Carrie
          Constance... female 47.0 1

```

	Parch	Ticket	Fare	Cabin	Embarked
0	0	330911	7.8292	NaN	Q
1	0	363272	7.0000	NaN	S
2	0	240276	9.6875	NaN	Q
3	0	315154	8.6625	NaN	S
4	1	3101298	12.2875	NaN	S
5	0	7538	9.2250	NaN	S
6	0	330972	7.6292	NaN	Q
7	1	248738	29.0000	NaN	S
8	0	2657	7.2292	NaN	C
9	0	A/4 48871	24.1500	NaN	S
10	0	349220	7.8958	NaN	S
11	0	694	26.0000	NaN	S
12	0	21228	82.2667	B45	S
13	0	24065	26.0000	NaN	S
14	0	W.E.P. 5734	61.1750	E31	S

```
[11]: df.tail(15)
```

```

[11]: PassengerId Survived Pclass \
403      1295          0        1
404      1296          0        1
405      1297          0        2
406      1298          0        2
407      1299          0        1
408      1300          1        3
409      1301          1        3
410      1302          1        3
411      1303          1        1
412      1304          1        3
413      1305          0        3
414      1306          1        1
415      1307          0        3
416      1308          0        3
417      1309          0        3

```

	Name	Sex	Age	SibSp	\
403	Carrau, Mr. Jose Pedro	male	17.0	0	
404	Frauenthal, Mr. Isaac Gerald	male	43.0	1	

```

405      Nourney, Mr. Alfred (Baron von Drachstedt)"
      male 20.0  0
406      Ware, Mr. William Jeffery      male 23.0  1
407      Widener, Mr. George Dunton    male 50.0  1
408      Riordan, Miss. Johanna Hannah"" female NaN
      0
409      Peacock, Miss. Treasteall female  3.0  1
410      Naughton, Miss. Hannah female    NaN
      0
411      Minahan, Mrs. William Edward (Lillian E
      Thorpe) female 37.0 1
412      Henriksson, Miss. Jenny Lovisa female
      28.0  0
413      Spector, Mr. Woolf  male NaN  0
414      Oliva y Ocana, Dona. Fermina female 39.0
      0
415      Saether, Mr. Simon Sivertsen    male 38.5
      0
416      Ware, Mr. Frederick male NaN  0
417      Peter, Master. Michael J male NaN  1

```

```

      Parch      Ticket    Fare Cabin Embarked
403      0  113059 47.1000   NaN    S
404      0  17765 27.7208   D40    C
405      0  SC/PARIS 2166   13.8625   D38    C
406      0  28666 10.5000   NaN    S
407      1  113503 211.5000  C80    C
408      0  334915    7.7208   NaN    Q
409      1  SOTON/O.Q. 3101315    13.7750   NaN    S
410      0  365237    7.7500   NaN    Q
411      0  19928 90.0000   C78    Q
412      0  347086    7.7750   NaN    S
413      0  A.5. 3236  8.0500   NaN    S
414      0  PC 17758 108.9000  C105    C
415      0  SOTON/O.Q. 3101262    7.2500   NaN    S
416      0  359309    8.0500   NaN    S
417      1  2668 22.3583   NaN    C

```

```
[26]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Index: 87 entries, 12 to 414
Data columns (total 12
columns):
#   Column      Non-Null Count  Dtype
---  -

```



```

0 PassengerId 87 non-null int64
1 Survived    87 non-null int64
2 Pclass      87 non-null int64
3 Name        87 non-null object
4 Sex         87 non-null object
5 Age         87 non-null float64
6 SibSp       87 non-null int64
7 Parch       87 non-null int64
8 Ticket      87 non-null object
9 Fare        87 non-null float64
10 Cabin      87 non-null object
11 Embarked   87 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 8.8+ KB

```

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

```
[13]: (418, 12)
```

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

```
[14]: 5016
```

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

```
[15]: 2
```

```
[16]: df.isna()
```

```
[16]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	\
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
..	
413	False	False	False	False	False	True	False	False	False	
414	False	False	False	False	False	False	False	False	False	
415	False	False	False	False	False	False	False	False	False	
416	False	False	False	False	False	True	False	False	False	
417	False	False	False	False	False	True	False	False	False	

	Fare	Cabin	Embarked
0	False	True	False
1	False	True	False
2	False	True	False

```

3    False  True    False
4    False  True    False
..     ...   ...     ...
413 False  True    False
414 False False    False
415 False  True    False
416 False  True    False
417 False  True    False
[418 rows x 12 columns]

```

```
[17]: df.isna().any()
```

```

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

```

```
[18]: df.isna().sum()
```

```

[18]: PassengerId    0
Survived           0
Pclass             0
Name               0
Sex                0
Age                86
SibSp              0
Parch              0
Ticket             0
Fare                1
Cabin              327
Embarked           0
dtype: int64

```

```
[19]: df.describe()
```

```

[19]: PassengerId  Survived  Pclass  Age  SibSp  \  count  418.000000
418.000000  418.000000  332.000000  418.000000  mean  1100.500000
0.363636  2.265550  30.272590  0.447368

