In [ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:

```
df = pd.read_csv('Iris.csv')
df.head()
```

Out[ ]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [ ]:

```
df.drop(columns = ['Id'], axis = 1, inplace = True)
```

In [ ]:

```
df['Species'].value_counts()
```

Out[ ]:

```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

In [ ]:

```
#Categorizing data species wise
setosa_df = df[df['Species'] == 'Iris-setosa']
versicolor_df = df[df['Species'] == 'Iris-versicolor']
virginica_df = df[df['Species'] == 'Iris-virginica']
```

In [ ]:

```
#Iris - setosa
setosa_df.head()
```

Out[ ]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [ ]:

```
#Iris - setosa statistical description
setosa_df.describe()
```

Out[ ]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.000000	50.000000	50.00000
mean	5.00600	3.418000	1.464000	0.24400
std	0.35249	0.381024	0.173511	0.10721
min	4.30000	2.300000	1.000000	0.10000
25%	4.80000	3.125000	1.400000	0.20000
50%	5.00000	3.400000	1.500000	0.20000
75%	5.20000	3.675000	1.575000	0.30000
max	5.80000	4.400000	1.900000	0.60000

In [ ]:

```
#Iris - versicolor
versicolor_df.head()
```

Out[ ]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
50	7.0	3.2	4.7	1.4	Iris-versicolor
51	6.4	3.2	4.5	1.5	Iris-versicolor
52	6.9	3.1	4.9	1.5	Iris-versicolor
53	5.5	2.3	4.0	1.3	Iris-versicolor
54	6.5	2.8	4.6	1.5	Iris-versicolor

In [ ]:

```
#Iris - versicolor statistical description
versicolor_df.describe()
```

Out[ ]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.000000	50.000000	50.000000	50.000000
mean	5.936000	2.770000	4.260000	1.326000
std	0.516171	0.313798	0.469911	0.197753
min	4.900000	2.000000	3.000000	1.000000
25%	5.600000	2.525000	4.000000	1.200000
50%	5.900000	2.800000	4.350000	1.300000
75%	6.300000	3.000000	4.600000	1.500000
max	7.000000	3.400000	5.100000	1.800000

In [ ]:

```
#Iris - virginica
virginica_df.head()
```

Out[ ]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
100	6.3	3.3	6.0	2.5	Iris-virginica
101	5.8	2.7	5.1	1.9	Iris-virginica
102	7.1	3.0	5.9	2.1	Iris-virginica
103	6.3	2.9	5.6	1.8	Iris-virginica
104	6.5	3.0	5.8	2.2	Iris-virginica

In [ ]:

```
#Iris - virginica statistical description
virginica_df.describe()
```

Out[ ]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.000000	50.000000	50.00000
mean	6.58800	2.974000	5.552000	2.02600
std	0.63588	0.322497	0.551895	0.27465
min	4.90000	2.200000	4.500000	1.40000
25%	6.22500	2.800000	5.100000	1.80000
50%	6.50000	3.000000	5.550000	2.00000
75%	6.90000	3.175000	5.875000	2.30000
max	7.90000	3.800000	6.900000	2.50000

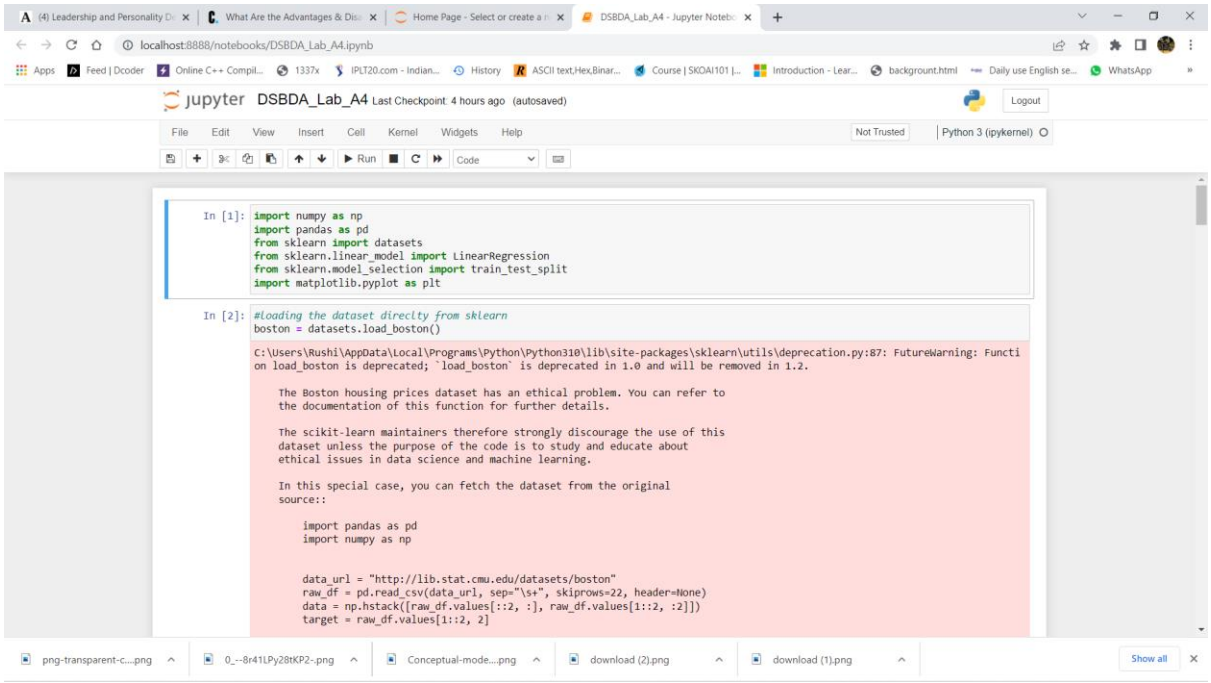
In [ ]:

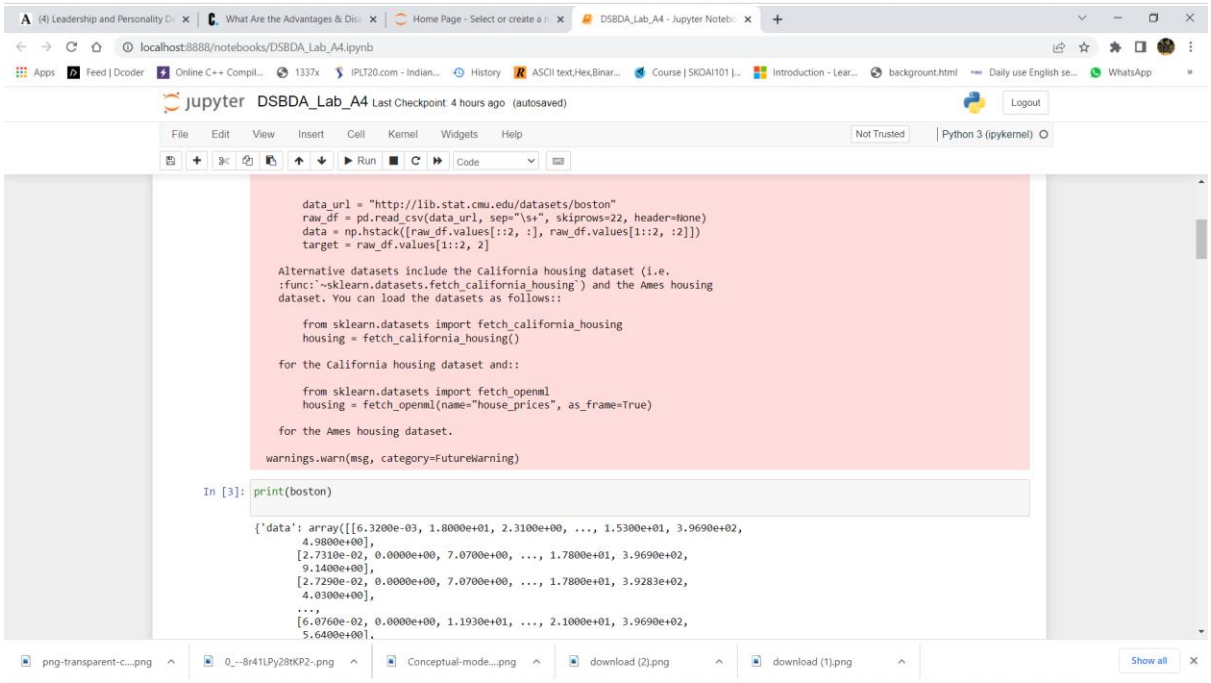
```

```

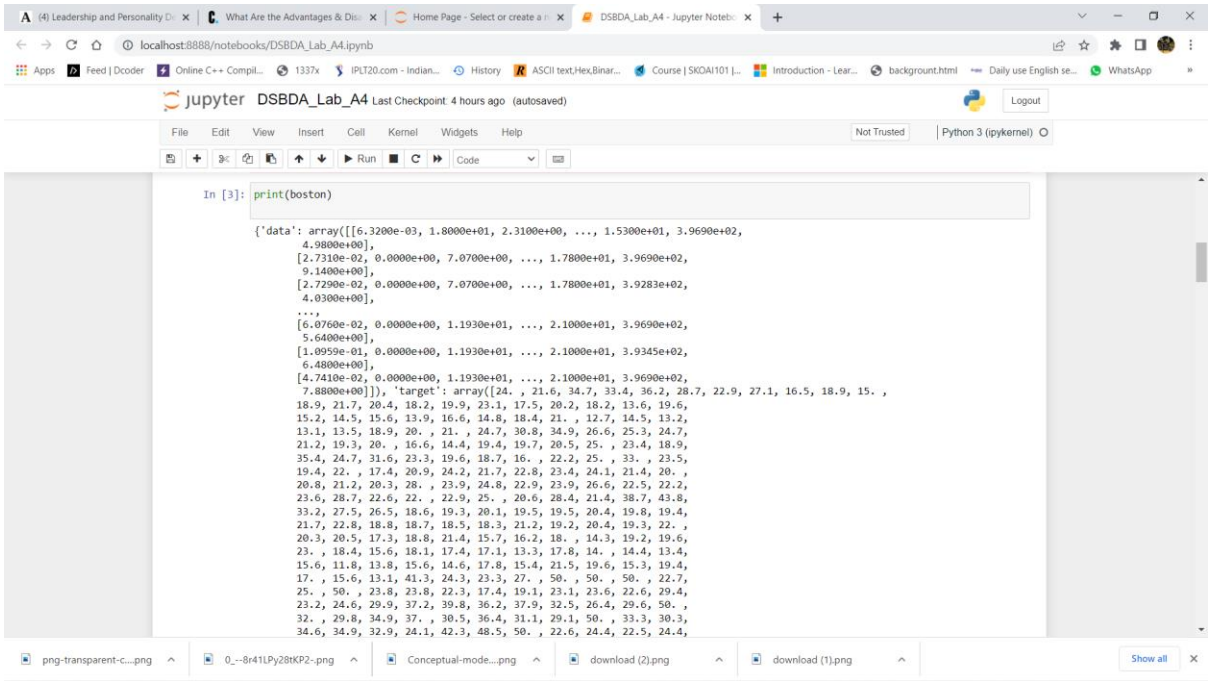
Data Science and Big Data Analytics

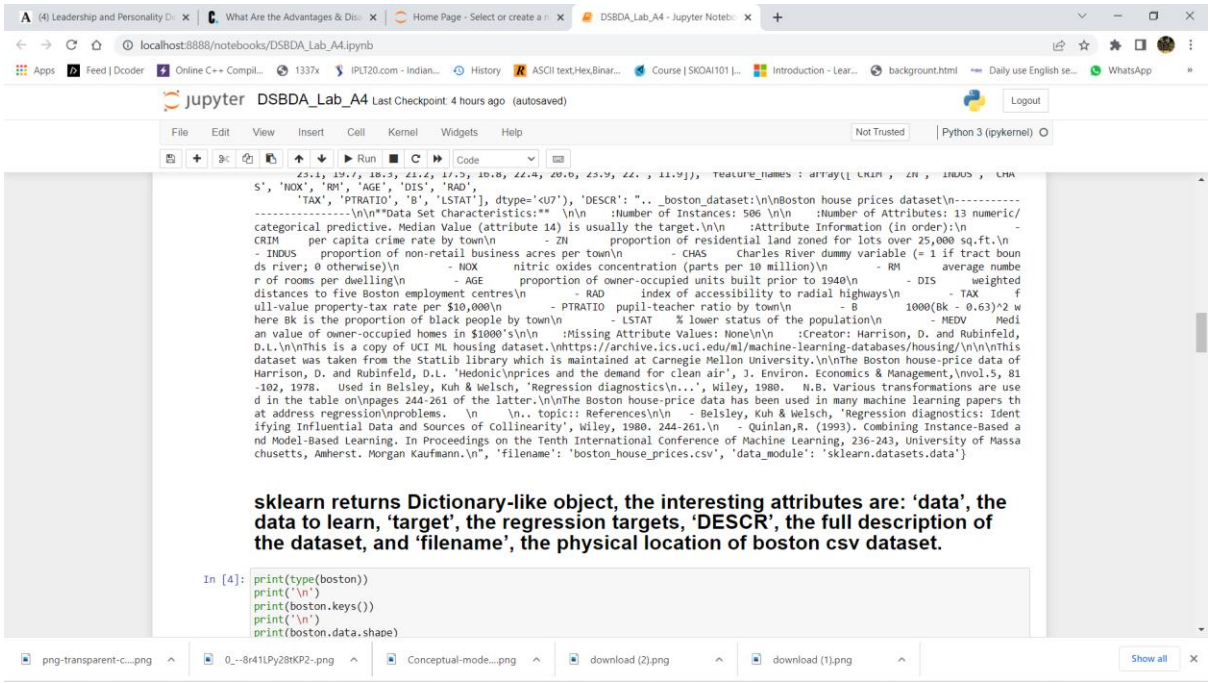
Group : A

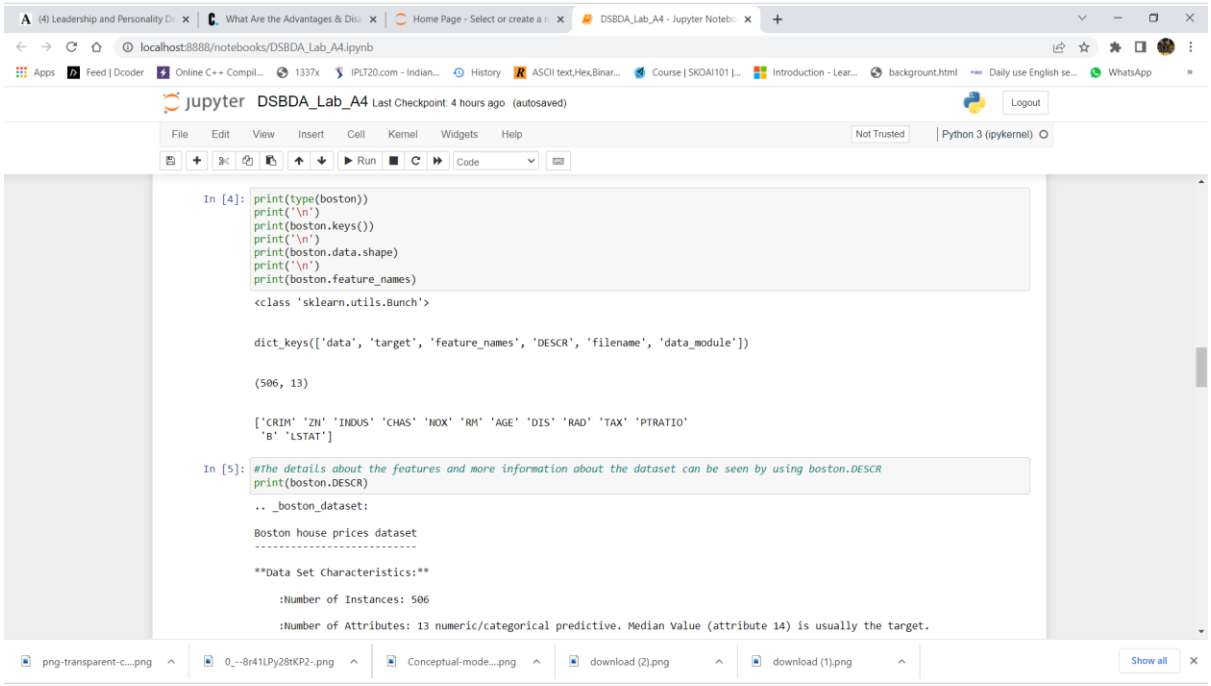
Lab Assignment : 04



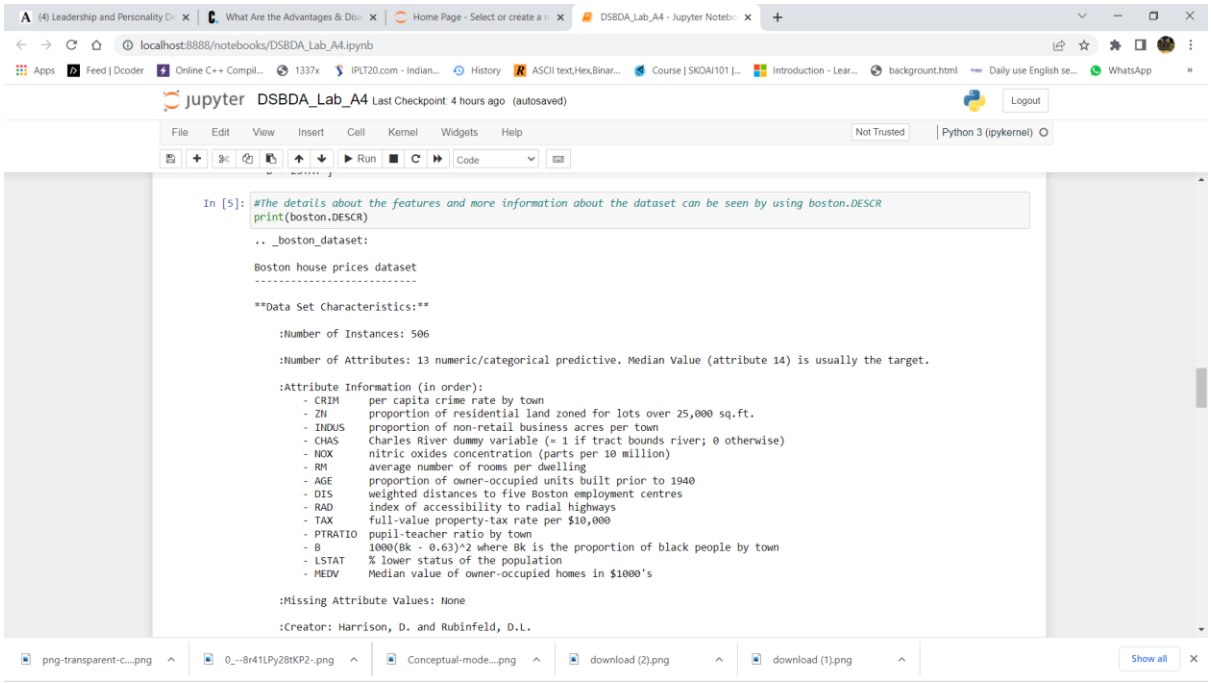


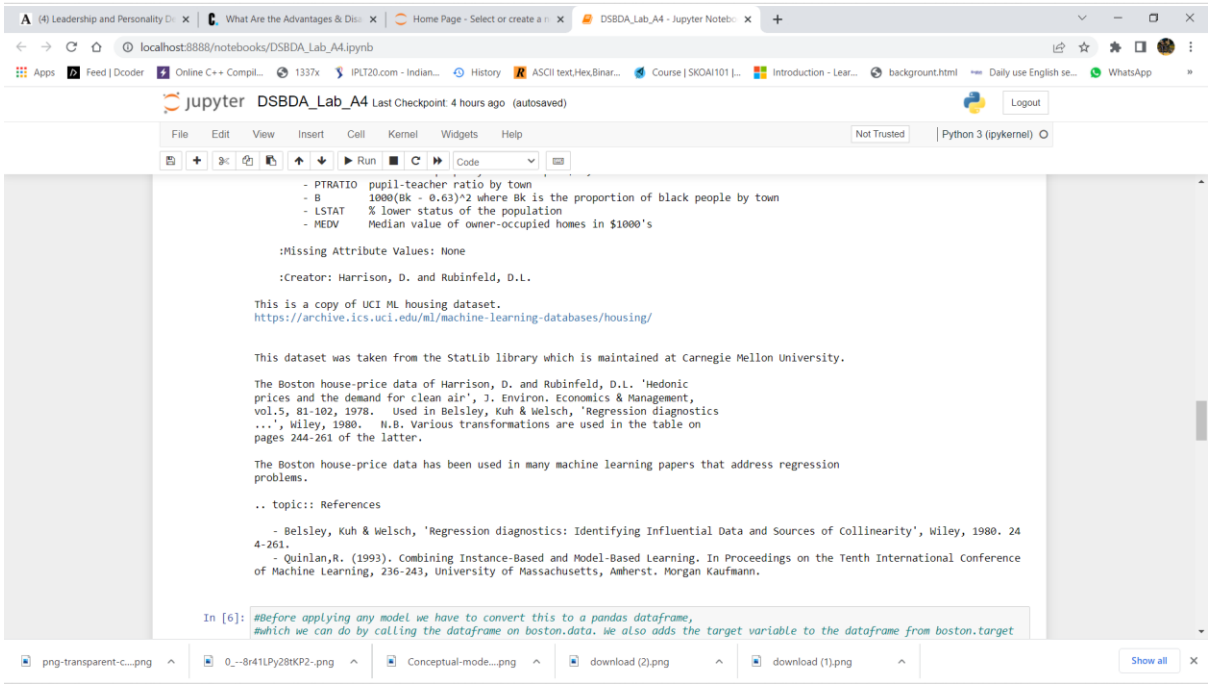

jupyter DSBDA_Lab_A4 Last Checkpoint: 4 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [3]: print(boston)
{'data': array([[6.3200e-03, 1.8000e+01, 2.3100e+00, ..., 1.5300e+01, 3.9690e+02,
4.9800e+00],
[2.7310e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9690e+02,
9.1400e+00],
[2.7290e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9283e+02,
4.0300e+00],
...,
[6.0760e-02, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9690e+02,
5.6400e+00],
[1.0959e-01, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9345e+02,
6.4800e+00],
[4.7410e-02, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9690e+02,
7.8800e+00]]), 'target': array([24. , 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5, 18.9, 15. ,
18.9, 21.7, 20.4, 18.2, 19.9, 23.1, 17.5, 20.2, 18.2, 13.6, 19.6,
15.2, 14.5, 15.6, 13.9, 16.6, 14.8, 18.4, 21. , 12.7, 14.5, 13.2,
13.1, 13.5, 18.9, 20. , 21. , 24.7, 30.8, 34.9, 26.6, 25.3, 24.7,
21.2, 19.3, 20. , 16.6, 14.4, 19.4, 19.7, 20.5, 25. , 23.4, 18.9,
35.4, 24.7, 31.6, 23.3, 19.6, 18.7, 16. , 22.2, 25. , 33. , 23.5,
19.4, 22. , 17.4, 20.9, 24.2, 21.7, 22.8, 23.4, 24.1, 21.4, 20. ,
20.8, 21.2, 20.3, 28. , 23.9, 24.8, 22.9, 23.9, 26.6, 22.5, 22.2,
23.6, 28.7, 22.6, 22. , 22.9, 25. , 20.6, 28.4, 21.4, 38.7, 43.8,
33.2, 27.5, 26.5, 18.6, 19.3, 20.1, 19.5, 19.5, 20.4, 19.8, 19.4,
21.7, 22.8, 18.8, 18.7, 18.5, 18.3, 21.2, 19.2, 20.4, 19.3, 22. ,
20.3, 20.5, 17.3, 18.8, 21.4, 15.7, 16.2, 18. , 14.3, 19.2, 19.6,
23. , 18.4, 15.6, 18.1, 17.4, 17.1, 13.3, 17.8, 14. , 14.4, 13.4,
15.6, 11.8, 13.8, 15.6, 14.6, 17.8, 15.4, 21.5, 19.6, 15.3, 19.4,
17. , 15.6, 13.1, 41.3, 24.3, 23.3, 27. , 50. , 50. , 50. , 22.7,
25. , 50. , 23.8, 23.8, 22.3, 17.4, 19.1, 23.1, 23.6, 22.6, 29.4,
23.2, 24.6, 29.9, 37.2, 39.8, 36.2, 37.9, 32.5, 26.4, 29.6, 50. ,
32. , 29.8, 34.9, 37. , 30.5, 36.4, 31.1, 29.1, 50. , 33.3, 30.3,
34.6, 34.9, 32.9, 24.1, 42.3, 48.5, 50. , 22.6, 24.4, 22.5, 24.4,
20. , 21.7, 19.3, 22.4, 28.1, 23.7, 25. , 23.3, 28.7, 21.5, 23. ,
26.7, 21.7, 27.5, 30.1, 44.8, 50. , 37.6, 31.6, 46.7, 31.5, 24.3,
31.7, 41.7, 48.3, 29. , 24. , 25.1, 31.5, 23.7, 23.3, 22. , 20.1,
22.2, 23.7, 17.6, 18.5, 24.3, 20.5, 24.5, 26.2, 24.4, 24.8, 29.6,
42.8, 21.9, 20.9, 44. , 50. , 36. , 30.1, 33.8, 43.1, 48.8, 31. ,
36.5, 22.8, 30.7, 50. , 43.5, 20.7, 21.1, 25.2, 24.4, 35.2, 32.4,
32. , 33.2, 33.1, 29.1, 35.1, 45.4, 35.4, 46. , 50. , 32.2, 22. ,
20.1, 23.2, 22.3, 24.8, 28.5, 37.3, 27.9, 23.9, 21.7, 28.6, 27.1,
20.3, 22.5, 29. , 24.8, 22. , 26.4, 33.1, 36.1, 28.4, 33.4, 28.2,
22.8, 20.3, 16.1, 22.1, 19.4, 21.6, 23.8, 16.2, 17.8, 19.8, 23.1,
21. , 23.8, 23.1, 20.4, 18.5, 25. , 24.6, 23. , 22.2, 19.3, 22.6,
19.8, 17.1, 19.4, 22.2, 20.7, 21.1, 19.5, 18.5, 20.6, 19. , 18.7,
32.7, 16.5, 23.9, 31.2, 17.5, 17.2, 23.1, 24.5, 26.6, 22.9, 24.1,
18.6, 30.1, 18.2, 20.6, 17.8, 21.7, 22.7, 22.6, 25. , 19.9, 20.8,
16.8, 21.9, 27.5, 21.9, 23.1, 50. , 50. , 50. , 50. , 50. , 13.8,
13.8, 15. , 13.9, 13.3, 13.1, 10.2, 10.4, 10.9, 11.3, 12.3, 8.8,
7.2, 10.5, 7.4, 10.2, 11.5, 15.1, 23.2, 9.7, 13.8, 12.7, 13.1,
12.5, 8.5, 5. , 6.3, 5.6, 7.2, 12.1, 8.3, 8.5, 5. , 11.9,
27.9, 17.2, 27.5, 15. , 17.2, 17.9, 16.3, 7. , 7.2, 7.5, 10.4,
8.8, 8.4, 16.7, 14.2, 20.8, 13.4, 11.7, 8.3, 10.2, 10.9, 11. ,
9.5, 14.5, 14.1, 16.1, 14.3, 11.7, 13.4, 9.6, 8.7, 8.4, 12.8,
10.5, 17.1, 18.4, 15.4, 10.8, 11.8, 14.9, 12.6, 14.1, 13. , 13.4,
15.2, 16.1, 17.8, 14.9, 14.1, 12.7, 13.5, 14.9, 20. , 16.4, 17.7,
19.5, 20.2, 21.4, 19.9, 19. , 19.1, 19.1, 20.1, 19.9, 19.6, 23.2,
29.8, 13.8, 13.3, 16.7, 12. , 14.6, 21.4, 23. , 23.7, 25. , 21.8,
20.6, 21.2, 19.1, 20.6, 15.2, 7. , 8.1, 13.6, 20.1, 21.8, 24.5,
23.1, 19.7, 18.3, 21.2, 17.5, 16.8, 22.4, 20.6, 23.9, 22. , 11.9]), 'feature_names': array(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD',
'TAX', 'PTRATIO', 'B', 'LSTAT'], dtype='<U7'), 'DESCR': ".. _boston_dataset:\n\nBoston house prices dataset\n---------------
----------\n\n**Data Set Characteristics:** \n\n   :Number of Instances: 506 \n\n   :Number of Attributes: 13 numeric/
categorical predictive. Median Value (attribute 14) is usually the target.\n\n   :Attribute Information (in order):\n
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

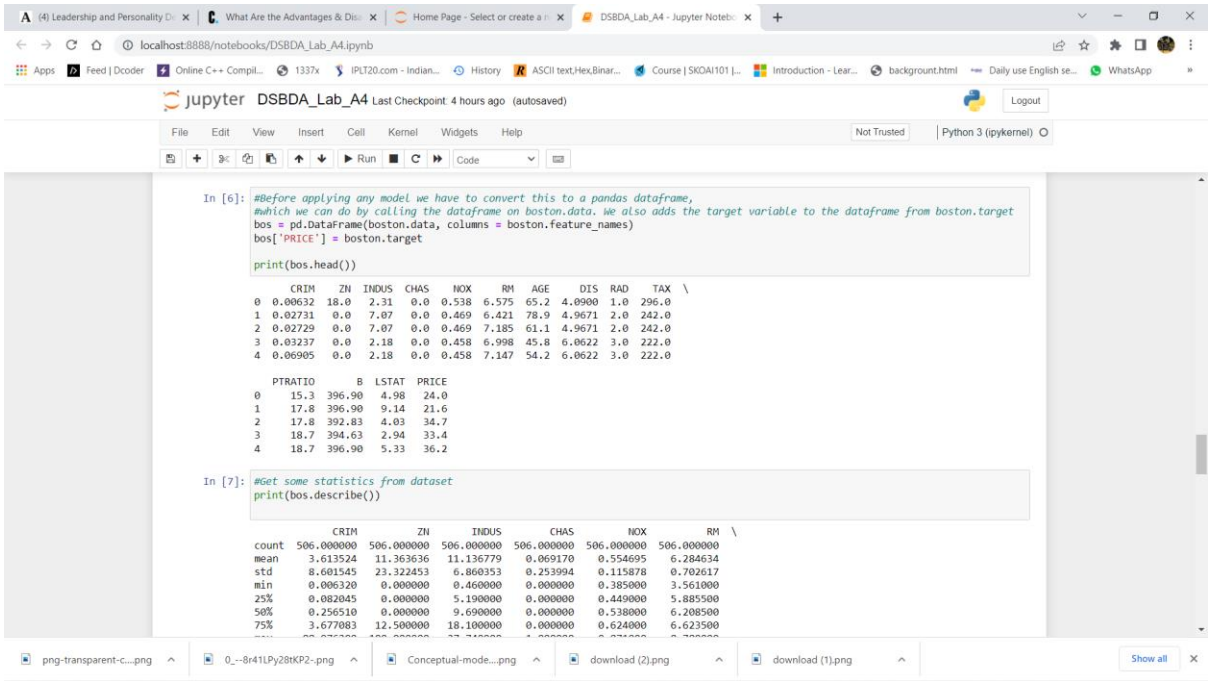

jupyter DSBDA_Lab_A4 Last Checkpoint: 4 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
S', 'NOX', 'RM', 'AGE', 'DIS', 'RAD',
'TAX', 'PTRATIO', 'B', 'LSTAT'], dtype='<U7'), 'DESCR': ".. _boston_dataset:\n\nBoston house prices dataset\n----------
----------\n\n**Data Set Characteristics:** \n\n   :Number of Instances: 506 \n\n   :Number of Attributes: 13 numeric/
categorical predictive. Median Value (attribute 14) is usually the target.\n\n   :Attribute Information (in order):\n   -
CRIM   per capita crime rate by town\n   - ZN   proportion of residential land zoned for lots over 25,000 sq.ft.\n
- INDUS   proportion of non-retail business acres per town\n   - CHAS   Charles River dummy variable (= 1 if tract boun
ds river; 0 otherwise)\n   - NOX   nitric oxides concentration (parts per 10 million)\n   - RM   average numbe
r of rooms per dwelling\n   - AGE   proportion of owner-occupied units built prior to 1940\n   - DIS   weighted
distances to five Boston employment centres\n   - RAD   index of accessibility to radial highways\n   - TAX   f
ull-value property-tax rate per $10,000\n   - PTRATIO pupil-teacher ratio by town\n   - B   1000(Bk - 0.63)^2 w
here Bk is the proportion of black people by town\n   - LSTAT   % lower status of the population\n   - MEDV   Medi
an value of owner-occupied homes in $1000's\n\n   :Missing Attribute Values: None\n\n   :Creator: Harrison, D. and Rubinfeld,
D.L.\n\nThis is a copy of UCI ML housing dataset.\nhttps://archive.ics.uci.edu/ml/machine-learning-databases/housing/\n\n\nThis
dataset was taken from the StatLib library which is maintained at Carnegie Mellon University.\n\nThe Boston house-price data of
Harrison, D. and Rubinfeld, D.L. 'Hedonic\nprices and the demand for clean air', J. Environ. Economics & Management,\nvol.5, 81
-102, 1978.   Used in Belsley, Kuh & Welsch, 'Regression diagnostics\n...', Wiley, 1980.   N.B. Various transformations are use
d in the table on\npages 244-261 of the latter.\n\nThe Boston house-price data has been used in many machine learning papers th
at address regression\nproblems.   \n   \n.. topic:: References\n\n   - Belsley, Kuh & Welsch, 'Regression diagnostics: Ident
ifying Influential Data and Sources of Collinearity', Wiley, 1980. 244-261.\n   - Quinlan,R. (1993). Combining Instance-Based a
nd Model-Based Learning. In Proceedings on the Tenth International Conference of Machine Learning, 236-243, University of Massa
chusetts, Amherst. Morgan Kaufmann.\n", 'filename': 'boston_house_prices.csv', 'data_module': 'sklearn.datasets.data'}
sklearn returns Dictionary-like object, the interesting attributes are: ‘data’, the data to learn, ‘target’, the regression targets, ‘DESCR’, the full description of the dataset, and ‘filename’, the physical location of boston csv dataset.
In [4]:
print(type(boston))
print('\n')
print(boston.keys())
print('\n')
print(boston.data.shape)
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all


jupyter DSBDA_Lab_A4 Last Checkpoint: 4 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [4]:
print(type(boston))
print('\n')
print(boston.keys())
print('\n')
print(boston.data.shape)
print('\n')
print(boston.feature_names)
<class 'sklearn.utils.Bunch'>
dict_keys(['data', 'target', 'feature_names', 'DESCR', 'filename', 'data_module'])
(506, 13)
['CRIM' 'ZN' 'INDUS' 'CHAS' 'NOX' 'RM' 'AGE' 'DIS' 'RAD' 'TAX' 'PTRATIO'
'B' 'LSTAT']
In [5]:
#The details about the features and more information about the dataset can be seen by using boston.DESCR
print(boston.DESCR)
.. _boston_dataset:
Boston house prices dataset

Data Set Characteristics:
:Number of Instances: 506
:Number of Attributes: 13 numeric/categorical predictive. Median Value (attribute 14) is usually the target.
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all


jupyter DSBDA_Lab_A4 Last Checkpoint: 4 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [5]:
#The details about the features and more information about the dataset can be seen by using boston.DESCR
print(boston.DESCR)
.. _boston_dataset:
Boston house prices dataset
Data Set Characteristics:
:Number of Instances: 506
:Number of Attributes: 13 numeric/categorical predictive. Median Value (attribute 14) is usually the target.
:Attribute Information (in order):
- CRIM per capita crime rate by town
- ZN proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS proportion of non-retail business acres per town
- CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- NOX nitric oxides concentration (parts per 10 million)
- RM average number of rooms per dwelling
- AGE proportion of owner-occupied units built prior to 1940
- DIS weighted distances to five Boston employment centres
- RAD index of accessibility to radial highways
- TAX full-value property-tax rate per $10,000
- PTRATIO pupil-teacher ratio by town
- B 1000(Bk - 0.63)^2 where Bk is the proportion of black people by town
- LSTAT % lower status of the population
- MEDV Median value of owner-occupied homes in $1000's
:Missing Attribute Values: None
:Creator: Harrison, D. and Rubinfeld, D.L.
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all


jupyter DSBDA_Lab_A4 Last Checkpoint: 4 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
- PTRATIO pupil-teacher ratio by town
- B 1000(Bk - 0.63)^2 where Bk is the proportion of black people by town
- LSTAT % lower status of the population
- MEDV Median value of owner-occupied homes in $1000's
:Missing Attribute Values: None
:Creator: Harrison, D. and Rubinfeld, D.L.
This is a copy of UCI ML housing dataset.
https://archive.ics.uci.edu/ml/machine-learning-databases/housing/
This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University.
The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic
prices and the demand for clean air', J. Environ. Economics & Management,
vol.5, 81-102, 1978. Used in Belsley, Kuh & Welsch, 'Regression diagnostics
...', Wiley, 1980. N.B. Various transformations are used in the table on
pages 244-261 of the latter.
The Boston house-price data has been used in many machine learning papers that address regression
problems.
.. topic:: References
- Belsley, Kuh & Welsch, 'Regression diagnostics: Identifying Influential Data and Sources of Collinearity', Wiley, 1980. 24
4-261.
- Quinlan,R. (1993). Combining Instance-Based and Model-Based Learning. In Proceedings on the Tenth International Conference
of Machine Learning, 236-243, University of Massachusetts, Amherst. Morgan Kaufmann.
In [6]:
#Before applying any model we have to convert this to a pandas dataframe,
#which we can do by calling the dataframe on boston.data. We also adds the target variable to the dataframe from boston.target
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all



(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a n... | DSBDA_Lab_A4 - Jupyter Noteb...

localhost:8888/notebooks/DSBDA_Lab_A4.ipynb

Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | background.html | Daily use English se... | WhatsApp

jupyter DSBDA_Lab_A4 Last Checkpoint: 4 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

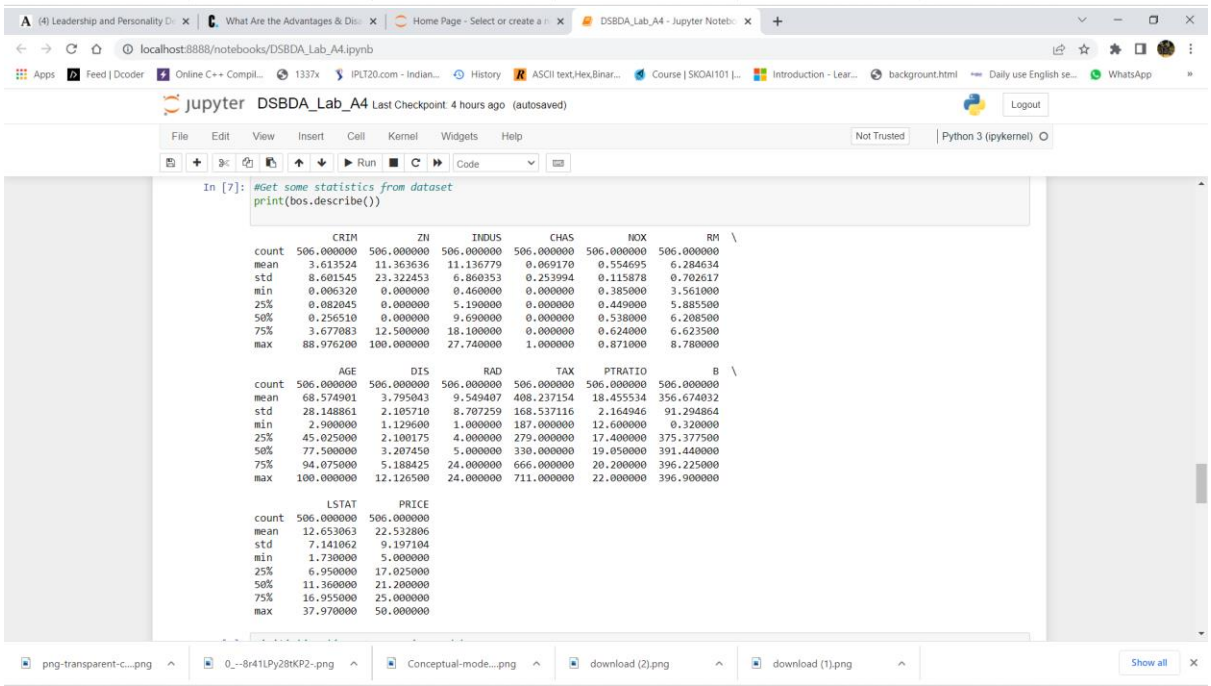
Run Code

In [6]:
```
#Before applying any model we have to convert this to a pandas dataframe,
#which we can do by calling the dataframe on boston.data. We also adds the target variable to the dataframe from boston.target
bos = pd.DataFrame(boston.data, columns = boston.feature_names)
bos['PRICE'] = boston.target

print(bos.head())
```

```
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD    TAX  \
0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900  1.0  296.0
1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671  2.0  242.0
2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671  2.0  242.0
3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622  3.0  222.0
4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622  3.0  222.0