```

std	120.810458	0.481622	0.841838	14.181209	0.896760
min	892.000000	0.000000	1.000000	0.170000	0.000000
25%	996.250000	0.000000	1.000000	21.000000	0.000000
50%	1100.500000	0.000000	3.000000	27.000000	0.000000
75%	1204.750000	1.000000	3.000000	39.000000	1.000000
max	1309.000000	1.000000	3.000000	76.000000	8.000000

	Parch	Fare
count	418.000000	
417.000000	mean	0.392344
	35.627188	std
	0.981429	55.907576
min	0.000000	0.000000
25%	0.000000	7.895800
50%	0.000000	14.454200
75%	0.000000	31.500000
max	9.000000	512.329200

```
[20]: df["Age"].fillna(30.272590)
```

```
[20]: 0    34.50000
      1    47.00000
      2    62.00000
      3    27.00000
      4    22.00000
      ...
      413   30.27259
      414   39.00000
      415   38.50000
      416   30.27259
      417   30.27259
      Name: Age, Length: 418, dtype: float64
```

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

```
[21]: PassengerId    0
      Survived       0
      Pclass        0
      Name          0
      Sex           0
      Age           86
      SibSp         0
      Parch         0
      Ticket        0
      Fare          1
      Cabin        327
```

```
Embarked      0
dtype:
int64
```

```
[22]: df.any()
```

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

```
[23]: df = df.dropna()
```

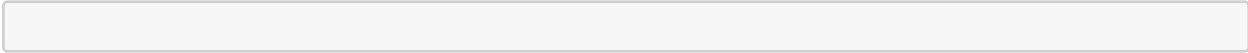
```
[24]: df.any()
```

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

```
[25]: df.isna().sum()
```

```
[25]: 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      [      ]:
```



Exp-7-DSS-Data-Visualisation

October 13, 2024

1 Data Visualisation

[5]:

```
[1]: # Name : Visshwajt S. Deshmukh  
# Roll : 70  
# s e c :
```

```
[2]: # Aim : Perform opertainon on Data Visualisation using matplotlib
```

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

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

x

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

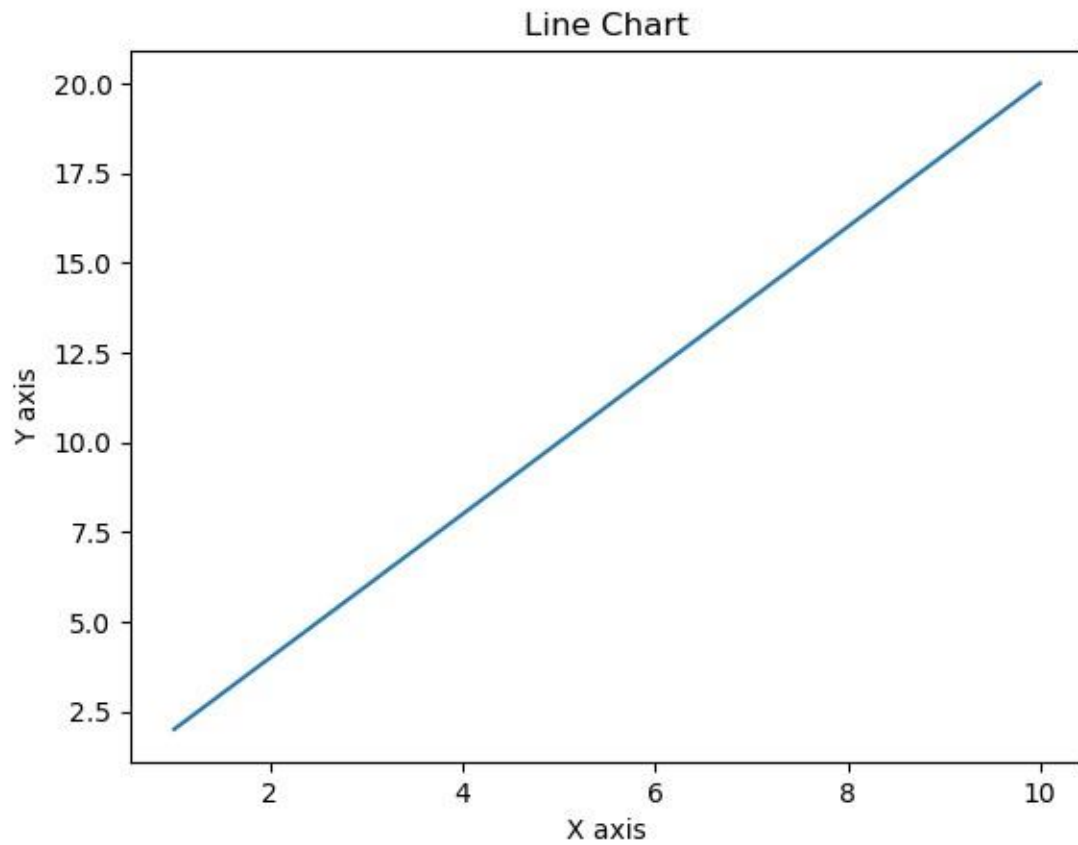
[7]:

```
[6]: y = 2*x
```

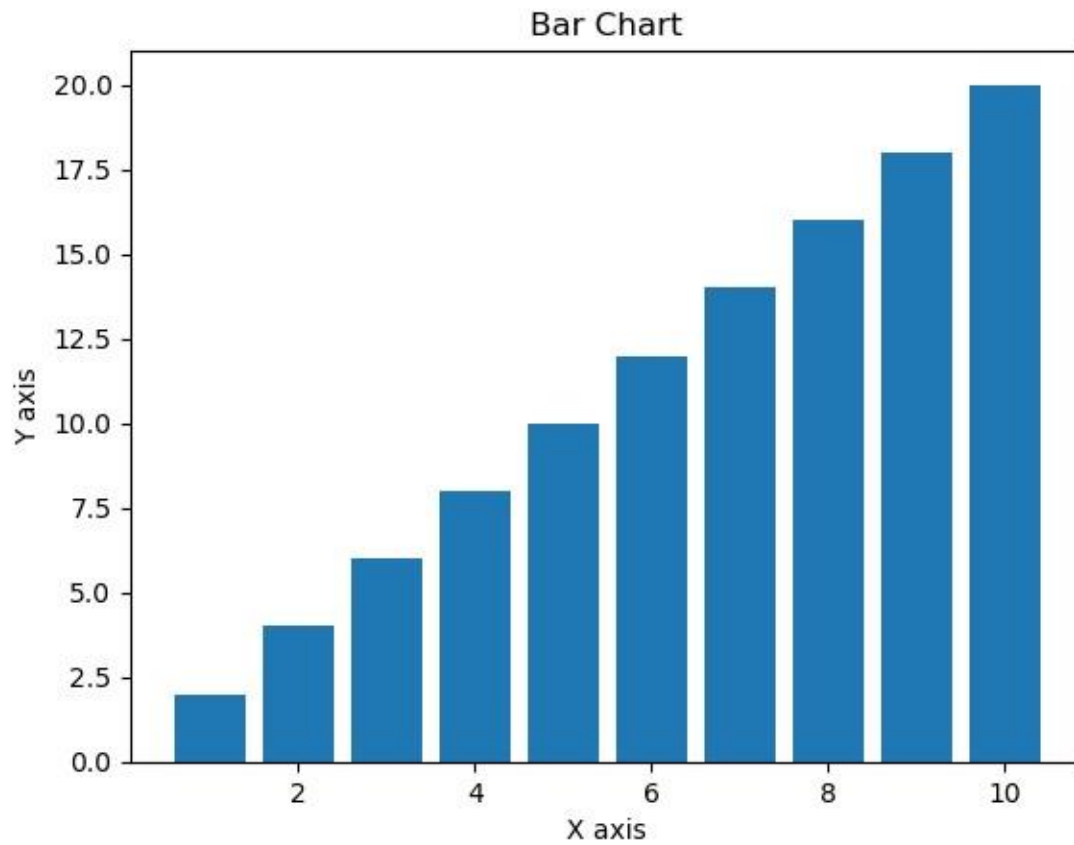
y

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

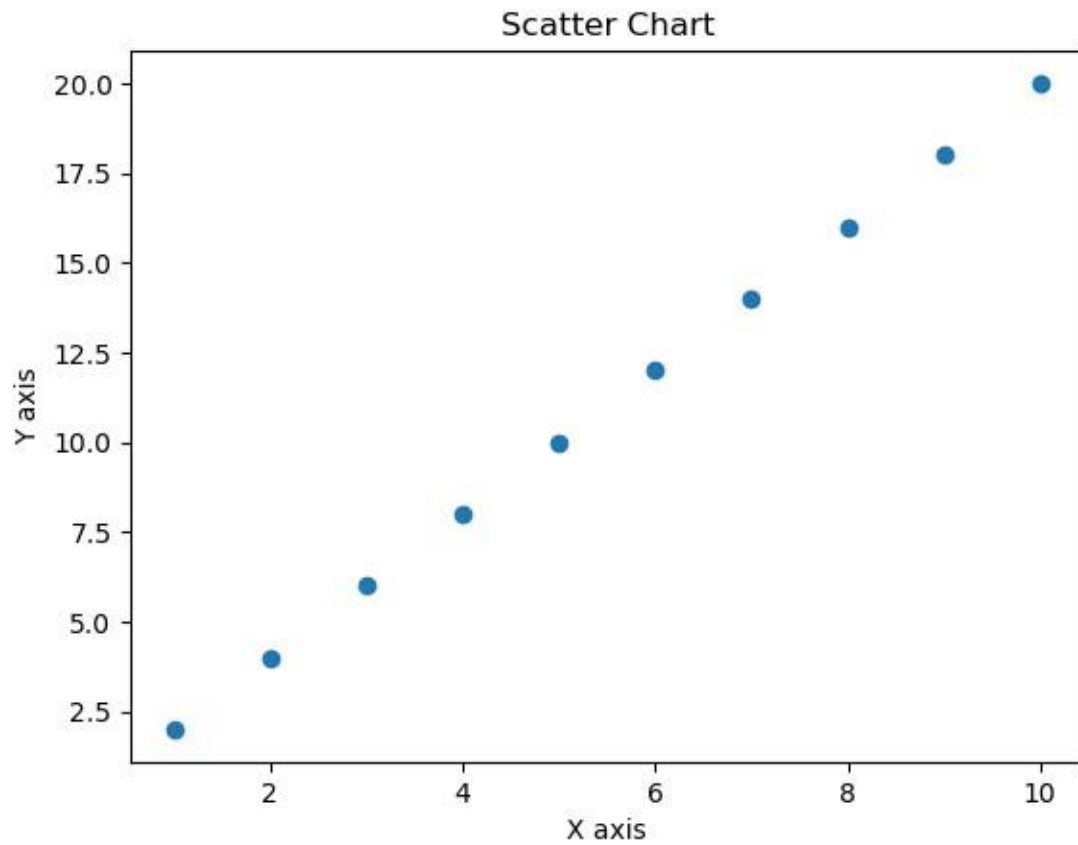
```
[8]: plt.plot(x,y)  
plt.title("Line Chart")  
plt.xlabel("X axis")  
plt.ylabel("Y axis")  
plt.show()
```