   PTRATIO       B  LSTAT  PRICE
0     15.3  396.90   4.98   24.0
1     17.8  396.90   9.14   21.6
2     17.8  392.83   4.03   34.7
3     18.7  394.63   2.94   33.4
4     18.7  396.90   5.33   36.2
```

In [7]:
```
#Get some statistics from dataset
print(bos.describe())
```

```
             CRIM          ZN       INDUS        CHAS         NOX          RM  \
count  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000
mean     3.613524   11.363636   11.136779    0.069170    0.554695    6.284634
std      8.601545   23.322453    6.860353    0.253994    0.115878    0.702617
min      0.006320    0.000000    0.460000    0.000000    0.385000    3.561000
25%      0.082045    0.000000    5.190000    0.000000    0.449000    5.885500
50%      0.256510    0.000000    9.690000    0.000000    0.538000    6.208500
75%      3.677083   12.500000   18.100000    0.000000    0.624000    6.623500
```

(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a n... | DSBDA_Lab_A4 - Jupyter Noteb...

localhost:8888/notebooks/DSBDA_Lab_A4.ipynb

Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | background.html | Daily use English se... | WhatsApp

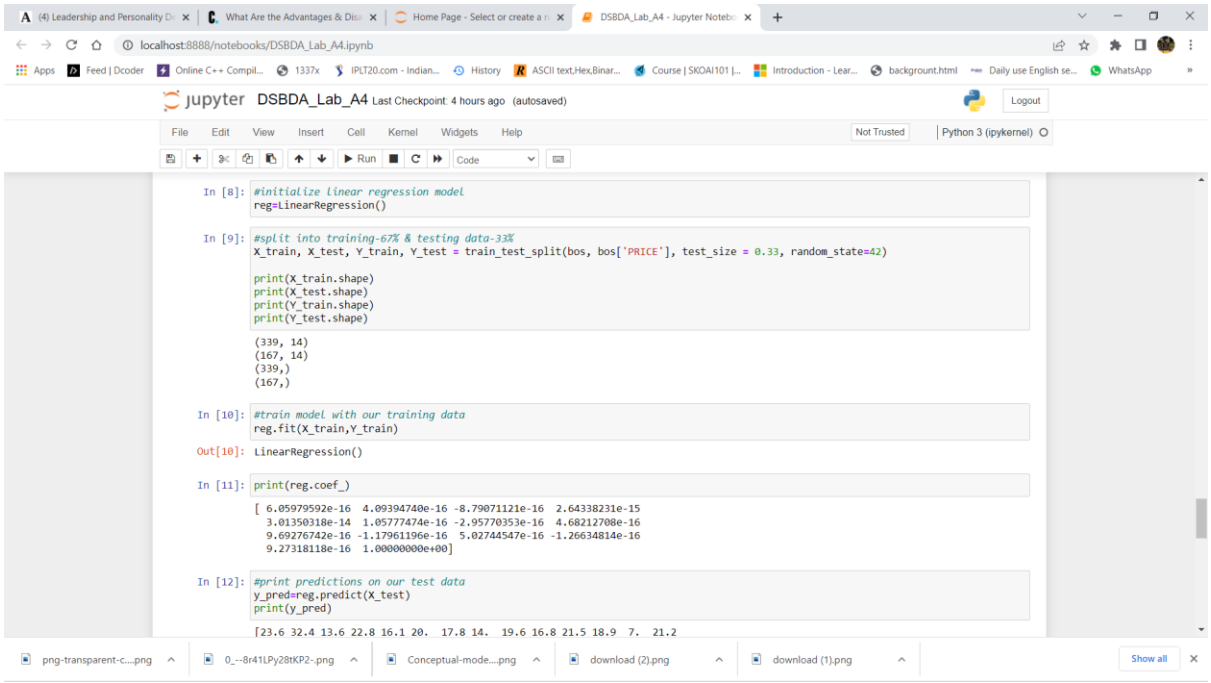
jupyter DSBDA_Lab_A4 Last Checkpoint: 4 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

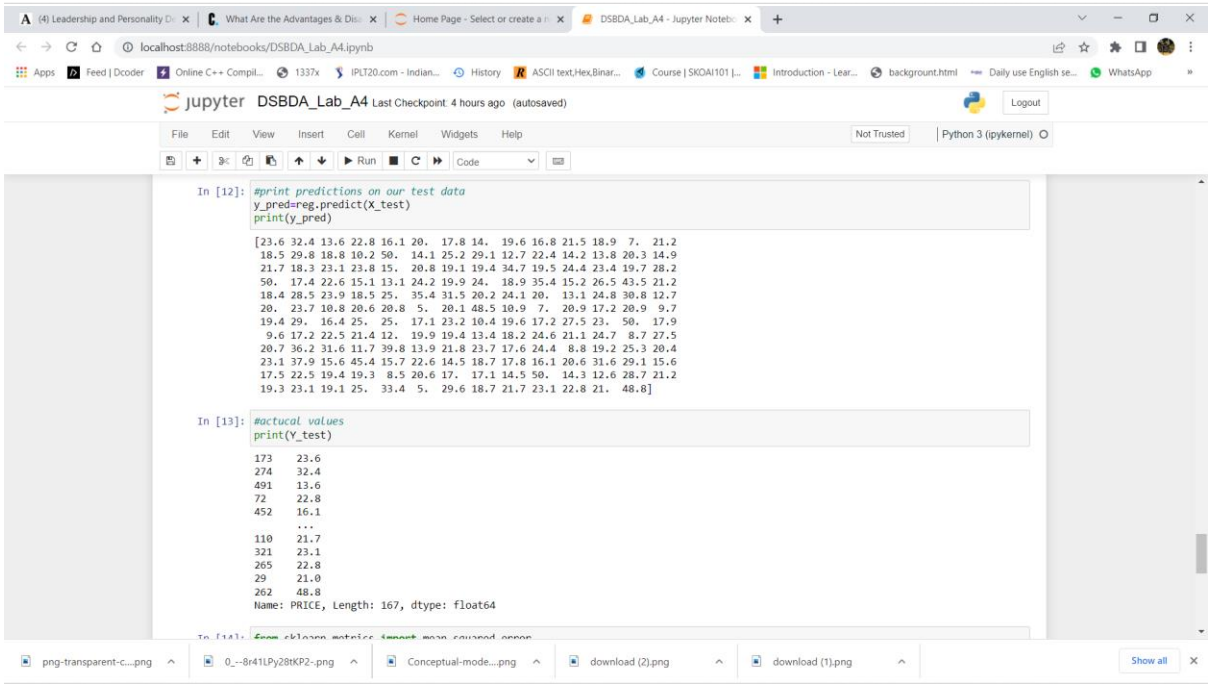
Run Code

In [7]:
```
#Get some statistics from dataset
print(bos.describe())
```

```
             CRIM          ZN       INDUS        CHAS         NOX          RM  \
count  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000
mean     3.613524   11.363636   11.136779    0.069170    0.554695    6.284634
std      8.601545   23.322453    6.860353    0.253994    0.115878    0.702617
min      0.006320    0.000000    0.460000    0.000000    0.385000    3.561000
25%      0.082045    0.000000    5.190000    0.000000    0.449000    5.885500
50%      0.256510    0.000000    9.690000    0.000000    0.538000    6.208500
75%      3.677083   12.500000   18.100000    0.000000    0.624000    6.623500
max     88.976200  100.000000   27.740000    1.000000    0.871000    8.780000

              AGE         DIS         RAD         TAX     PTRATIO           B  \
count  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000
mean    68.574901    3.795043    9.549407  408.237154   18.455534  356.674032
std     28.148861    2.105710    8.707259  168.537116    2.164946   91.294864
min      2.900000    1.129600    1.000000  187.000000   12.600000    0.320000
25%     45.025000    2.100175    4.000000  279.000000   17.400000  375.377500
50%     77.500000    3.207450    5.000000  330.000000   19.050000  391.440000
75%     94.075000    5.188425   24.000000  666.000000   20.200000  396.225000
max    100.000000   12.126500   24.000000  711.000000   22.000000  396.900000

            LSTAT       PRICE
count  506.000000  506.000000
mean    12.653063   22.532806
std      7.141062    9.197104
min      1.730000    5.000000
25%      6.950000   17.025000
50%     11.360000   21.200000
75%     16.955000   25.000000
max     37.970000   50.000000
```



```
In [8]: #initialize linear regression model
        reg=LinearRegression()

In [9]: #split into training-67% & testing data-33%
        X_train, X_test, Y_train, Y_test = train_test_split(bos, bos['PRICE'], test_size = 0.33, random_state=42)

        print(X_train.shape)
        print(X_test.shape)
        print(Y_train.shape)
        print(Y_test.shape)

        (339, 14)
        (167, 14)
        (339,)
        (167,)

In [10]: #train model with our training data
         reg.fit(X_train,Y_train)

Out[10]: LinearRegression()

In [11]: print(reg.coef_)

         [ 6.05979592e-16  4.09394740e-16 -8.79071121e-16  2.64338231e-15
           3.01350318e-14  1.05777474e-16 -2.95770353e-16  4.68212708e-16
           9.69276742e-16 -1.17961196e-16  5.02744547e-16 -1.26634814e-16
           9.27318118e-16  1.00000000e+00]

In [12]: #print predictions on our test data
         y_pred=reg.predict(X_test)
         print(y_pred)

         [23.6 32.4 13.6 22.8 16.1 20.  17.8 14.  19.6 16.8 21.5 18.9  7.  21.2
```

```
In [12]: #print predictions on our test data
         y_pred=reg.predict(X_test)
         print(y_pred)

         [23.6 32.4 13.6 22.8 16.1 20.  17.8 14.  19.6 16.8 21.5 18.9  7.  21.2
          18.5 29.8 18.8 10.2 50.  14.1 25.2 29.1 12.7 22.4 14.2 13.8 20.3 14.9
          21.7 18.3 23.1 23.8 15.  20.8 19.1 19.4 34.7 19.5 24.4 23.4 19.7 28.2
          50.  17.4 22.6 15.1 13.1 24.2 19.9 24.  18.9 35.4 15.2 26.5 43.5 21.2
          18.4 28.5 23.9 18.5 25.  35.4 31.5 20.2 24.1 20.  13.1 24.8 30.8 12.7
          20.  23.7 10.8 20.6 20.8  5.  20.1 48.5 10.9  7.  20.9 17.2 20.9  9.7
          19.4 29.  16.4 25.  25.  17.1 23.2 10.4 19.6 17.2 27.5 23.  50.  17.9
           9.6 17.2 22.5 21.4 12.  19.9 19.4 13.4 18.2 24.6 21.1 24.7  8.7 27.5
          20.7 36.2 31.6 11.7 39.8 13.9 21.8 23.7 17.6 24.4  8.8 19.2 25.3 20.4
          23.1 37.9 15.6 45.4 15.7 22.6 14.5 18.7 17.8 16.1 20.6 31.6 29.1 15.6
          17.5 22.5 19.4 19.3  8.5 20.6 17.  17.1 14.5 50.  14.3 12.6 28.7 21.2
          19.3 23.1 19.1 25.  33.4  5.  29.6 18.7 21.7 23.1 22.8 21.  48.8]

In [13]: #actucal values
         print(Y_test)

         173    23.6
         274    32.4
         491    13.6
         72     22.8
         452    16.1
                ...
         110    21.7
         321    23.1
         265    22.8
         29     21.0
         262    48.8
         Name: PRICE, Length: 167, dtype: float64

In [14]: from sklearn.metrics import mean_squared_error
```

(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a ... | DSBDA_Lab_A4 - Jupyter Noteb...

localhost:8888/notebooks/DSBDA_Lab_A4.ipynb

Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | background.html | Daily use English se... | WhatsApp

jupyter DSBDA_Lab_A4 Last Checkpoint: 4 hours ago (autosaved) Logout

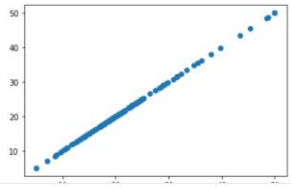
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

In [14]:
```
from sklearn.metrics import mean_squared_error
y_pred = reg.predict(X_test)
rmse = (np.sqrt(mean_squared_error(Y_test, y_pred)))
#r2 = round(reg.score(X_test, Y_test),2)

print("The model performance for training set")
print("--------------------------------------")
print("Root Mean Squared Error: {}".format(rmse))
#print("R^2: {}".format(r2))
print("\n")
```

```
The model performance for training set
--------------------------------------
Root Mean Squared Error: 2.294818025447708e-14
```

In [15]:
```
plt.scatter(Y_test, y_pred)
plt.show()
```



png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a ... | DSBDA_Lab_A4 - Jupyter Noteb...

localhost:8888/notebooks/DSBDA_Lab_A4.ipynb

Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | background.html | Daily use English se... | WhatsApp

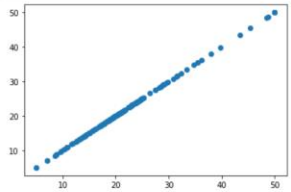
jupyter DSBDA_Lab_A4 Last Checkpoint: 4 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

```
The model performance for training set
--------------------------------------
Root Mean Squared Error: 2.294818025447708e-14
```

In [15]:
```
plt.scatter(Y_test, y_pred)
plt.show()
```



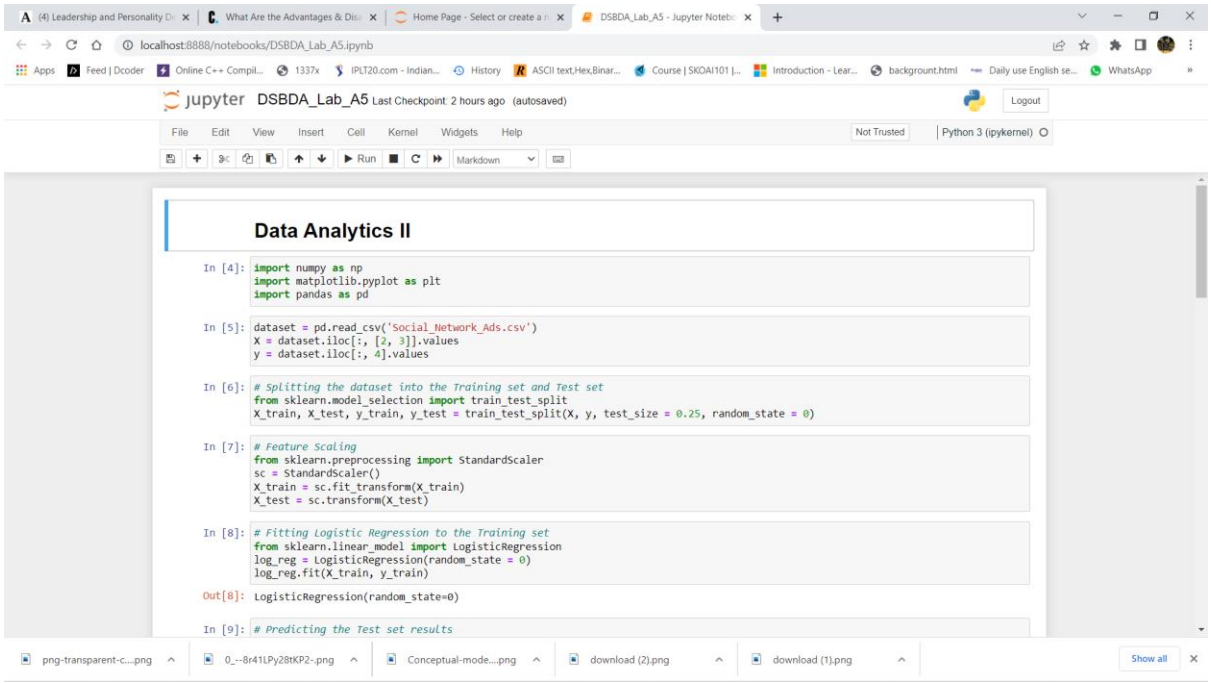
In [ ]:

png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

Data Science and Big Data Analytics

Group : A

Lab Assignment : 05



Data Analytics II

In [4]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [5]:
```
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
```

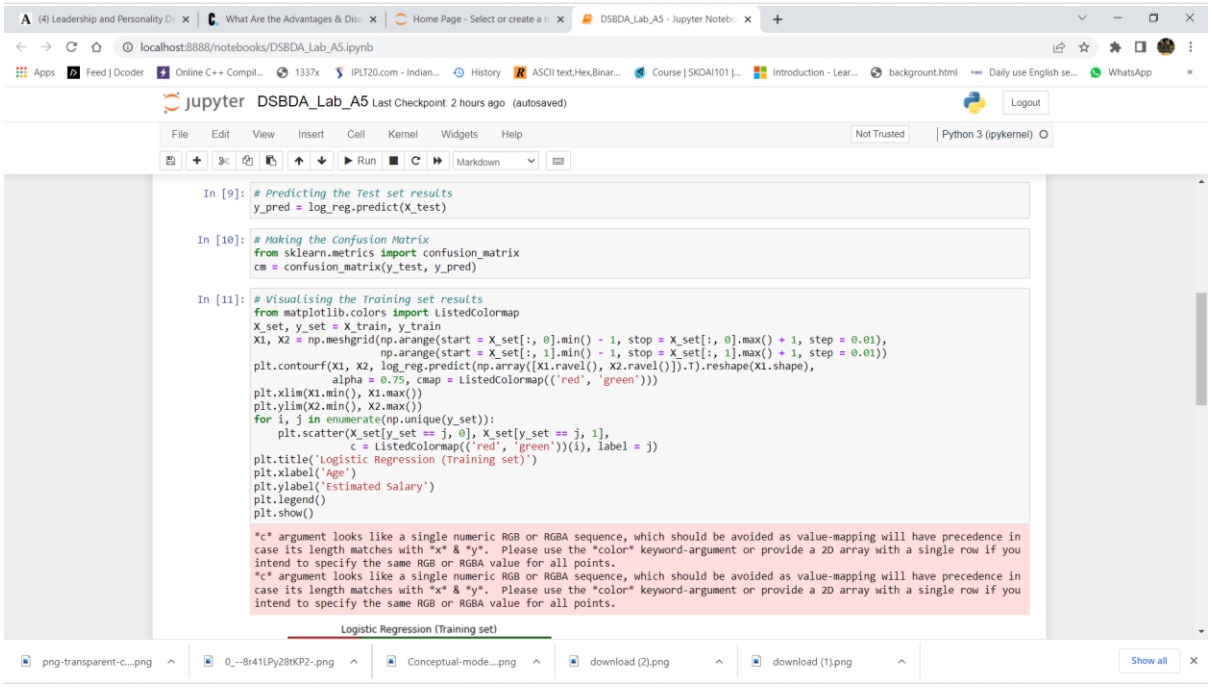
In [6]:
```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