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



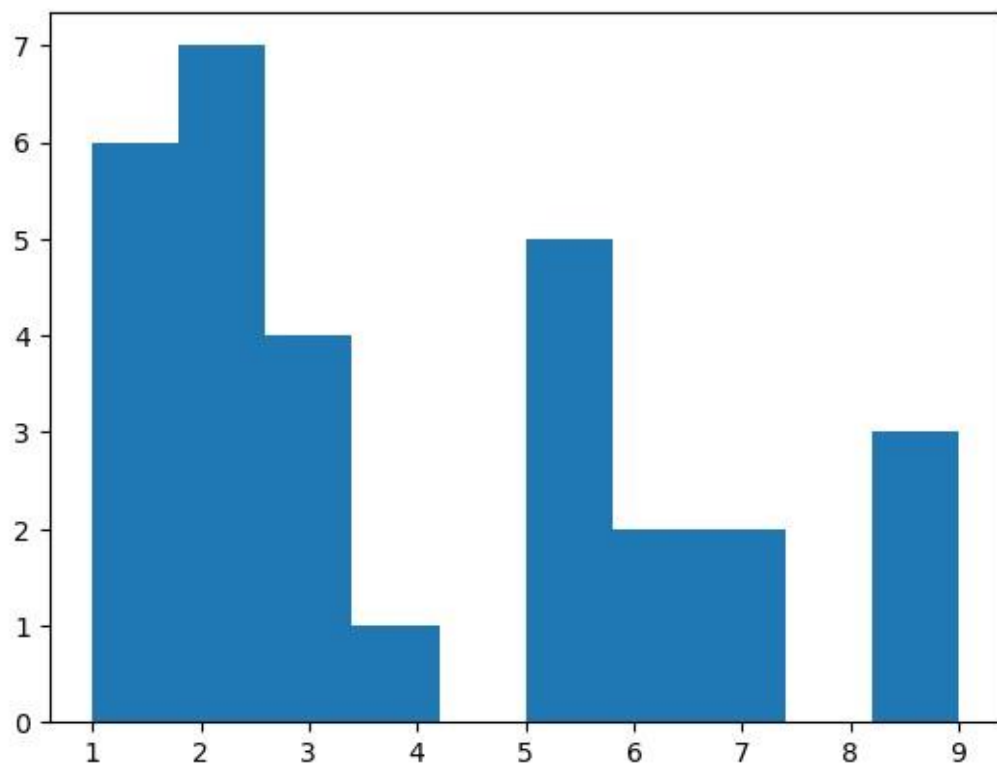
```
[10]: plt.scatter(x,y)
plt.title("Scatter Chart ")
plt.xlabel("X axis")
plt.ylabel("Y axis")
plt.show()
```

2 Hisogram

```
[11]: H = [1,2,1,3,2,1,5,7,9,5,4,2,1,3,5,6,3,2,3,5,9,7,6,9,2,5,2,2,1,1]
```

```
[12]: plt.hist(H)  
plt.show()
```



[]:

8-DSS-Central-Tendency-of-measures

October 13, 2024

1 Central Tendency of Measures

[5]:

```
[1]: # Experiment No. 8
```

```
[2]: # Name : Vishwajit S. Deshmukh  
# Roll : 70  
# s e c :
```

```
[3]: # Aim : To perform Central Tendency of Measures
```

```
[4]: age =[21,22,22,23,25,19,20,22,21,22,18,24,19,23,24,23,22,22,22]
```

age

```
[5]: [21, 22, 22, 23, 25, 19, 20, 22, 21, 22, 18, 24, 19, 23, 24, 23,  
22, 22, 22]
```

```
[7]: import statistics as st
```

```
[8]: Mean = st.mean(age)
```

```
[10]: Mean
```

```
[10]: 21.789473684210527
```

```
[11]: Median = st.median(age)
```

```
[12]: Median
```

```
[12]: 22
```

```
[14]:
```

```
[13]: Mode = st.mode(age)
```