In [7]:
```
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [8]:
```
# Fitting Logistic Regression to the Training set
from sklearn.linear_model import LogisticRegression
log_reg = LogisticRegression(random_state = 0)
log_reg.fit(X_train, y_train)
```

Out[8]:
```
LogisticRegression(random_state=0)
```

In [9]:
```
# Predicting the Test set results
```

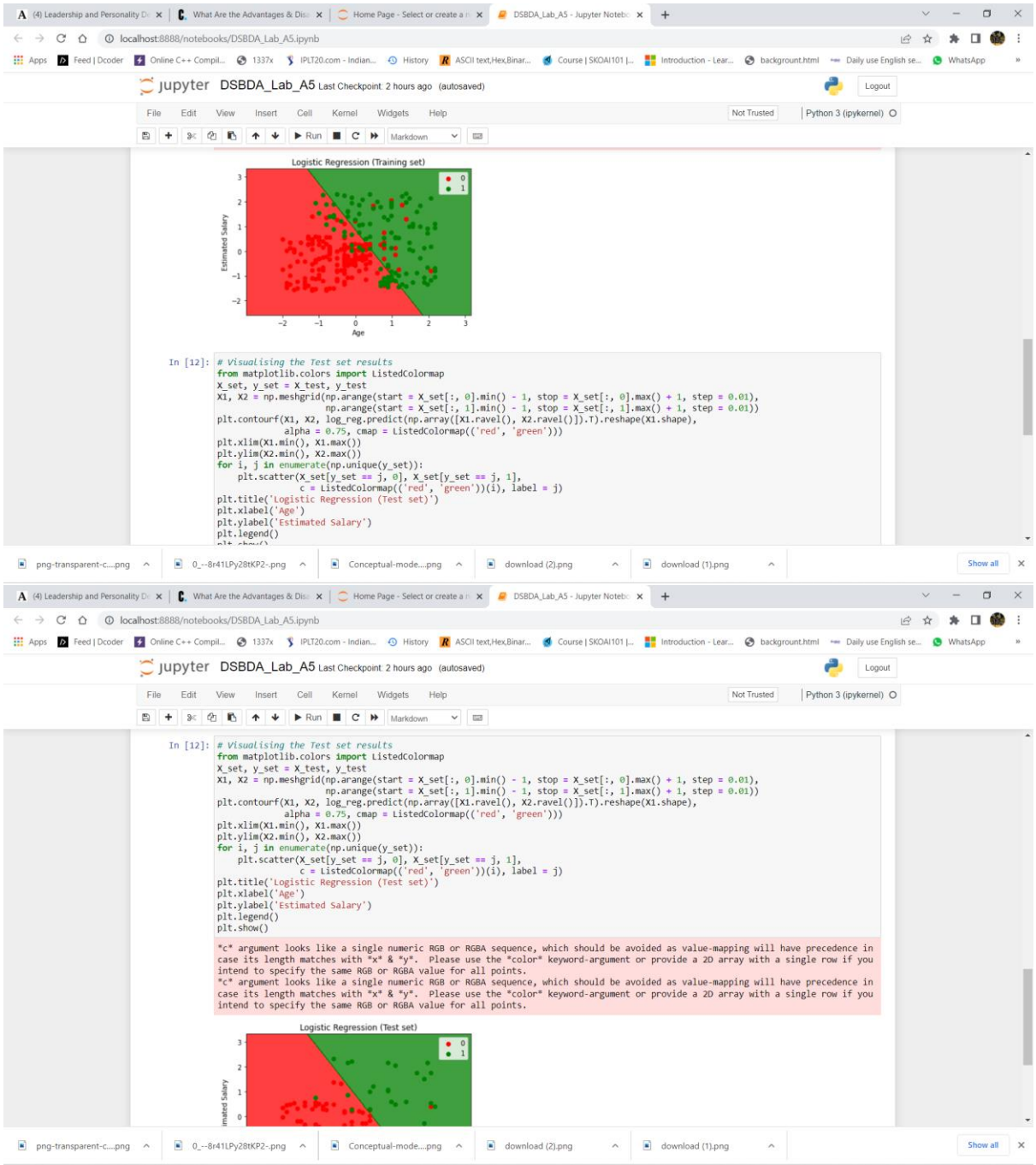


In [9]:
```
# Predicting the Test set results
y_pred = log_reg.predict(X_test)
```

In [10]:
```
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

In [11]:
```
# Visualising the Training set results
from matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, log_reg.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('Logistic Regression (Training set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in
case its length matches with *x* & *y*.  Please use the *color* keyword-argument or provide a 2D array with a single row if you
intend to specify the same RGB or RGBA value for all points.
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in
case its length matches with *x* & *y*.  Please use the *color* keyword-argument or provide a 2D array with a single row if you
intend to specify the same RGB or RGBA value for all points.
```


DSBDA_Lab_A5 Last Checkpoint: 2 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
Logistic Regression (Training set)
Estimated Salary
Age
0
1
In [12]:
# Visualising the Test set results
from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),
np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, log_reg.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('Logistic Regression (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

DSBDA_Lab_A5 Last Checkpoint: 2 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [12]:
# Visualising the Test set results
from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),
np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, log_reg.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('Logistic Regression (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
c argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*. Please use the *color* keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGBA value for all points.
c argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*. Please use the *color* keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGBA value for all points.
Logistic Regression (Test set)
0
1
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

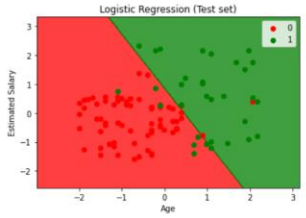
jupyter DSBDA_Lab_A5 Last Checkpoint: 2 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3 (ipykernel)

Run | Markdown

```
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('Logistic Regression (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*.  Please use the *color* keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGBA value for all points.
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*.  Please use the *color* keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGBA value for all points.
```

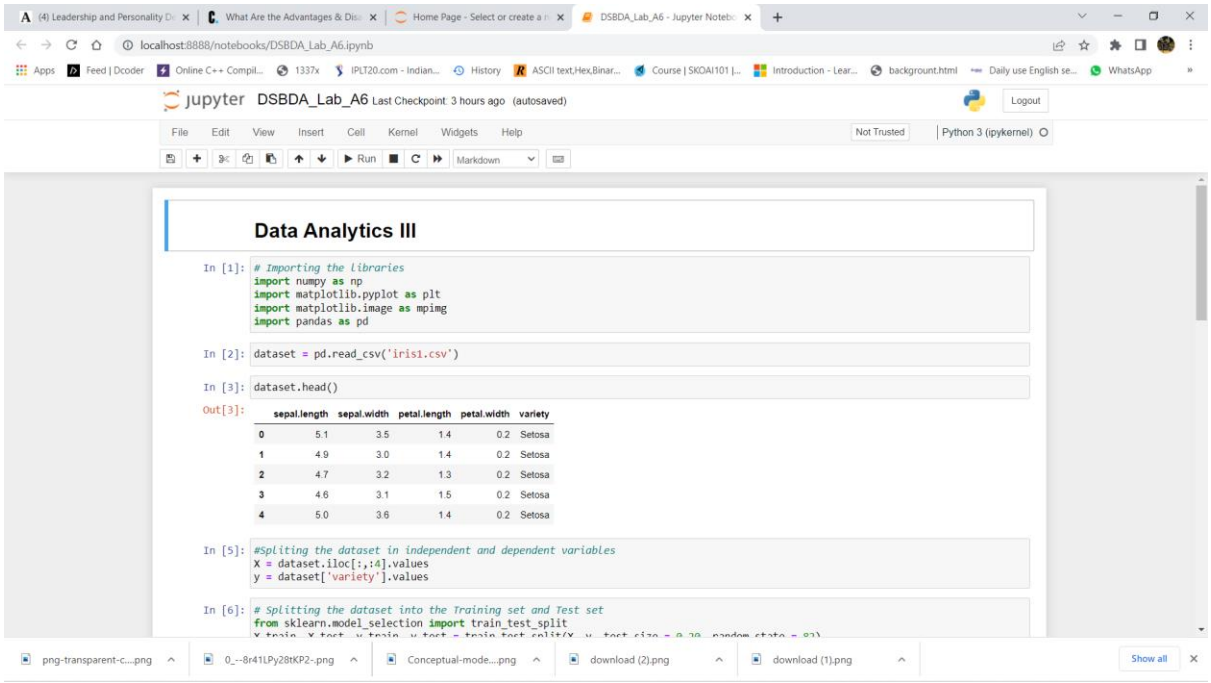


In [ ]:

Data Science and Big Data Analytics

Group : A

Lab Assignment : 06



Data Analytics III

In [1]:
```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import pandas as pd
```

In [2]:
```
dataset = pd.read_csv('iris1.csv')
```

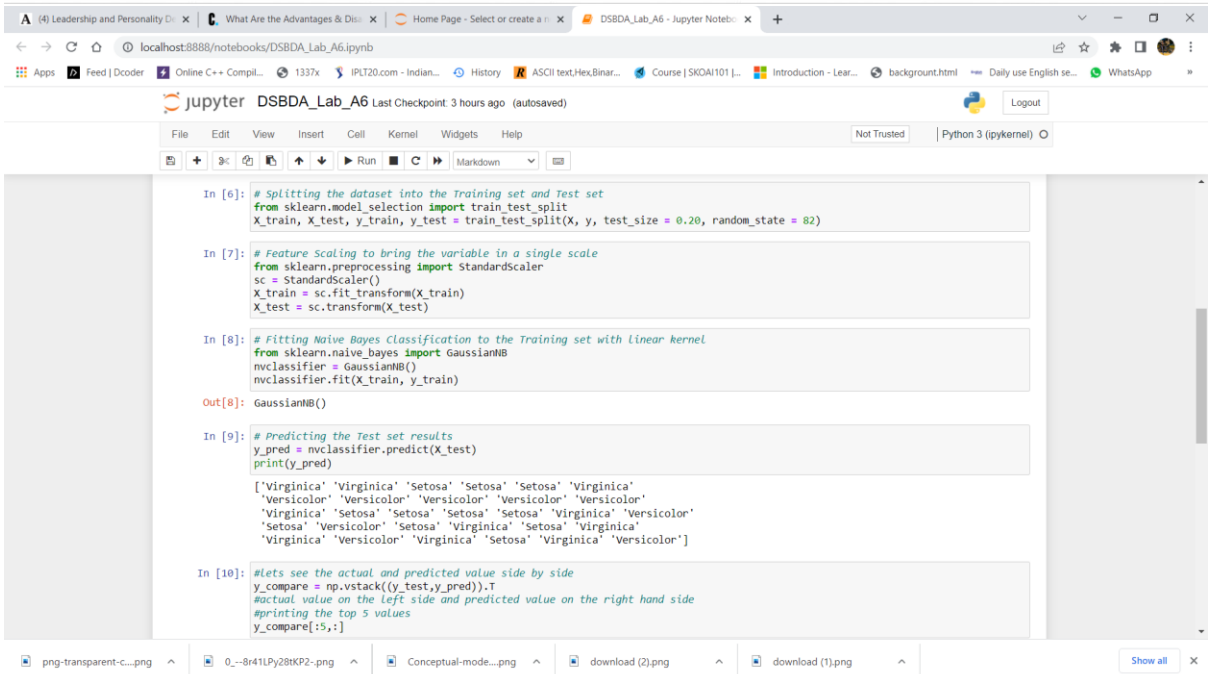
In [3]:
```
dataset.head()
```

Out[3]:

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa

In [5]:
```
#Spliting the dataset in independent and dependent variables
X = dataset.iloc[:,:4].values
y = dataset['variety'].values
```

In [6]:
```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
```



In [6]:
```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 82)
```

In [7]:
```
# Feature Scaling to bring the variable in a single scale
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

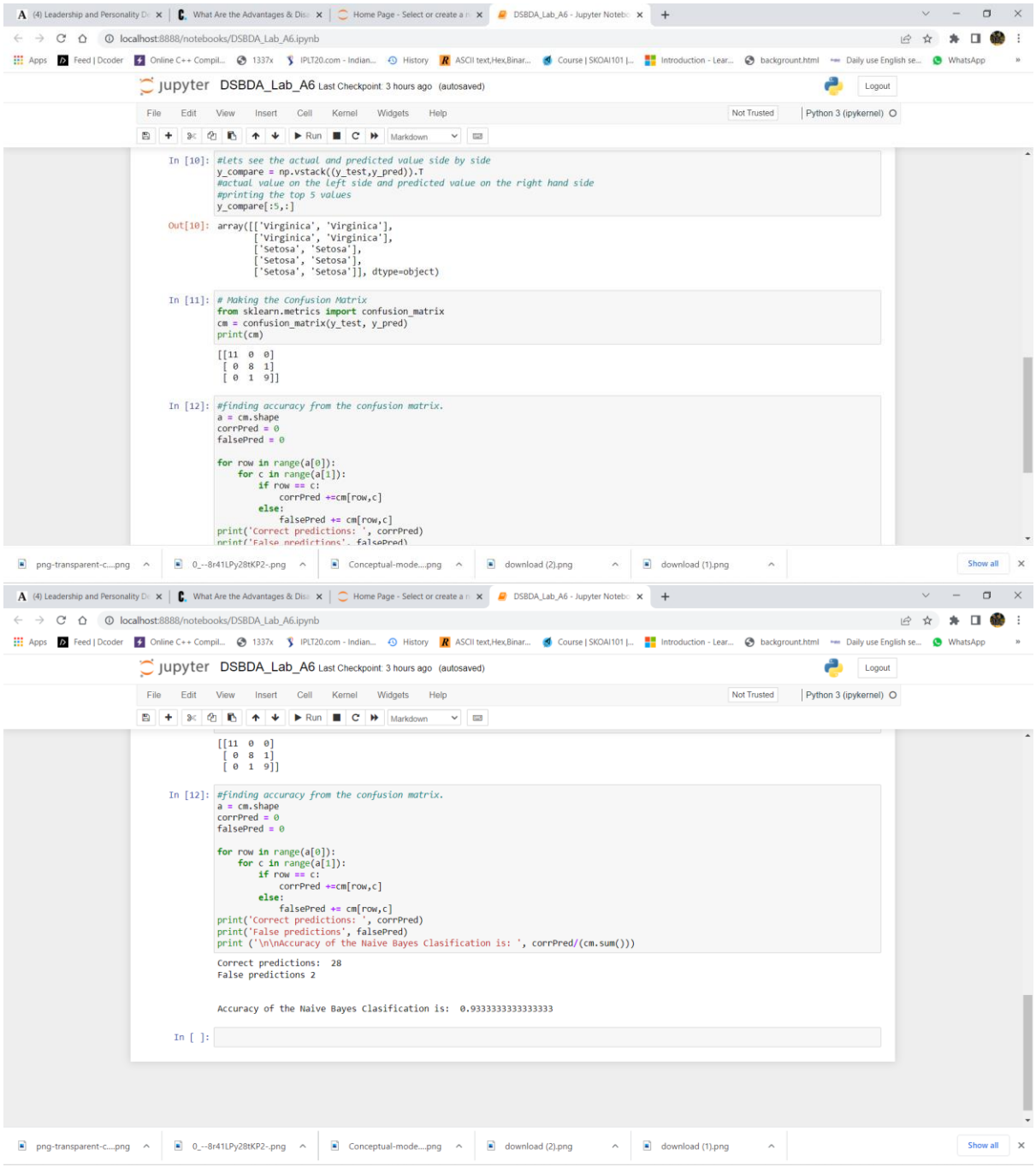
In [8]:
```
# Fitting Naive Bayes Classification to the Training set with linear kernel
from sklearn.naive_bayes import GaussianNB
nvclassifier = GaussianNB()
nvclassifier.fit(X_train, y_train)
```

Out[8]:
```
GaussianNB()
```

In [9]:
```
# Predicting the Test set results
y_pred = nvclassifier.predict(X_test)
print(y_pred)
```

```
['Virginica' 'Virginica' 'Setosa' 'Setosa' 'Setosa' 'Virginica'
 'Versicolor' 'Versicolor' 'Versicolor' 'Versicolor' 'Versicolor'
 'Virginica' 'Setosa' 'Setosa' 'Setosa' 'Setosa' 'Virginica' 'Versicolor'
 'Setosa' 'Versicolor' 'Setosa' 'Virginica' 'Setosa' 'Virginica'
 'Virginica' 'Versicolor' 'Virginica' 'Setosa' 'Virginica' 'Versicolor']
```

In [10]:
```
#lets see the actual and predicted value side by side
y_compare = np.vstack((y_test,y_pred)).T
#actual value on the left side and predicted value on the right hand side
#printing the top 5 values
y_compare[:5,:]
```



jupyter DSBDA_Lab_A6 Last Checkpoint: 3 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run | Markdown

In [10]:
```
#lets see the actual and predicted value side by side
y_compare = np.vstack((y_test,y_pred)).T
#actual value on the left side and predicted value on the right hand side
#printing the top 5 values
y_compare[:5,:]
```

Out[10]:
```
array([['Virginica', 'Virginica'],
       ['Virginica', 'Virginica'],
       ['Setosa', 'Setosa'],
       ['Setosa', 'Setosa'],
       ['Setosa', 'Setosa']], dtype=object)
```

In [11]:
```
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[11  0  0]
 [ 0  8  1]
 [ 0  1  9]]
```

In [12]:
```
#finding accuracy from the confusion matrix.
a = cm.shape
corrPred = 0
falsePred = 0

for row in range(a[0]):
    for c in range(a[1]):
        if row == c:
            corrPred +=cm[row,c]
        else:
            falsePred += cm[row,c]
print('Correct predictions: ', corrPred)
print('False predictions', falsePred)
```

png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

jupyter DSBDA_Lab_A6 Last Checkpoint: 3 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run | Markdown

```
[[11  0  0]
 [ 0  8  1]
 [ 0  1  9]]
```

In [12]:
```
#finding accuracy from the confusion matrix.
a = cm.shape
corrPred = 0
falsePred = 0

for row in range(a[0]):
    for c in range(a[1]):
        if row == c:
            corrPred +=cm[row,c]
        else:
            falsePred += cm[row,c]
print('Correct predictions: ', corrPred)
print('False predictions', falsePred)
print ('\n\nAccuracy of the Naive Bayes Clasification is: ', corrPred/(cm.sum()))
```