Mode

```
[14]: 22 [ ]:
```

Exp-9-DSS-Simple-Linear-Regreation

October 13, 2024

1 Simple Linear Regreation

```
[1]: # Name :Vishwajit S. Deshmukh
      # Roll : 70
      # s e c :
```

```
[ ]: #Aim : To perform Simple Linear Regreation on Data
```

```
[1]: import pandas as pd
      import os
```

```
[2]: os.getcwd()
```

```
[2]: 'C:\\\\Users\\Hp'
```

```
[5]: path = os.chdir('E://Downloads')
```

```
[6]: data = pd.read_csv('salary.csv')
```

```
[7]: data.head()
```

```
[7]:   YearsExperience  Salary
0              1.1   39343
1              1.3   46205
2              1.5   37731
3              2.0   43525
4              2.2   39891
```

```
[8]: data.tail()
```

```
[8]:   YearsExperience  Salary
30              11.2  127345
31              11.5  126756
32              12.3  128765
33              12.9  135675
34              13.5  139465
```

```
[9]: data.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: 692.0 bytes

```

```
[10]: data.describe()
```

```

[10]:  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

```

```
[11]: data.shape
```

```
[11]: (35, 2)
```

```
[12]: data.size
```

```
[12]: 70
```

```
[13]: data.ndim
```

```
[13]: 2
```

```
[14]: data.isnull
```

```

[14]: <bound method DataFrame.isnull of 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>

```

```
[16]:
```

```
[15]: a = "Ashish"
```

```
a
```

```
[16]: 'Ashish'
```

```
[17]:
```

```
a[0]
```

```
[17]: 'A'
```

```
[18]: a[-1]
```

```
[18]: 'h'
```

```
[19]: a[1:3]
```

```
[19]: 'sh'
```

```
[20]: a[1:4]
```

```
[20]: 'shi'
```

```
[63]: x = data.iloc[:, :-1].values
      y = data.iloc[:, -1].values
```

```
[64]: 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]]
```

```
[65]: 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]
```

```
[66]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

```
[67]: 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)
```

```
[68]: print(x_test)
```

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

```
[69]: 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]]
```

```
[70]: print(y_test)
```

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

```
[71]: 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]
```

```
[72]: from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train, y_train)
```

```
[72]: LinearRegression()
```

```
[73]: m = lr.coef_
```

```
[74]: print("Coefficient :", m)
```

```
Coefficient : [8555.33918938]
```

```
[75]: c = lr.intercept_
```

```
[76]: print("Intercept :", c)
```

```
Intercept : 29602.07353482097
```

```
[77]: lr.score(x_test, y_test) * 100
```

```
[77]: 91.71426108885095
```

```
[ ]:
```

Exp-10-DSS-Logistic-Regression-mode

October 13, 2024

```
[ ]: # Practical no: 10
```

```
[1]: # Name : Vishwajit S. Deshmukh  
# Roll : 70  
# s e c :
```

```
[1]: # Aim : To Perform Logistic Regression on Data
```

```
[2]: 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')
```

```
[1]: import os
```

```
[2]: os.getcwd()
```

```
[2]: 'C:\\Users\\Hp'
```

```
[5]: path = os.chdir('E://Downloads')
```

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

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

```
[7]: male age education currentSmoker cigsPerDay BPMeds prevalentStroke \  
0 1 39 4.0 0 0.0 0.0 0  
1 0 46 2.0 0 0.0 0.0 0  
2 1 48 1.0 1 20.0 0.0 0  
3 0 61 3.0 1 30.0 0.0 0  
4 0 46 3.0 1 23.0 0.0 0  
  
prevalentHyp diabetes totChol sysBP diaBP BMI heartRate glucose \  
0 0 0 195.0 106.0 70.0 26.97 80.0 77.0  
1 0 0 250.0 121.0 81.0 28.73 95.0 76.0  
2 0 0 245.0 127.5 80.0 25.34 75.0 70.0
```

3	1	0	225.0	150.0	95.0	28.58	65.0	103.0
4	0	0	285.0	130.0	84.0	23.10	85.0	85.0

TenYearCHD	
0	0
1	0
2	0
3	1
4	0

```
[8]: df.describe()
```

```
[8]:
```

	male	age	education	currentSmoker	cigsPerDay \
count	4238.000000	4238.000000		4238.000000	
	4133.000000			4209.000000	
mean	0.429212	49.584946	1.978950	0.494101	9.003089
std	0.495022	8.572160	1.019791	0.500024	11.920094
min	0.000000	32.000000	1.000000	0.000000	0.000000
25%	0.000000	42.000000	1.000000	0.000000	0.000000
50%	0.000000	49.000000	2.000000	0.000000	0.000000
75%	1.000000	56.000000	3.000000	1.000000	20.000000
max	1.000000	70.000000	4.000000	1.000000	70.000000

	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol \
count	4185.000000	4238.000000	4238.000000	4238.000000	
		4188.000000			
mean	0.029630	0.005899	0.310524	0.025720	236.721585
std	0.169584	0.076587	0.462763	0.158316	44.590334
min	0.000000	0.000000	0.000000	0.000000	107.000000
25%	0.000000	0.000000	0.000000	0.000000	206.000000
50%	0.000000	0.000000	0.000000	0.000000	234.000000
75%	0.000000	0.000000	1.000000	0.000000	263.000000
max	1.000000	1.000000	1.000000	1.000000	696.000000

	sysBP	diaBP	BMI	heartRate	glucose \
count	4238.000000	4238.000000	4219.000000	4237.000000	3850.000000
mean	132.352407	82.893464	25.802008	75.878924	81.966753
std	22.038097	11.910850	4.080111	12.026596	23.959998
min	83.500000	48.000000	15.540000	44.000000	40.000000
25%	117.000000	75.000000	23.070000	68.000000	71.000000
50%	128.000000	82.000000	25.400000	75.000000	78.000000
75%	144.000000	89.875000	28.040000	83.000000	87.000000
max	295.000000	142.500000	56.800000	143.000000	394.000000

TenYearCHD	
count	4238.000000
mean	0.151958
std	0.359023