```
Correct predictions:  28
False predictions 2


Accuracy of the Naive Bayes Clasification is:  0.9333333333333333
```

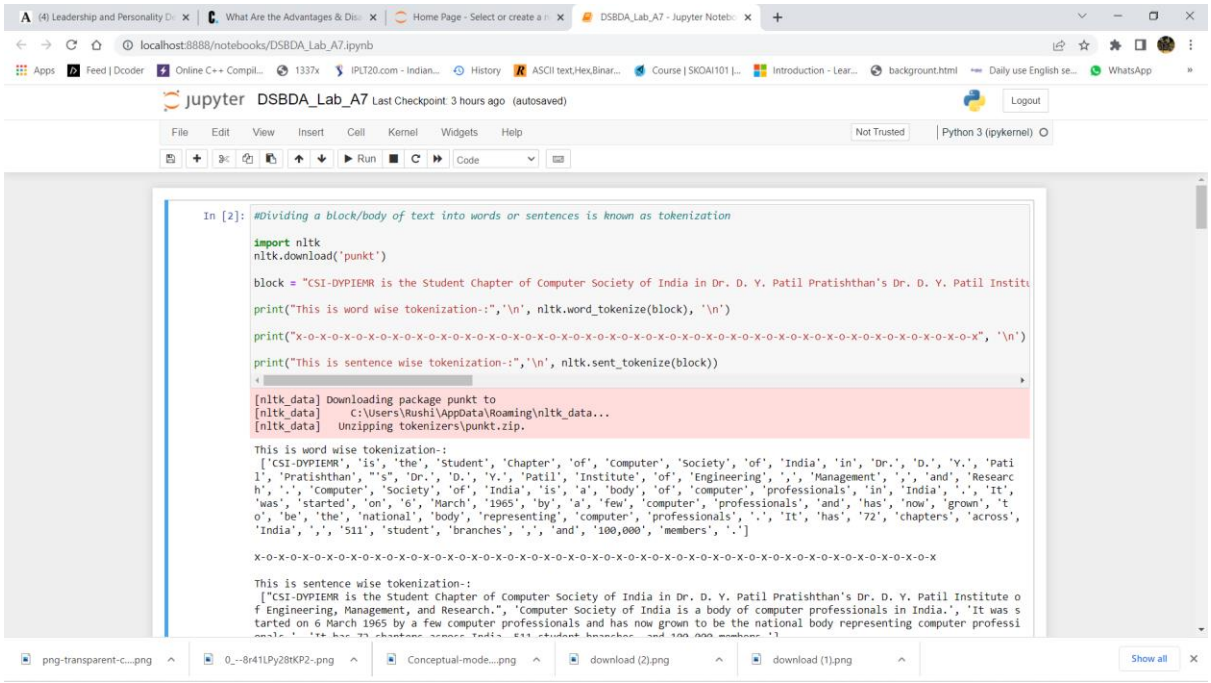
In [ ]:

png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

Data Science and Big Data Analytics

Group : A

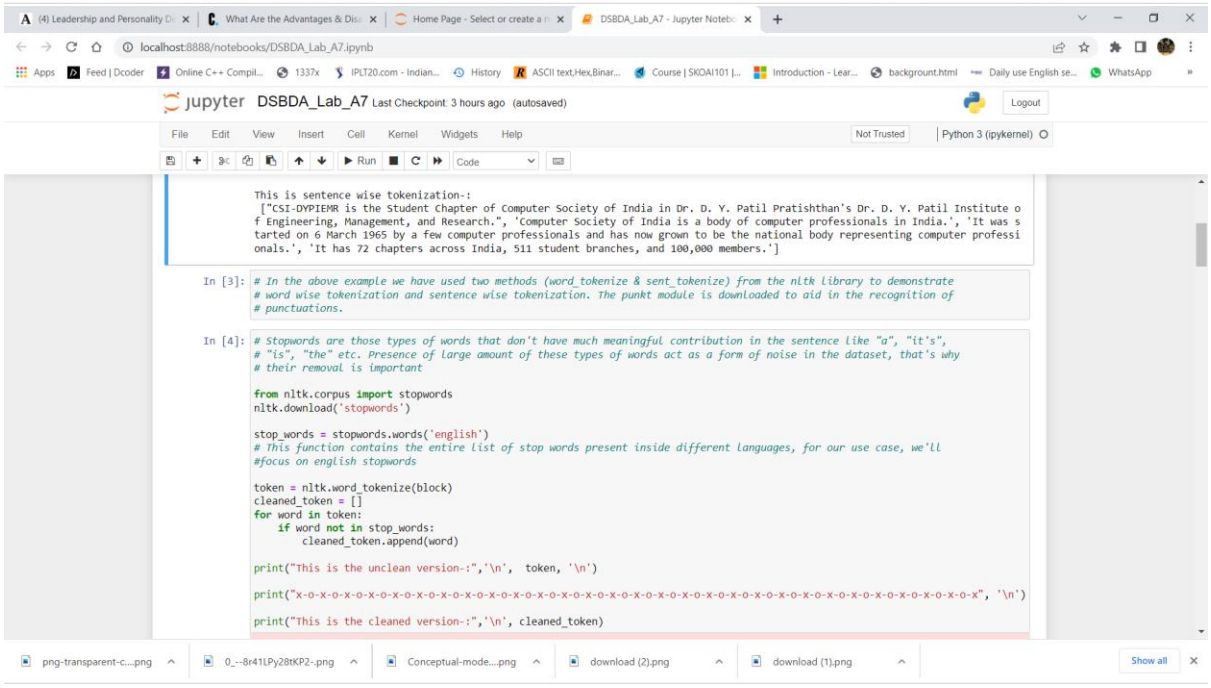
Lab Assignment : 07



```
In [2]: #Dividing a block/body of text into words or sentences is known as tokenization

import nltk
nltk.download('punkt')

block = "CSI-DYPIEMR is the Student Chapter of Computer Society of India in Dr. D. Y. Patil Pratishthan's Dr. D. Y. Patil Institu

print("This is word wise tokenization-:",'\n', nltk.word_tokenize(block), '\n')

print("x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x", '\n')

print("This is sentence wise tokenization-:",'\n', nltk.sent_tokenize(block))
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Rushi\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
```

```
This is word wise tokenization-:
 ['CSI-DYPIEMR', 'is', 'the', 'Student', 'Chapter', 'of', 'Computer', 'Society', 'of', 'India', 'in', 'Dr.', 'D.', 'Y.', 'Pati
l', 'Pratishthan', "'s", 'Dr.', 'D.', 'Y.', 'Patil', 'Institute', 'of', 'Engineering', ',', 'Management', ',', 'and', 'Researc
h', '.', 'Computer', 'Society', 'of', 'India', 'is', 'a', 'body', 'of', 'computer', 'professionals', 'in', 'India', '.', 'It',
'was', 'started', 'on', '6', 'March', '1965', 'by', 'a', 'few', 'computer', 'professionals', 'and', 'has', 'now', 'grown', 't
o', 'be', 'the', 'national', 'body', 'representing', 'computer', 'professionals', '.', 'It', 'has', '72', 'chapters', 'across',
'India', ',', '511', 'student', 'branches', ',', 'and', '100,000', 'members', '.']

x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x

This is sentence wise tokenization-:
 ["CSI-DYPIEMR is the Student Chapter of Computer Society of India in Dr. D. Y. Patil Pratishthan's Dr. D. Y. Patil Institute o
f Engineering, Management, and Research.", 'Computer Society of India is a body of computer professionals in India.', 'It was s
tarted on 6 March 1965 by a few computer professionals and has now grown to be the national body representing computer professi
onals.', 'It has 72 chapters across India, 511 student branches, and 100,000 members.']
```



```
In [3]: # In the above example we have used two methods (word_tokenize & sent_tokenize) from the nltk library to demonstrate
        # word wise tokenization and sentence wise tokenization. The punkt module is downloaded to aid in the recognition of
        # punctuations.

In [4]: # Stopwords are those types of words that don't have much meaningful contribution in the sentence like "a", "it's",
        # "is", "the" etc. Presence of large amount of these types of words act as a form of noise in the dataset, that's why
        # their removal is important

        from nltk.corpus import stopwords
        nltk.download('stopwords')

        stop_words = stopwords.words('english')
        # This function contains the entire list of stop words present inside different languages, for our use case, we'll
        #focus on english stopwords

        token = nltk.word_tokenize(block)
        cleaned_token = []
        for word in token:
            if word not in stop_words:
                cleaned_token.append(word)

        print("This is the unclean version-:",'\n',  token, '\n')

        print("x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x", '\n')

        print("This is the cleaned version-:",'\n', cleaned_token)
```

```
print('This is the cleaned version-:', \n', cleaned_token)
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Rushi\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

```
This is the unclean version-:
 ['CSI-DYPIEMR', 'is', 'the', 'Student', 'Chapter', 'of', 'Computer', 'Society', 'of', 'India', 'in', 'Dr.', 'D.', 'Y.', 'Pati
l', 'Pratishthan', "'s", 'Dr.', 'D.', 'Y.', 'Patil', 'Institute', 'of', 'Engineering', ',', 'Management', ',', 'and', 'Researc
h', '.', 'Computer', 'Society', 'of', 'India', 'is', 'a', 'body', 'of', 'computer', 'professionals', 'in', 'India', '.', 'It',
'was', 'started', 'on', '6', 'March', '1965', 'by', 'a', 'few', 'computer', 'professionals', 'and', 'has', 'now', 'grown', 'to
', 'be', 'the', 'national', 'body', 'representing', 'computer', 'professionals', '.', 'It', 'has', '72', 'chapters', 'across',
'India', ',', '511', 'student', 'branches', ',', 'and', '100,000', 'members', '.']

x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x-o-x

This is the cleaned version-:
 ['CSI-DYPIEMR', 'Student', 'Chapter', 'Computer', 'Society', 'India', 'Dr.', 'D.', 'Y.', 'Patil', 'Pratishthan', "'s", 'Dr.',
'D.', 'Y.', 'Patil', 'Institute', 'Engineering', ',', 'Management', ',', 'Research', '.', 'Computer', 'Society', 'India', 'bod
y', 'computer', 'professionals', 'India', '.', 'It', 'started', '6', 'March', '1965', 'computer', 'professionals', 'grown', 'na
tional', 'body', 'representing', 'computer', 'professionals', '.', 'It', '72', 'chapters', 'across', 'India', ',', '511', 'stud
ent', 'branches', ',', '100,000', 'members', '.']
```

In [5]:
```
# A sentence may contain words that convey the same meaning but are written in different forms, taking for example
# verbs in different forms of tenses like 'running', 'ran', 'run', 'runs' ultimately convey the same meaning but
# are written in different forms to suit the different types of tenses, for computer analysis these types of words
# are kept under same section as their base form (which in this case will be "run") and this process is known as stemming
```

In [6]:
```
from nltk.stem import PorterStemmer
stemmer = nltk.PorterStemmer()
words = ['rain', 'rained', 'raining', 'rains']
stemmed = [stemmer.stem(word) for word in words]
print(stemmed)
```

```
['rain', 'rain', 'rain', 'rain']
```


(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a n... | DSBDA_Lab_A7 - Jupyter Noteb...
localhost:8888/notebooks/DSBDA_Lab_A7.ipynb
Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | background.html | Daily use English se... | WhatsApp
jupyter DSBDA_Lab_A7 Last Checkpoint: 3 hours ago (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3 (ipykernel)
Run Code

In [6]:
```
from nltk.stem import PorterStemmer
stemmer = nltk.PorterStemmer()
words = ['rain', 'rained', 'raining', 'rains']
stemmed = [stemmer.stem(word) for word in words]
print(stemmed)
```

```
['rain', 'rain', 'rain', 'rain']
```

In [7]:
```
from nltk import pos_tag
nltk.download('averaged_perceptron_tagger')
tagged = nltk.pos_tag(cleaned_token)
print(tagged)
```

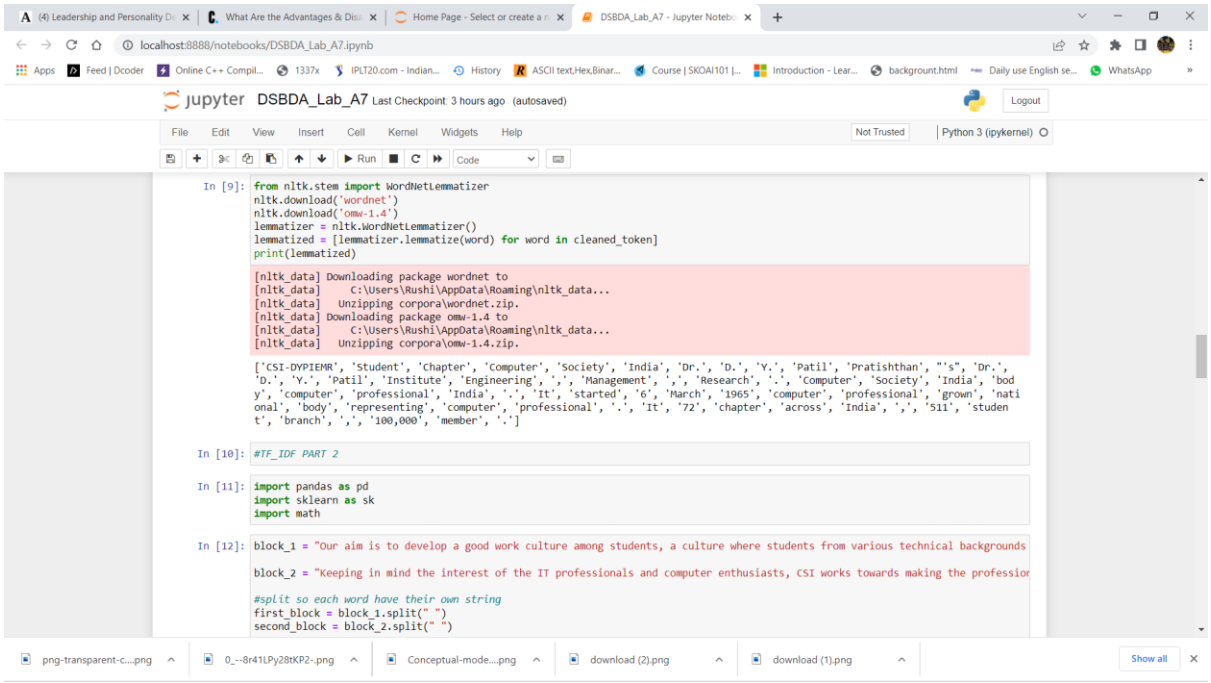
```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Rushi\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
```

```
[('CSI-DYPIEMR', 'JJ'), ('Student', 'NNP'), ('Chapter', 'NNP'), ('Computer', 'NNP'), ('Society', 'NNP'), ('India', 'NNP'), ('D
r.', 'NNP'), ('D.', 'NNP'), ('Y.', 'NNP'), ('Patil', 'NNP'), ('Pratishthan', 'NNP'), ("'s", 'POS'), ('Dr.', 'NNP'), ('D.', 'NN
P'), ('Y.', 'NNP'), ('Patil', 'NNP'), ('Institute', 'NNP'), ('Engineering', 'NNP'), (',', ','), ('Management', 'NNP'), (',',
','), ('Research', 'NNP'), ('.', '.'), ('Computer', 'NNP'), ('Society', 'NNP'), ('India', 'NNP'), ('body', 'NN'), ('computer',
'NN'), ('professionals', 'NNS'), ('India', 'NNP'), ('.', '.'), ('It', 'PRP'), ('started', 'VBD'), ('6', 'CD'), ('March', 'NN
P'), ('1965', 'CD'), ('computer', 'NN'), ('professionals', 'NNS'), ('grown', 'VBP'), ('national', 'JJ'), ('body', 'NN'), ('repr
esenting', 'VBG'), ('computer', 'NN'), ('professionals', 'NNS'), ('.', '.'), ('It', 'PRP'), ('72', 'CD'), ('chapters', 'NNS'),
('across', 'IN'), ('India', 'NNP'), (',', ','), ('511', 'CD'), ('student', 'NN'), ('branches', 'NNS'), (',', ','), ('100,000',
'CD'), ('members', 'NNS'), ('.', '.')]
```

In [8]:
```
# Lemmatization is the process of finding the form of the related word in the dictionary. Lemmatization does not
# simply chop off inflections. Instead, it uses lexical knowledge bases to get the correct base forms of words.
```

In [9]:
```
from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')
nltk.download('omw-1.4')
```

png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all



```
In [9]: from nltk.stem import WordNetLemmatizer
        nltk.download('wordnet')
        nltk.download('omw-1.4')
        lemmatizer = nltk.WordNetLemmatizer()
        lemmatized = [lemmatizer.lemmatize(word) for word in cleaned_token]
        print(lemmatized)
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Rushi\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\wordnet.zip.
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\Rushi\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\omw-1.4.zip.
```

```
['CSI-DYPIEMR', 'Student', 'Chapter', 'Computer', 'Society', 'India', 'Dr.', 'D.', 'Y.', 'Patil', 'Pratishthan', "'s", 'Dr.',
'D.', 'Y.', 'Patil', 'Institute', 'Engineering', ',', 'Management', ',', 'Research', '.', 'Computer', 'Society', 'India', 'bod
y', 'computer', 'professional', 'India', '.', 'It', 'started', '6', 'March', '1965', 'computer', 'professional', 'grown', 'nati
onal', 'body', 'representing', 'computer', 'professional', '.', 'It', '72', 'chapter', 'across', 'India', ',', '511', 'studen
t', 'branch', ',', '100,000', 'member', '.']
```

```
In [10]: #TF_IDF PART 2
```

```
In [11]: import pandas as pd
         import sklearn as sk
         import math
```

```
In [12]: block_1 = "Our aim is to develop a good work culture among students, a culture where students from various technical backgrounds

         block_2 = "Keeping in mind the interest of the IT professionals and computer enthusiasts, CSI works towards making the professior

         #split so each word have their own string
         first_block = block_1.split(" ")
         second_block = block_2.split(" ")
```

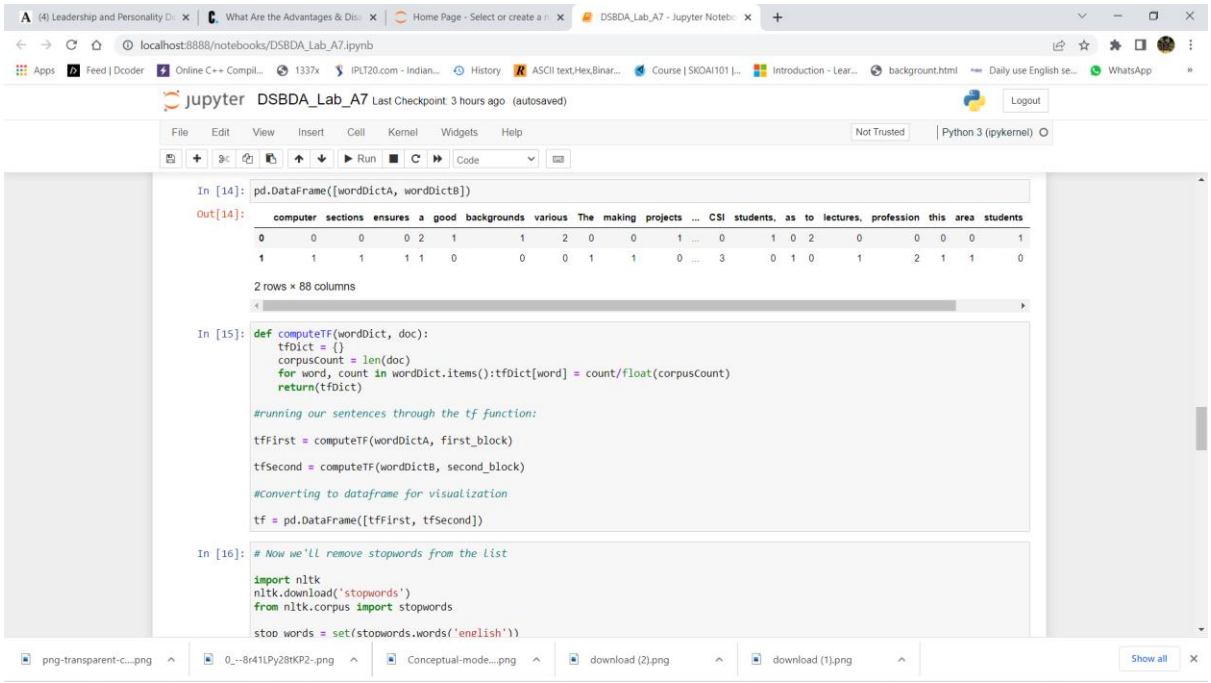

```
In [12]: block_1 = "Our aim is to develop a good work culture among students, a culture where students from various technical backgrounds

         block_2 = "Keeping in mind the interest of the IT professionals and computer enthusiasts, CSI works towards making the professior

         #split so each word have their own string
         first_block = block_1.split(" ")
         second_block = block_2.split(" ")
```


```
         #join them to remove common duplicate words
         total= set(first_block).union(set(second_block))

         print(total)
```

```
{'computer', 'sections', 'ensures', 'a', 'good', 'backgrounds', 'various', 'The', 'making', 'projects', 'of', 'professionals.',
'work', 'technical', 'works', 'promotion', 'each', 'at', 'together.', 'interest', 'where', 'Technology', 'organizes', 'award
s.', 'To', 'conventions,', 'is', 'organized', 'priority', 'the', 'And', 'top', 'today.', 'towards', 'projects,', 'with', 'als
o', 'same', 'on', 'amongst', 'all', 'training', 'develop', 'IT', 'in', 'conferences,', 'from', 'objective,', 'choice', 'keepin
g', 'that', 'among', 'come', 'teach,', 'it', 'society.', 'for', 'enthusiasts,', 'together', 'updating', 'future', 'Informatio
n', 'regular', 'culture', 'fulfill', 'an', 'other', 'skill', 'mind', 'and', 'regularly', 'time,', 'aim', 'Our', 'guide', 'colla
borate', 'grow', 'are', 'CSI', 'students,', 'as', 'to', 'lectures,', 'profession', 'this', 'area', 'students', 'professionals'}
```

```
In [13]: wordDictA = dict.fromkeys(total, 0)
         wordDictB = dict.fromkeys(total, 0)

         for word in first_block:
             wordDictA[word]+=1

         for word in second_block:
             wordDictB[word]+=1
```

```
In [14]: pd.DataFrame([wordDictA, wordDictB])
```



In [14]:
```
pd.DataFrame([wordDictA, wordDictB])
```

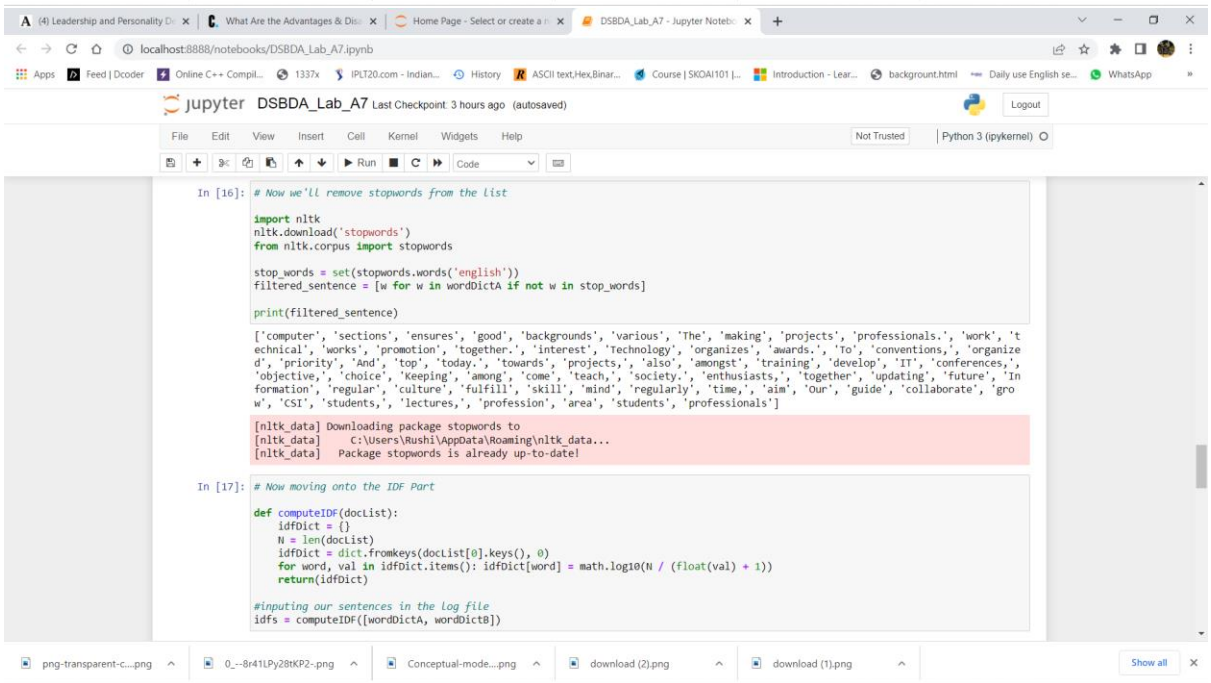
Out[14]:

	computer	sections	ensures	a	good	backgrounds	various	The	making	projects	...	CSI	students,	as	to	lectures,	profession	this	area	students
0	0	0	0	2	1	1	2	0	0	1	...	0	1	0	2	0	0	0	0	1
1	1	1	1	1	0	0	0	1	1	0	...	3	0	1	0	1	2	1	1	0

2 rows × 88 columns

In [15]:
```
def computeTF(wordDict, doc):
    tfDict = {}
    corpusCount = len(doc)
    for word, count in wordDict.items():tfDict[word] = count/float(corpusCount)
    return(tfDict)

#running our sentences through the tf function:

tfFirst = computeTF(wordDictA, first_block)

tfSecond = computeTF(wordDictB, second_block)

#Converting to dataframe for visualization

tf = pd.DataFrame([tfFirst, tfSecond])
```

In [16]:
```
# Now we'll remove stopwords from the list

import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

stop_words = set(stopwords.words('english'))
```

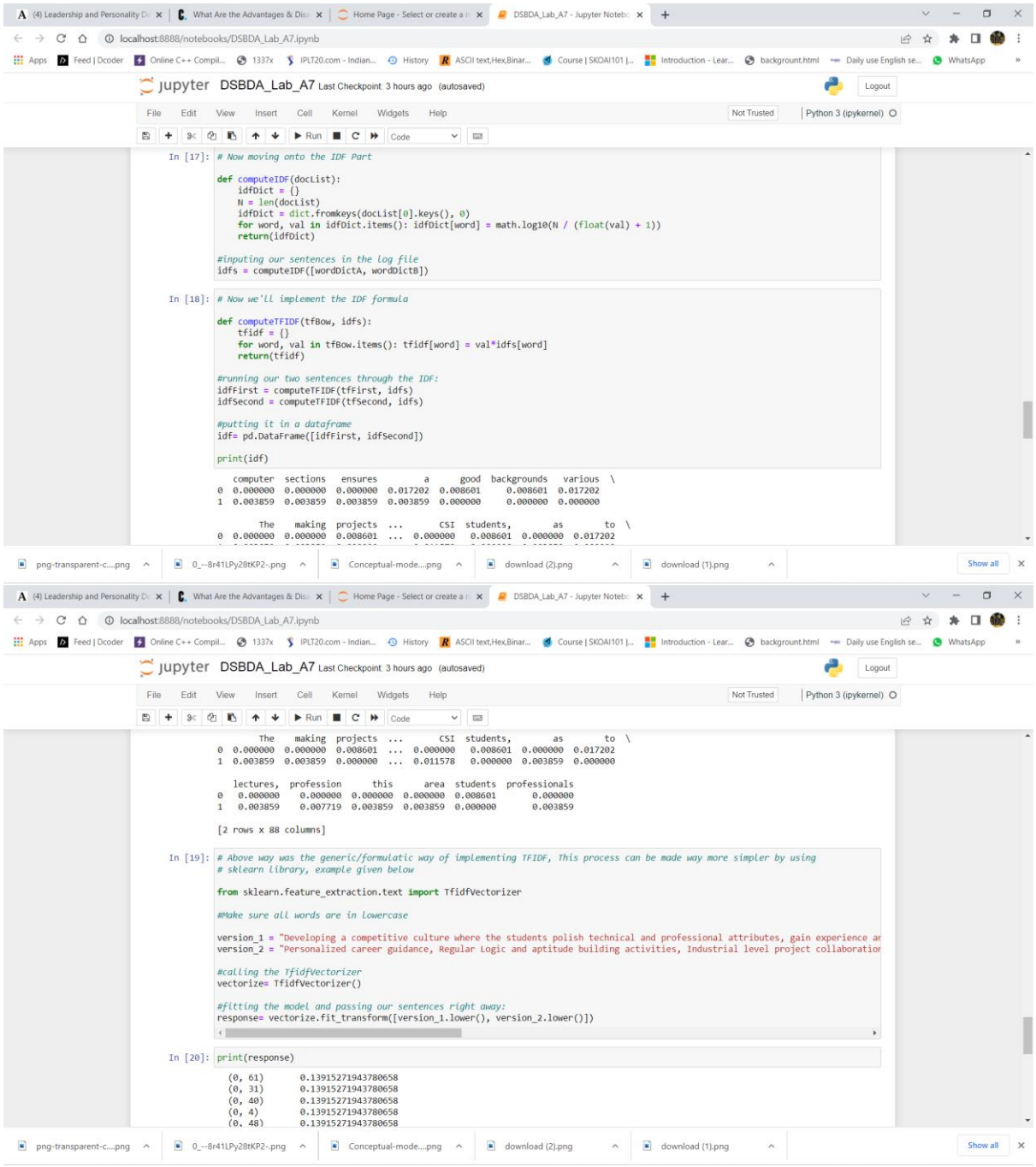



In [16]:
```
# Now we'll remove stopwords from the list

import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

stop_words = set(stopwords.words('english'))
filtered_sentence = [w for w in wordDictA if not w in stop_words]

print(filtered_sentence)
```

```
['computer', 'sections', 'ensures', 'good', 'backgrounds', 'various', 'The', 'making', 'projects', 'professionals.', 'work', 't
echnical', 'works', 'promotion', 'together.', 'interest', 'Technology', 'organizes', 'awards.', 'To', 'conventions,', 'organize
d', 'priority', 'And', 'top', 'today.', 'towards', 'projects,', 'also', 'amongst', 'training', 'develop', 'IT', 'conferences,',
'objective,', 'choice', 'Keeping', 'among', 'come', 'teach,', 'society.', 'enthusiasts,', 'together', 'updating', 'future', 'In
formation', 'regular', 'culture', 'fulfill', 'skill', 'mind', 'regularly', 'time,', 'aim', 'Our', 'guide', 'collaborate', 'gro
w', 'CSI', 'students,', 'lectures,', 'profession', 'area', 'students', 'professionals']
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Rushi\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [17]:
```
# Now moving onto the IDF Part

def computeIDF(docList):
    idfDict = {}
    N = len(docList)
    idfDict = dict.fromkeys(docList[0].keys(), 0)
    for word, val in idfDict.items(): idfDict[word] = math.log10(N / (float(val) + 1))
    return(idfDict)

#inputing our sentences in the log file
idfs = computeIDF([wordDictA, wordDictB])
```



jupyter DSBDA_Lab_A7 Last Checkpoint: 3 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

In [17]:
```
# Now moving onto the IDF Part

def computeIDF(docList):
    idfDict = {}
    N = len(docList)
    idfDict = dict.fromkeys(docList[0].keys(), 0)
    for word, val in idfDict.items(): idfDict[word] = math.log10(N / (float(val) + 1))
    return(idfDict)

#inputing our sentences in the log file
idfs = computeIDF([wordDictA, wordDictB])
```

In [18]:
```
# Now we'll implement the IDF formula

def computeTFIDF(tfBow, idfs):
    tfidf = {}
    for word, val in tfBow.items(): tfidf[word] = val*idfs[word]
    return(tfidf)

#running our two sentences through the IDF:
idfFirst = computeTFIDF(tfFirst, idfs)
idfSecond = computeTFIDF(tfSecond, idfs)

#putting it in a dataframe
idf= pd.DataFrame([idfFirst, idfSecond])

print(idf)
```

```
   computer  sections   ensures         a      good  backgrounds   various  \
0  0.000000  0.000000  0.000000  0.017202  0.008601     0.008601  0.017202
1  0.003859  0.003859  0.003859  0.003859  0.000000     0.000000  0.000000

        The    making  projects  ...       CSI  students,        as        to  \
0  0.000000  0.000000  0.008601  ...  0.000000   0.008601  0.000000  0.017202
```

png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

jupyter DSBDA_Lab_A7 Last Checkpoint: 3 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

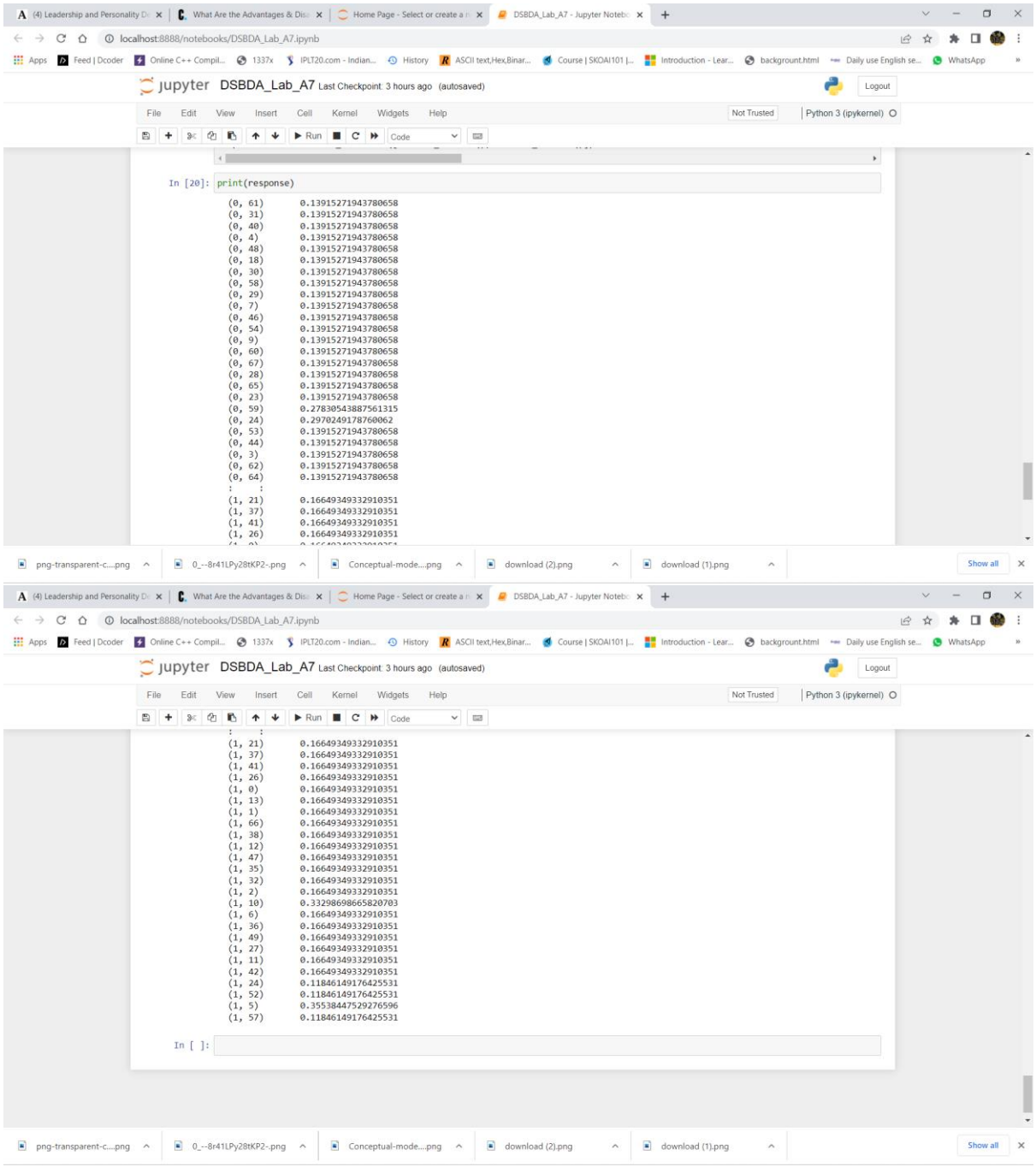
```
        The    making  projects  ...       CSI  students,        as        to  \
0  0.000000  0.000000  0.008601  ...  0.000000   0.008601  0.000000  0.017202
1  0.003859  0.003859  0.000000  ...  0.011578   0.000000  0.003859  0.000000

   lectures,  profession      this      area  students  professionals
0   0.000000    0.000000  0.000000  0.000000  0.008601       0.000000
1   0.003859    0.007719  0.003859  0.003859  0.000000       0.003859

[2 rows x 88 columns]
```

In [19]:
```
# Above way was the generic/formulatic way of implementing TFIDF, This process can be made way more simpler by using
# sklearn library, example given below

from sklearn.feature_extraction.text import TfidfVectorizer

#Make sure all words are in lowercase

version_1 = "Developing a competitive culture where the students polish technical and professional attributes, gain experience ar
version_2 = "Personalized career guidance, Regular Logic and aptitude building activities, Industrial level project collaboratior

#calling the TfidfVectorizer
vectorize= TfidfVectorizer()

#fitting the model and passing our sentences right away:
response= vectorize.fit_transform([version_1.lower(), version_2.lower()])
```

In [20]:
```
print(response)
```