```

min      0.000000
25%      0.000000
50%      0.000000
75%      0.000000
max      1.000000

```

```
[9]: 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

```

```
[10]: df.isna().sum()
```

```

[10]: male                0
      age                 0
      education           105
      currentSmoker        0
      cigsPerDay           29
      BPMeds              53
      prevalentStroke      0
      prevalentHyp         0
      diabetes             0
      totChol              50
      sysBP                0
      diaBP                0

```

[11] :

df

```
[11]: male age education currentSmoker cigsPerDay BPMeds \
```

0		1	39	4.0	0	0.0	0.0						
1		0	46	2.0	0	0.0	0.0	2	1		48	1.0	1
			20.0	0.0									
3		0	61	3.0	1	30.0	0.0						
4		0	46	3.0	1	23.0	0.0						
...											
4233	1		50	1.0	1	1.0	0.0	4234		1		51	
3.0	1		43.0	0.0									
4235	0		48	2.0	1	20.0	NaN						
4236	0		44	1.0	1	15.0	0.0						
4237	0		52	2.0	0	0.0	0.0						

	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI \
0	0	0	0	195.0	106.0	70.0	26.97
1	0	0	0	250.0	121.0	81.0	28.73
2	0	0	0	245.0	127.5	80.0	25.34
3	0	1	0	225.0	150.0	95.0	28.58
4	0	0	0	285.0	130.0	84.0	23.10
...
4233	0	1	0	313.0	179.0	92.0	25.97
4234	0	0	0	207.0	126.5	80.0	19.71
4235	0	0	0	248.0	131.0	72.0	22.00
4236	0	0	0	210.0	126.5	87.0	19.16
4237	0	0	0	269.0	133.5	83.0	21.47

	heartRate	glucose	TenYearCHD
0	80.0	77.0	0
1	95.0	76.0	0
2	75.0	70.0	0
3	65.0	103.0	1
4	85.0	85.0	0
...
4233	66.0	86.0	1

```

4234      65.0    68.0  0
4235      84.0    86.0  0
4236      86.0    NaN   0
4237      80.0   107.0  0
[4238 rows x 16 columns]

```

1 Missing Value Tretment

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

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

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

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

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

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

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

```
[19]: df.isna().sum()
```

```

[19]: 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
```

```
[20]: x = df.drop("TenYearCHD",axis=1)
      y = df['TenYearCHD']
```

```
[21]: x
```

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

[4238 rows x 15 columns]

2 Train Test Split

```
[22]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, _  
      ↪ random_state =42)
```

```
[23]: y_train
```

```
[23]: 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
```

3 Logistic Regressionon Algo

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

```
[24]: 0.8486725663716814
```

```
[  
]:
```

```
[  
]:
```


4 Decision Tree Algo

```
[31]: from sklearn.tree import DecisionTreeClassifier
      dtc = DecisionTreeClassifier()
      dtc.fit(x_train, y_train)
      model.score(x_train, y_train)
      cc = dtc.score(x_test, y_test) * 100
      print(acc)
```

85.37735849056604

5 Random Forest Classifier

```
[32]: from sklearn.ensemble import RandomForestClassifier classifier =
      RandomForestClassifier(n_estimators = 10, criterion = 'entropy',
      ↪random_state = 0)
      classifier.fit(x_test, y_test) acc =
      classifier.score(x_test, y_test) * 100
      print(acc)
```

98.58490566037736

```
[ ]:
```

K Nearest Nabors

```
# Practical no: 11

# Name : Vishwajit S. Deshmukh
# Roll : 70
# sec : A # Aim : To Perform

KNN on data

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')
import os
os.getcwd()
'C:\\Users\\Hp'
path = os.chdir('E://Downloads')

df=pd.read_csv("framingham.csv")

df.head()
male age education currentSmoker cigsPerDay BPMeds
prevalentStroke \
1 39 4.0 0 0.0 0.0 0
0 46 2.0 0 0.0 0.0 1
1 48 1.0 1 20.0 0.0 2
0 61 3.0 1 30.0 0.0 3
0 46 3.0 1 23.0 0.0 4

prevalentHyp diabetes totChol sysBP diaBP BMI heartRate
glucose \
0 0 0 195.0 106.0 70.0 26.97 80.0
77.0
1 0 0 250.0 121.0 81.0 28.73 95.0
76.0
2 0 0 245.0 127.5 80.0 25.34 75.0
```

```

3          1          0      225.0  150.0   95.0  28.58      65.0
103
4          0          0      285.0  130.0   84.0  23.10      85.0
85
    TenYearCHD    0
1
2
3

```