```
  (0, 61)       0.13915271943780658
  (0, 31)       0.13915271943780658
  (0, 40)       0.13915271943780658
  (0, 4)        0.13915271943780658
  (0, 48)       0.13915271943780658
```

png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

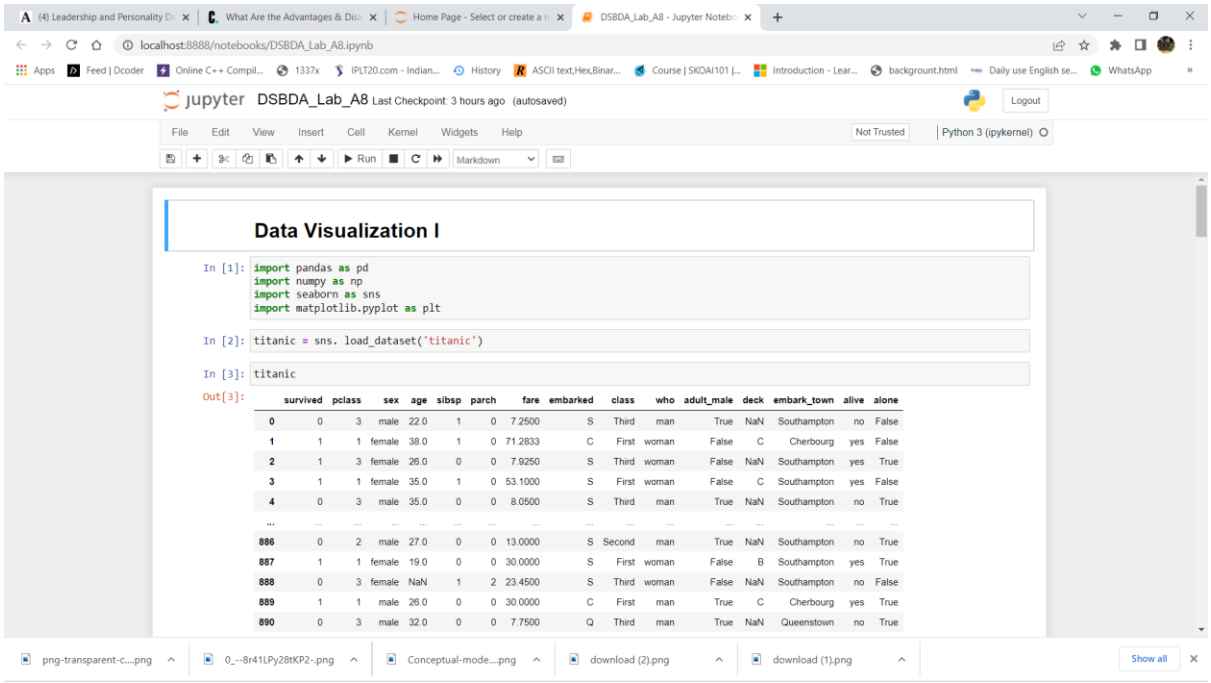

jupyter DSBDA_Lab_A7 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [20]: print(response)
(0, 61) 0.13915271943780658
(0, 31) 0.13915271943780658
(0, 40) 0.13915271943780658
(0, 4) 0.13915271943780658
(0, 48) 0.13915271943780658
(0, 18) 0.13915271943780658
(0, 30) 0.13915271943780658
(0, 58) 0.13915271943780658
(0, 29) 0.13915271943780658
(0, 7) 0.13915271943780658
(0, 46) 0.13915271943780658
(0, 54) 0.13915271943780658
(0, 9) 0.13915271943780658
(0, 60) 0.13915271943780658
(0, 67) 0.13915271943780658
(0, 28) 0.13915271943780658
(0, 65) 0.13915271943780658
(0, 23) 0.13915271943780658
(0, 59) 0.27830543887561315
(0, 24) 0.2970249178760062
(0, 53) 0.13915271943780658
(0, 44) 0.13915271943780658
(0, 3) 0.13915271943780658
(0, 62) 0.13915271943780658
(0, 64) 0.13915271943780658
: :
(1, 21) 0.16649349332910351
(1, 37) 0.16649349332910351
(1, 41) 0.16649349332910351
(1, 26) 0.16649349332910351
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

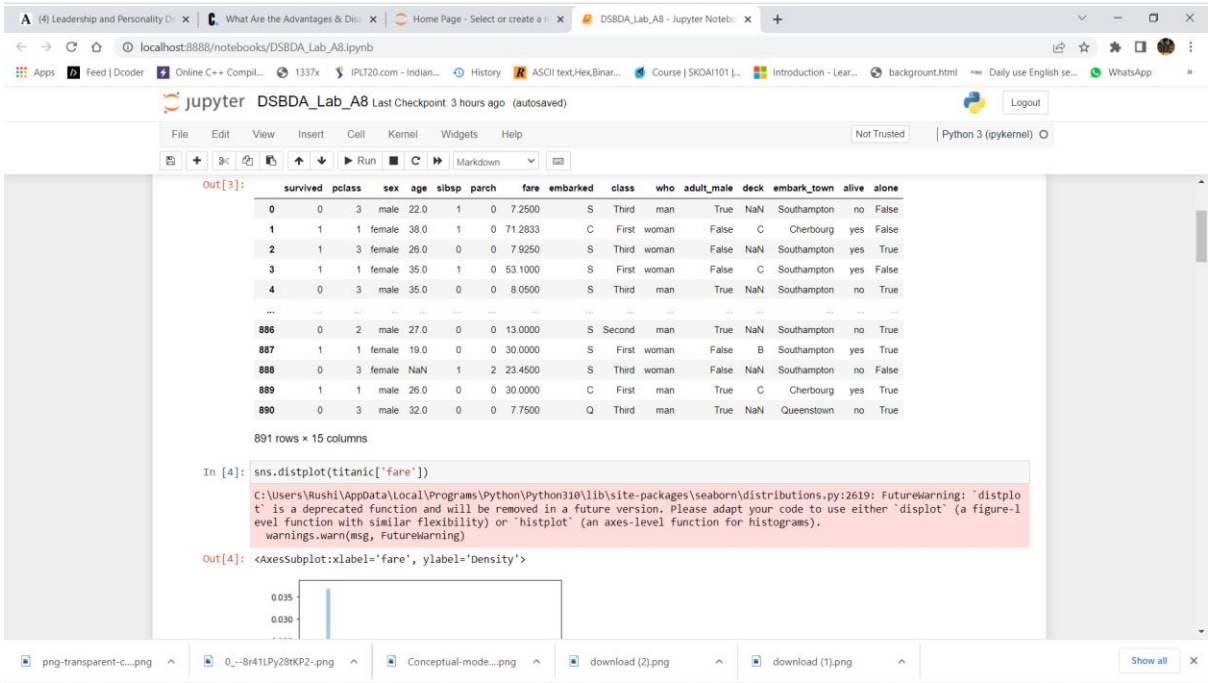
jupyter DSBDA_Lab_A7 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
: :
(1, 21) 0.16649349332910351
(1, 37) 0.16649349332910351
(1, 41) 0.16649349332910351
(1, 26) 0.16649349332910351
(1, 0) 0.16649349332910351
(1, 13) 0.16649349332910351
(1, 1) 0.16649349332910351
(1, 66) 0.16649349332910351
(1, 38) 0.16649349332910351
(1, 12) 0.16649349332910351
(1, 47) 0.16649349332910351
(1, 35) 0.16649349332910351
(1, 32) 0.16649349332910351
(1, 2) 0.16649349332910351
(1, 10) 0.33298698665820703
(1, 6) 0.16649349332910351
(1, 36) 0.16649349332910351
(1, 49) 0.16649349332910351
(1, 27) 0.16649349332910351
(1, 11) 0.16649349332910351
(1, 42) 0.16649349332910351
(1, 24) 0.11846149176425531
(1, 52) 0.11846149176425531
(1, 5) 0.35538447529276596
(1, 57) 0.11846149176425531
In [ ]:
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

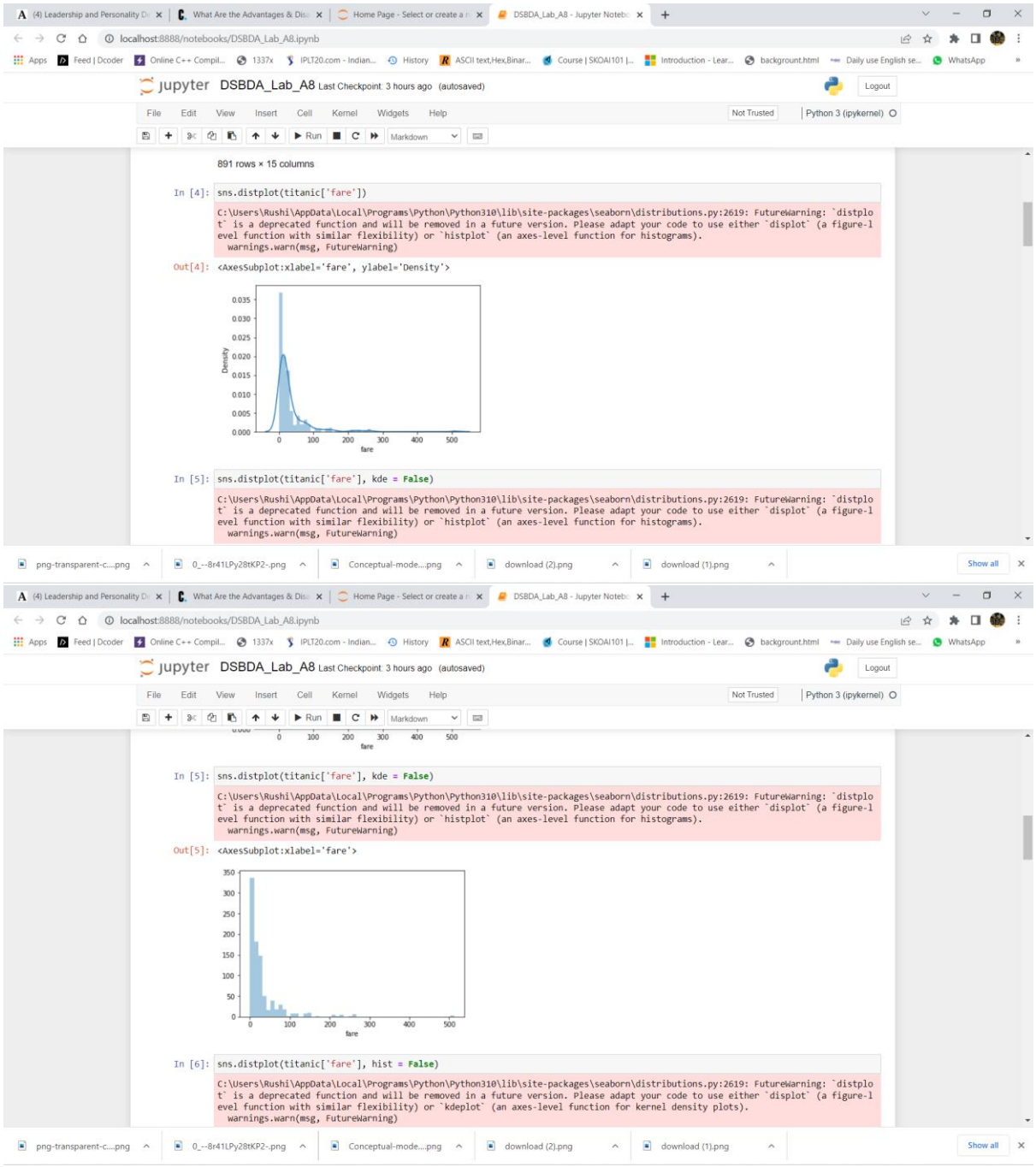
Data Science and Big Data Analytics

Group : A

Lab Assignment : 08






jupyter DSBDA_Lab_A8 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
891 rows × 15 columns
In [4]:
sns.distplot(titanic['fare'])
C:\Users\Rushi\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
Out[4]:
<AxesSubplot:xlabel='fare', ylabel='Density'>
Density
fare
In [5]:
sns.distplot(titanic['fare'], kde = False)
C:\Users\Rushi\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

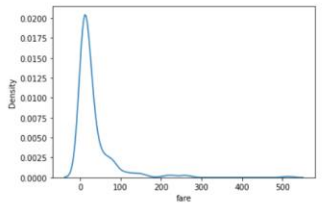
jupyter DSBDA_Lab_A8 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
fare
In [5]:
sns.distplot(titanic['fare'], kde = False)
C:\Users\Rushi\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
Out[5]:
<AxesSubplot:xlabel='fare'>
fare
In [6]:
sns.distplot(titanic['fare'], hist = False)
C:\Users\Rushi\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
warnings.warn(msg, FutureWarning)
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

fare

In [6]:
```
sns.distplot(titanic['fare'], hist = False)
```

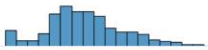
```
C:\Users\Rushi\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```

Out[6]: <AxesSubplot:xlabel='fare', ylabel='Density'>



In [7]:
```
sns.jointplot(x = 'age', y = 'fare', data = titanic)
```

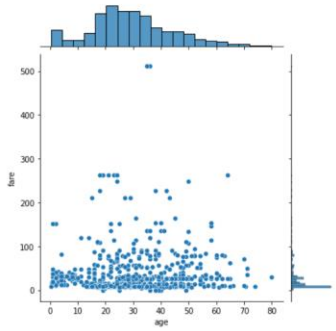
Out[7]: <seaborn.axisgrid.JointGrid at 0x17c6cf91ae0>



(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a ... | DSBDA_Lab_A8 - Jupyter Noteb...
localhost:8888/notebooks/DSBDA_Lab_A8.ipynb
Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | background.html | Daily use English se... | WhatsApp
jupyter DSBDA_Lab_A8 Last Checkpoint: 3 hours ago (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)
Run Markdown

In [7]:
```
sns.jointplot(x = 'age', y = 'fare', data = titanic)
```

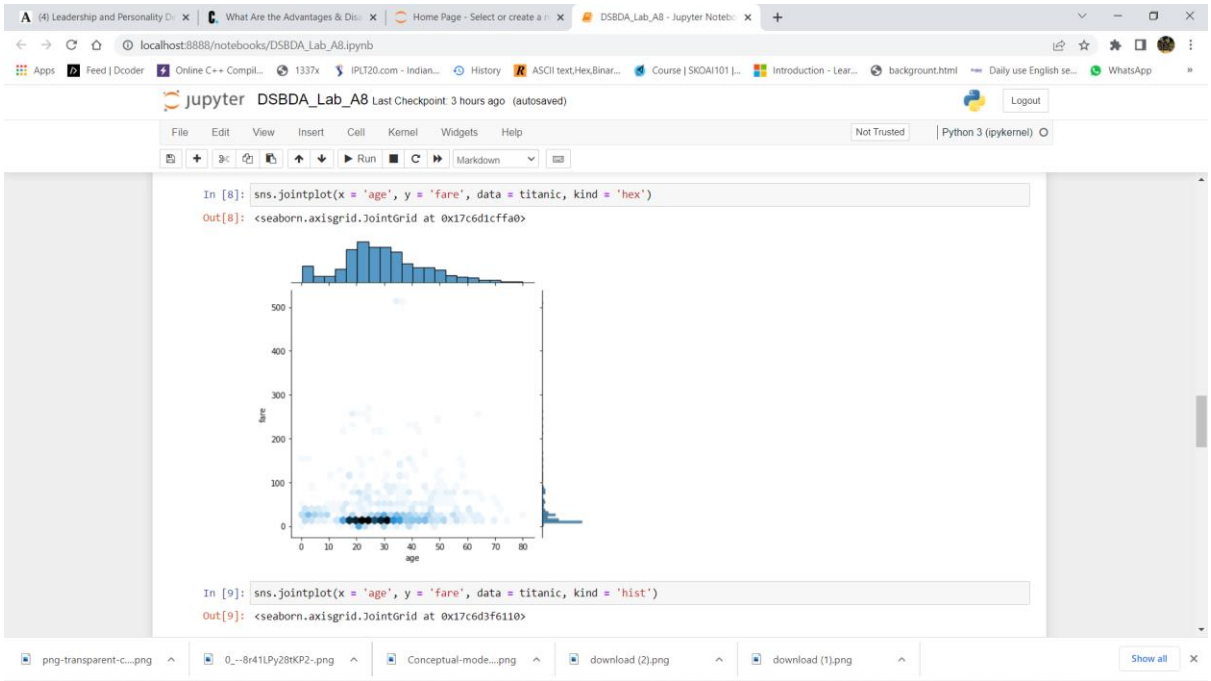
Out[7]: <seaborn.axisgrid.JointGrid at 0x17c6cf91ae0>

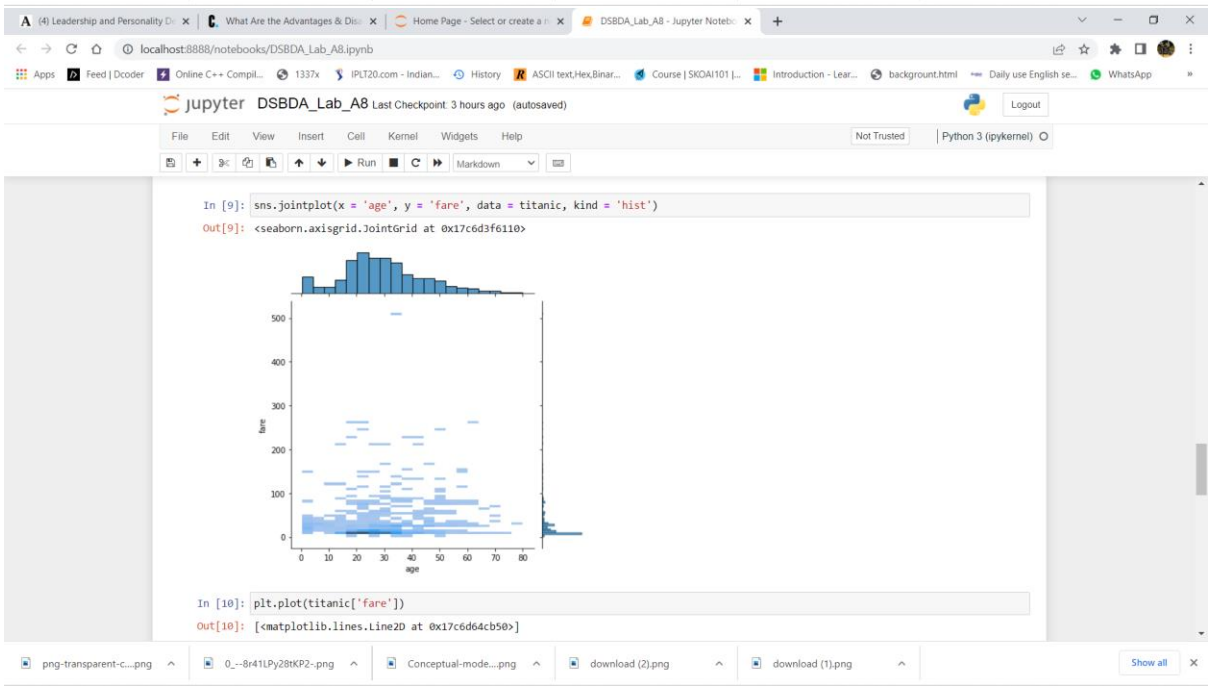


In [8]:
```
sns.jointplot(x = 'age', y = 'fare', data = titanic, kind = 'hex')
```

Out[8]: <seaborn.axisgrid.JointGrid at 0x17c6d1cffa0>

png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all


jupyter DSBDA_Lab_A8 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [8]: sns.jointplot(x = 'age', y = 'fare', data = titanic, kind = 'hex')
Out[8]: <seaborn.axisgrid.JointGrid at 0x17c6d1cffa0>
fare
age
In [9]: sns.jointplot(x = 'age', y = 'fare', data = titanic, kind = 'hist')
Out[9]: <seaborn.axisgrid.JointGrid at 0x17c6d3f6110>
Show all


jupyter DSBDA_Lab_A8 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [9]: sns.jointplot(x = 'age', y = 'fare', data = titanic, kind = 'hist')
Out[9]: <seaborn.axisgrid.JointGrid at 0x17c6d3f6110>
fare
age
In [10]: plt.plot(titanic['fare'])
Out[10]: [<matplotlib.lines.Line2D at 0x17c6d64cb50>]
Show all

(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a n... | DSBDA_Lab_A8 - Jupyter Noteb...

localhost:8888/notebooks/DSBDA_Lab_A8.ipynb

Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | background.html | Daily use English se... | WhatsApp

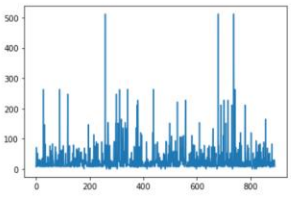
jupyter **DSBDA_Lab_A8** Last Checkpoint: 3 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3 (ipykernel)

Run | Markdown

In [10]:
```
plt.plot(titanic['fare'])
```

Out[10]:
```
[<matplotlib.lines.Line2D at 0x17c6d64cb50>]
```



In [11]:
```
plt.hist(titanic['fare'])
```

Out[11]:
```
(array([732., 106.,  31.,   2.,  11.,   6.,   0.,   0.,   0.,   3.]),
 array([  0.     ,  51.23292, 102.46584, 153.69876, 204.93168, 256.1646 ,
        307.39752, 358.63044, 409.86336, 461.09628, 512.3292 ]),
 <BarContainer object of 10 artists>)
```

png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a n... | DSBDA_Lab_A8 - Jupyter Noteb...