```
4          0
```

```
df.describe()
```

```

          male          age  education  currentSmoker
cigsPerDay \
count  4238.000000  4238.000000  4133.000000    4238.000000
4209.000000
mean      0.429212    49.584946    1.978950    0.494101
9.003089
std      0.495022    8.572160    1.019791    0.500024
11.920094
min      0.000000    32.000000    1.000000    0.000000
0.000000
25%      0.000000    42.000000    1.000000    0.000000
0.000000
50%      0.000000    49.000000    2.000000    0.000000
0.000000
75%      1.000000    56.000000    3.000000    1.000000
20.000000
max      1.000000    70.000000    4.000000    1.000000
70.000000

```

```

          BPMeds  prevalentStroke  prevalentHyp    diabetes
totChol \
count  4185.000000    4238.000000  4238.000000  4238.000000
4188.000000
mean      0.029630          0.005899    0.310524    0.025720
236.721585
std      0.169584          0.076587    0.462763    0.158316
44.590334
min      0.000000          0.000000    0.000000    0.000000
107.000000
25%      0.000000          0.000000    0.000000    0.000000
206.000000
50%      0.000000          0.000000    0.000000    0.000000
234.000000
75%      0.000000          0.000000    1.000000    0.000000
263.000000

```

```
max      1.000000      1.000000      1.000000      1.000000
696.000000
```

```
      sysBP      diaBP      BMI      heartRate      glucose
count  4238.000000  4238.000000  4219.000000  4237.000000  3850.000000
```

```
\
mean      132.352407      82.893464      25.802008      75.878924      81.966753
std        22.038097      11.910850       4.080111      12.026596      23.959998
min        83.500000      48.000000      15.540000      44.000000      40.000000
25%        117.000000      75.000000      23.070000      68.000000      71.000000
50%        128.000000      82.000000      25.400000      75.000000      78.000000
75%        144.000000      89.875000      28.040000      83.000000      87.000000
max        295.000000     142.500000      56.800000     143.000000     394.000000
```

```
      TenYearCHD
count  4238.000000
mean      0.151958
std        0.359023
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         1.000000
```

```
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
```

```
df.isna().sum()
```

```
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
```

Missing Value Treatment

```
df['glucose'].fillna(value = df['glucose'].mean(), inplace=True)
df['education'].fillna(value = df['education'].mean(), inplace=True)
df['heartRate'].fillna(value = df['heartRate'].mean(), inplace=True)
df['BMI'].fillna(value = df['BMI'].mean(), inplace=True)
df['cigsPerDay'].fillna(value = df['cigsPerDay'].mean(), inplace=True)
df['totChol'].fillna(value = df['totChol'].mean(), inplace=True)
df['BPMeds'].fillna(value = df['BPMeds'].mean(), inplace=True)
df.isna().sum()
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

```

```

x = df.drop("TenYearCHD",axis=1)

```

```

y = df['TenYearCHD'] x

```

```

      male  age  education  currentSmoker  cigsPerDay  BPMeds
\
0      1   39      4.0          0          0.0  0.00000
1      0   46      2.0          0          0.0  0.00000
2      1   48      1.0          1         20.0  0.00000
3      0   61      3.0          1         30.0  0.00000
4      0   46      3.0          1         23.0  0.00000...
...  ...      ...      ...      ...      ...      ...
4233   1   50      1.0          1          1.0  0.00000
4234   1   51      3.0          1         43.0  0.00000
4235   0   48      2.0          1         20.0  0.02963
4236   0   44      1.0          1         15.0  0.00000
4237   0   52      2.0          0          0.0  0.00000

```

```

      prevalentStroke  prevalentHyp  diabetes  totChol  sysBP
diaBP
BMI \
0      0      0      0      195.0  106.0
70.0
26.97
1      0      0      0      250.0  121.0
81.0
28.73
2      0      0      0      245.0  127.5
80.0
25.34
3      0      1      0      225.0  150.0
95.0
28.58
4      0      0      0      285.0  130.0
84.0
23.10

```


...
...					
..					
4233	0	1	0	313.0	179.0
92.0					
25.97					
4234	0	0	0	207.0	126.5
80.0					
19.71					
4235	0	0	0	248.0	131.0
72.0					

```

22.00
4236          0          0          0    210.0  126.5   87.0
19.16
4237          0          0          0    269.0  133.5   83.0
21.47

      heartRate      glucose  0
80.0    77.000000
1       95.0    76.000000
2       75.0    70.000000
3       65.0   103.000000
4       85.0    85.000000  ...      ...      ...
4233     66.0    86.000000
4234     65.0    68.000000
4235     84.0    86.000000
4236     86.0    81.966753
4237     80.0   107.000000

[4238 rows x 15 columns]

```

Train Test Split

```

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
random_state =42) y_train
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

```

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
```

Exp-12-DSS-SVM

October 13, 2024

1 SVM Classifier

```
[1]: # Practical no: 12
```

```
[2]: # Name :Vishwajit S. Deshmukh  
# Roll : 70  
# s e c :
```

```
[3]: # Aim : To Perform SVM on Data
```

```
[4]: 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')
```

```
[1]: import os
```

```
[2]: os.getcwd()
```

```
[2]: 'C:\\\\Users\\Hp'
```

```
[7]: path = os.chdir('E://Downloads')
```

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

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

```
[9]: male age education currentSmoker cigsPerDay BPMeds prevalentStroke \  
0 1 39 4.0 0 0.0 0.0 0  
1 0 46 2.0 0 0.0 0.0 0  
2 1 48 1.0 1 20.0 0.0 0  
3 0 61 3.0 1 30.0 0.0 0  
4 0 46 3.0 1 23.0 0.0 0 prevalentHyp diabetes  
totChol sysBP diaBP BMI heartRate glucose \  
0 0 0 195.0 106.0 70.0 26.97 80.0 77.0  
1 0 0 250.0 121.0 81.0 28.73 95.0 76.0  
2 0 0 245.0 127.5 80.0 25.34 75.0 70.0
```

```

3          1    0    225.0 150.0      95.0 28.58 65.0 103.0
4          0    0    285.0 130.0      84.0 23.10 85.0 85.0

```

```

TenYearCHD
0          0
1          0
2          0
3          1
4          0

```

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

```

[10]:          male          age  education  currentSmoker  cigsPerDay \
count  4238.000000  4238.000000          4238.000000
4133.000000          4209.000000
mean      0.429212   49.584946    1.978950      0.494101    9.003089
std       0.495022    8.572160    1.019791    0.500024   11.920094
min       0.000000   32.000000    1.000000    0.000000    0.000000
25%       0.000000   42.000000    1.000000    0.000000    0.000000
50%       0.000000   49.000000    2.000000    0.000000    0.000000
75%       1.000000   56.000000    3.000000    1.000000   20.000000
max       1.000000   70.000000    4.000000    1.000000   70.000000

          BPMeds  prevalentStroke  prevalentHyp  diabetes  totChol \
count  4185.000000    4238.000000    4238.000000    4238.000000
4188.000000
mean      0.029630      0.005899    0.310524    0.025720  236.721585
std      0.169584      0.076587    0.462763    0.158316  44.590334
min      0.000000      0.000000    0.000000    0.000000  107.000000
25%      0.000000      0.000000    0.000000    0.000000  206.000000
50%      0.000000      0.000000    0.000000    0.000000  234.000000
75%      0.000000      0.000000    1.000000    0.000000  263.000000
max      1.000000      1.000000    1.000000    1.000000  696.000000

          sysBP          diaBP          BMI  heartRate  glucose \
count  4238.000000  4238.000000  4219.000000  4237.000000  3850.000000
mean   132.352407   82.893464   25.802008   75.878924   81.966753
std    22.038097   11.910850    4.080111   12.026596   23.959998
min     83.500000   48.000000   15.540000   44.000000   40.000000
25%    117.000000   75.000000   23.070000   68.000000   71.000000
50%    128.000000   82.000000   25.400000   75.000000   78.000000
75%    144.000000   89.875000   28.040000   83.000000   87.000000
max    295.000000  142.500000   56.800000  143.000000  394.000000

          TenYearCHD
count  4238.000000
mean    0.151958
std     0.359023

```

```

min      0.000000
25%      0.000000
50%      0.000000
75%      0.000000
max      1.000000

```

```
[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

```

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

```

[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

```

2 Missing Value Tretment

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

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

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

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

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

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

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

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

```

[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

```

```

[21]: x = df.drop("TenYearCHD",axis=1)
      y = df['TenYearCHD']

```

```
[22]:
```

```

[22]:   male age education currentSmoker cigsPerDay BPMeds \
      0      1      39      4.0      0      0.0 0.00000

```

```

1          0  46  2.0  0  0.0 0.00000 2  1  48
          1.0  1  20.0 0.00000
3      0  61  3.0  1  30.0 0.00000
4      0  46  3.0  1  23.0 0.00000
...
4233  1  50  1.0  1  1.0 0.00000 4234 1  51
3.0  1  43.0 0.00000
4235  0  48  2.0  1  20.0 0.02963
4236  0  44  1.0  1  15.0 0.00000
4237  0  52  2.0  0  0.0 0.00000

```

```

prevalentStroke prevalentHyp diabetes totChol sysBP diaBP BMI \
0          0  0  0  195.0 106.0  70.0 26.97
1          0  0  0  250.0 121.0  81.0 28.73
2          0  0  0  245.0 127.5  80.0 25.34
3          0  1  0  225.0 150.0  95.0 28.58
4          0  0  0  285.0 130.0  84.0 23.10
...
4233      0  1  0  313.0 179.0  92.0 25.97
4234      0  0  0  207.0 126.5  80.0 19.71
4235      0  0  0  248.0 131.0  72.0 22.00
4236      0  0  0  210.0 126.5  87.0 19.16
4237      0  0  0  269.0 133.5  83.0 21.47

```

```

heartRate  glucose
0      80.0  77.000000
1      95.0  76.000000
2      75.0  70.000000
3      65.0 103.000000
4      85.0  85.000000
...
4233     66.0  86.000000
4234     65.0  68.000000
4235     84.0  86.000000
4236     86.0  81.966753
4237     80.0 107.000000

```

[4238 rows x 15 columns]

3 Train Test Split

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

```
[25]: y_train
```

```
[25]: 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
```

4 SVM Classifier

```
[26]: 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
```

```
[ ]:
```