localhost:8888/notebooks/DSBDA_Lab_A8.ipynb

Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | background.html | Daily use English se... | WhatsApp

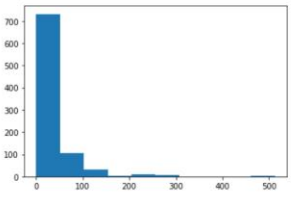
jupyter **DSBDA_Lab_A8** Last Checkpoint: 3 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3 (ipykernel)

Run | Markdown

In [11]:
```
plt.hist(titanic['fare'])
```

Out[11]:
```
(array([732., 106.,  31.,   2.,  11.,   6.,   0.,   0.,   0.,   3.]),
 array([  0.     ,  51.23292, 102.46584, 153.69876, 204.93168, 256.1646 ,
        307.39752, 358.63044, 409.86336, 461.09628, 512.3292 ]),
 <BarContainer object of 10 artists>)
```



In [12]:
```
sns.barplot(x="sex",y="age", data = titanic)
```

Out[12]:
```
<AxesSubplot:xlabel='sex', ylabel='age'>
```

png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

(4) Leadership and Personality D... | What Are the Advantages & Disa... | Home Page - Select or create a n... | DSBDA_Lab_A8 - Jupyter Noteb...

localhost:8888/notebooks/DSBDA_Lab_A8.ipynb

Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | background.html | Daily use English se... | WhatsApp

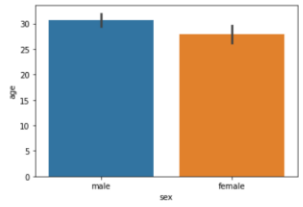
jupyter **DSBDA_Lab_A8** Last Checkpoint: 3 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Run | Markdown

In [12]:
```
sns.barplot(x="sex",y="age", data = titanic)
```

Out[12]: <AxesSubplot:xlabel='sex', ylabel='age'>



In [13]:
```
sns.countplot(x = 'sex', data=titanic)
```

Out[13]: <AxesSubplot:xlabel='sex', ylabel='count'>

png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

(4) Leadership and Personality D... | What Are the Advantages & Disa... | Home Page - Select or create a n... | DSBDA_Lab_A8 - Jupyter Noteb...

localhost:8888/notebooks/DSBDA_Lab_A8.ipynb

Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | background.html | Daily use English se... | WhatsApp

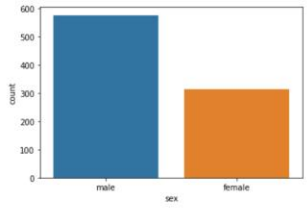
jupyter **DSBDA_Lab_A8** Last Checkpoint: 3 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Run | Markdown

In [13]:
```
sns.countplot(x = 'sex', data=titanic)
```

Out[13]: <AxesSubplot:xlabel='sex', ylabel='count'>



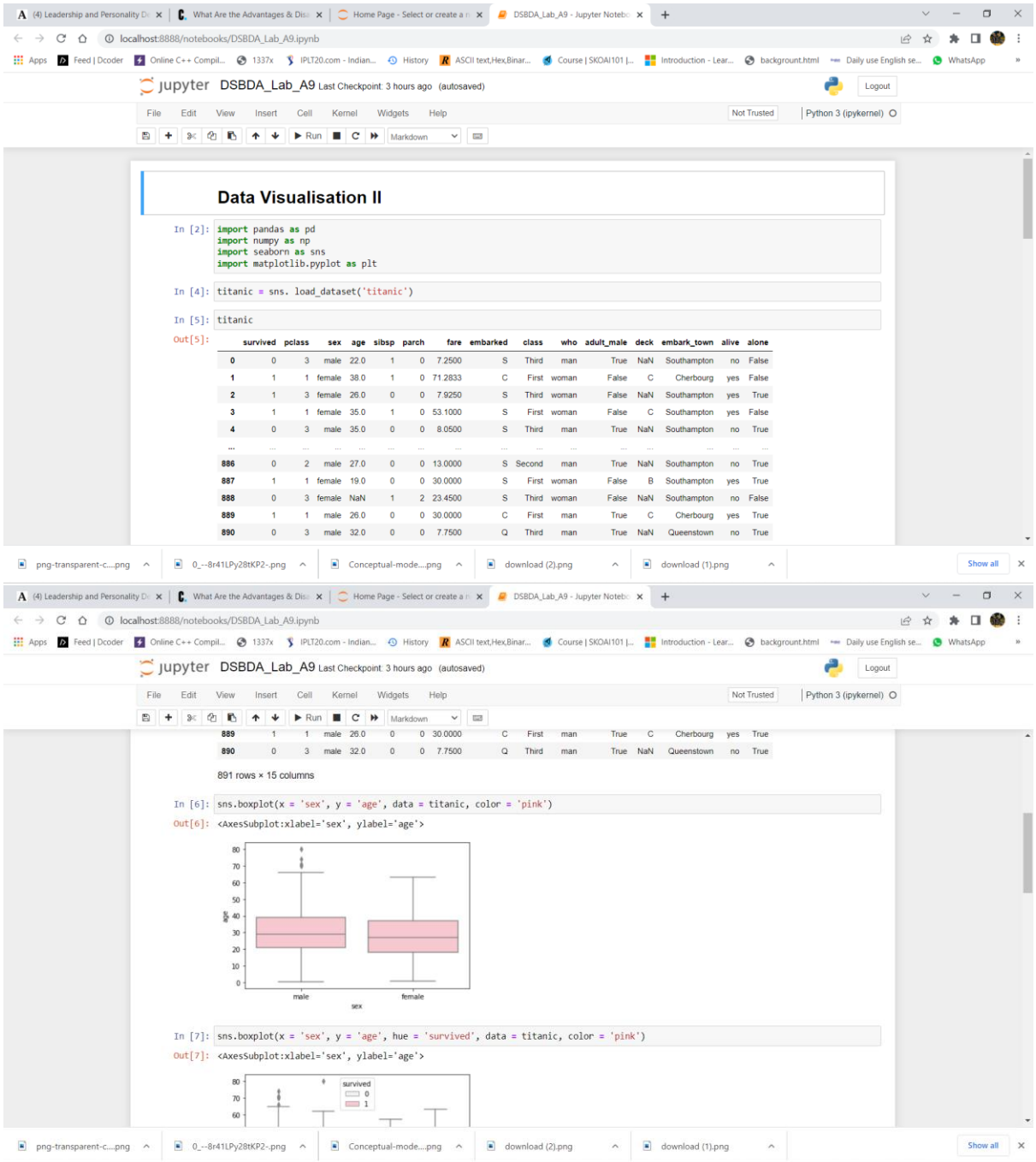
In [ ]:

png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

Data Science and Big Data Analytics

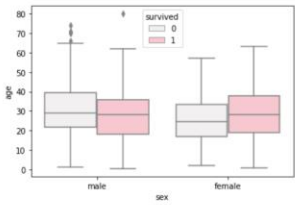
Group : A

Lab Assignment : 09



In [7]:
```
sns.boxplot(x = 'sex', y = 'age', hue = 'survived', data = titanic, color = 'pink')
```

Out[7]: <AxesSubplot:xlabel='sex', ylabel='age'>



male mar gaye 😂😂😂😂😂😂

In [8]:
```
df_copy = titanic
df_copy.groupby('sex')['survived'].value_counts()
```

Out[8]:
```
sex     survived
female  1           233
        0            81
```


(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a ... | DSBDA_Lab_A9 - Jupyter Noteb...
localhost:8888/notebooks/DSBDA_Lab_A9.ipynb
Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | backgrount.html | Daily use English se... | WhatsApp
jupyter DSBDA_Lab_A9 Last Checkpoint: 3 hours ago (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3 (ipykernel)
Run | Markdown

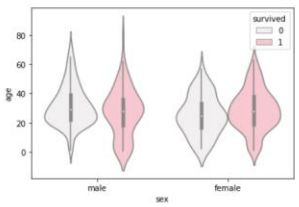
male mar gaye 😂😂😂😂😂😂

In [8]:
```
df_copy = titanic
df_copy.groupby('sex')['survived'].value_counts()
```

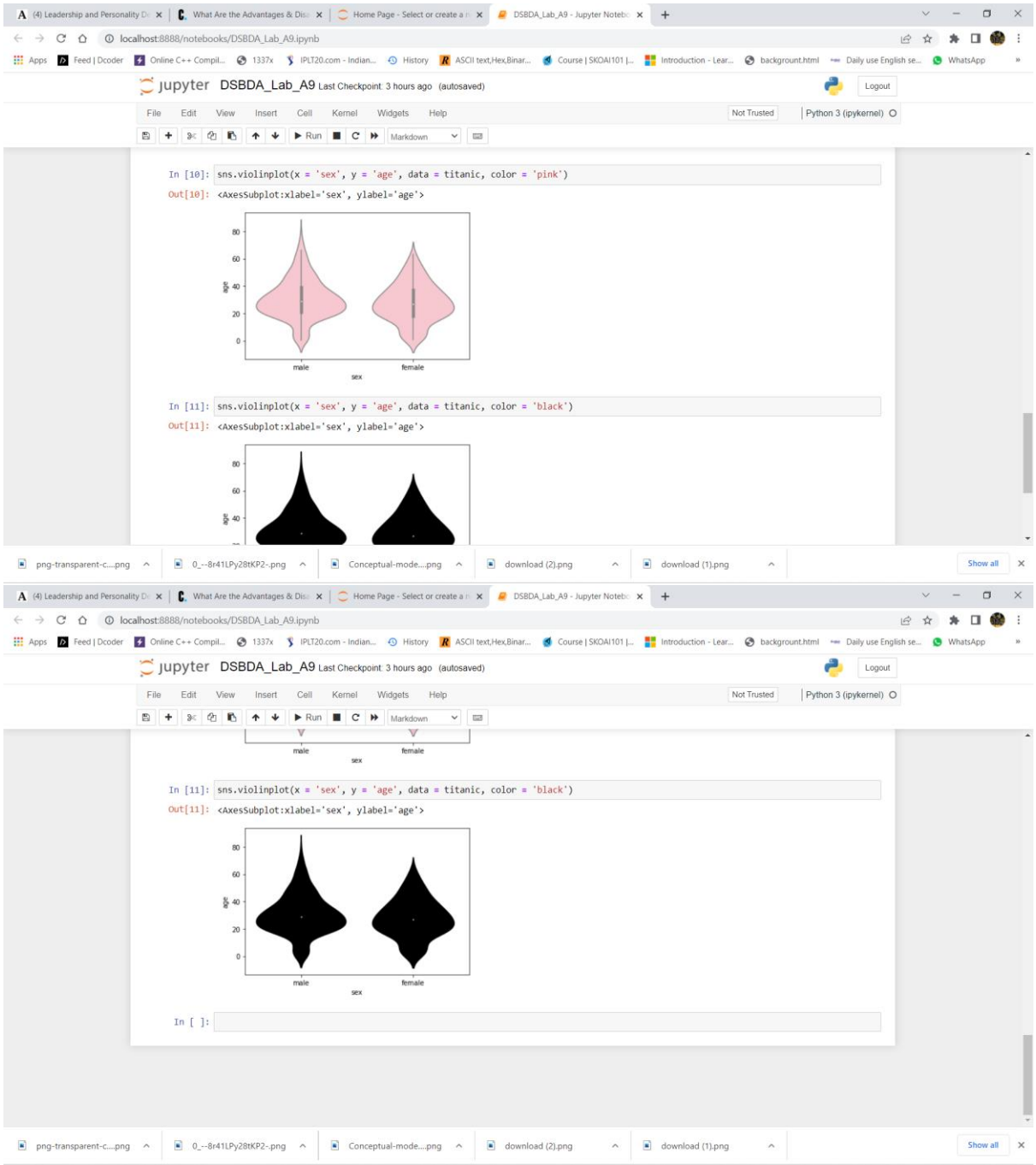
Out[8]:
```
sex     survived
female  1           233
        0            81
male    0           468
        1           109
Name: survived, dtype: int64
```

In [9]:
```
sns.violinplot(x = 'sex', y = 'age', hue = 'survived', data = titanic, color = 'pink')
```

Out[9]: <AxesSubplot:xlabel='sex', ylabel='age'>



png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

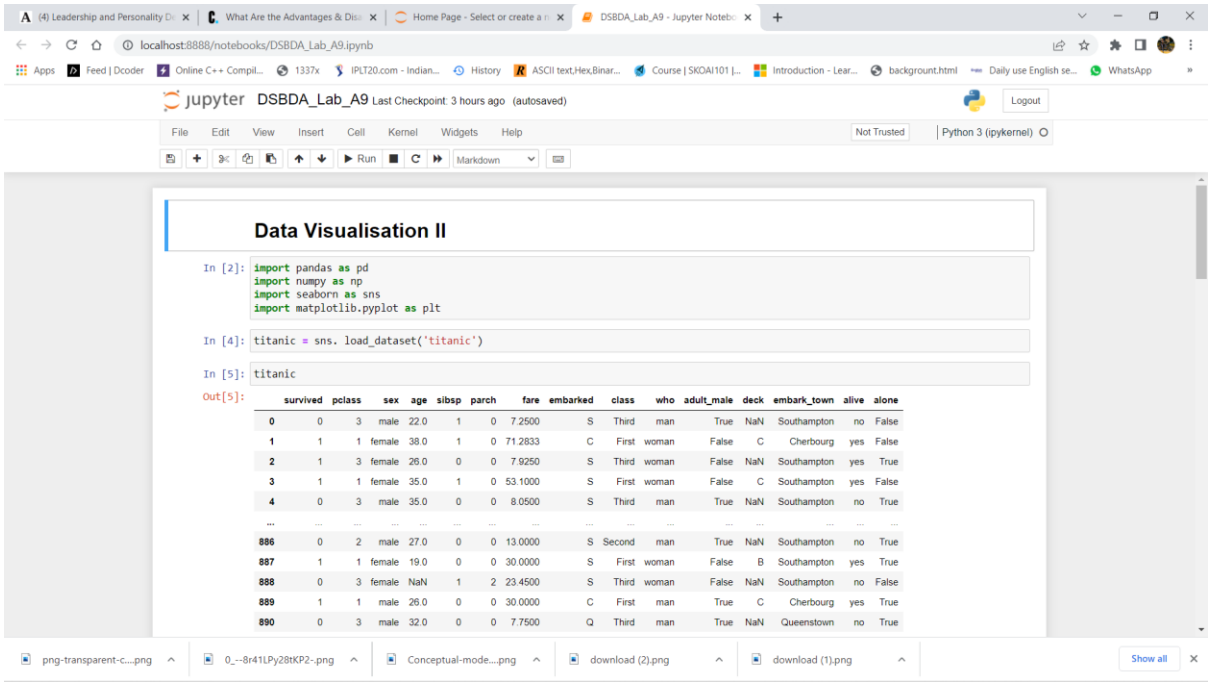

jupyter DSBDA_Lab_A9 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [10]: sns.violinplot(x = 'sex', y = 'age', data = titanic, color = 'pink')
Out[10]: <AxesSubplot:xlabel='sex', ylabel='age'>
age
sex
male
female
In [11]: sns.violinplot(x = 'sex', y = 'age', data = titanic, color = 'black')
Out[11]: <AxesSubplot:xlabel='sex', ylabel='age'>
Show all

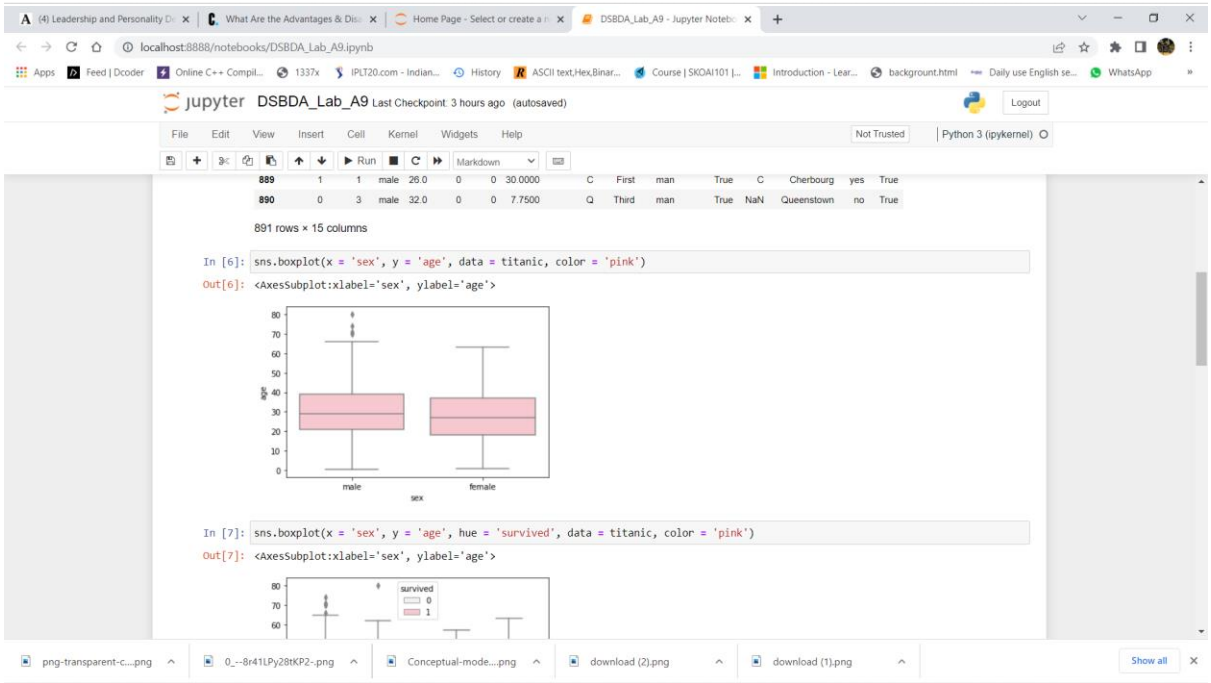
jupyter DSBDA_Lab_A9 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [11]: sns.violinplot(x = 'sex', y = 'age', data = titanic, color = 'black')
Out[11]: <AxesSubplot:xlabel='sex', ylabel='age'>
age
sex
male
female
In [ ]:
Show all

Data Science and Big Data Analytics

Group : A

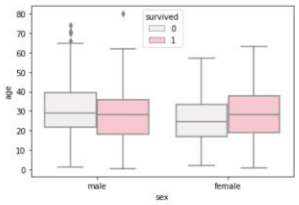
Lab Assignment : 10





In [7]:
```
sns.boxplot(x = 'sex', y = 'age', hue = 'survived', data = titanic, color = 'pink')
```

Out[7]: <AxesSubplot:xlabel='sex', ylabel='age'>



male mar gaye 😂😂😂😂😂😂

In [8]:
```
df_copy = titanic
df_copy.groupby('sex')['survived'].value_counts()
```

Out[8]:
```
sex     survived
female  1           233
        0            81
```


(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a ... | DSBDA_Lab_A9 - Jupyter Noteb...
localhost:8888/notebooks/DSBDA_Lab_A9.ipynb
Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | backgrount.html | Daily use English se... | WhatsApp
jupyter DSBDA_Lab_A9 Last Checkpoint: 3 hours ago (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3 (ipykernel)
Run | Markdown

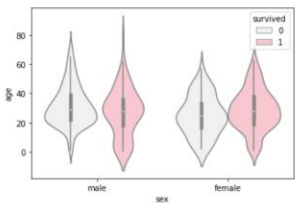
male mar gaye 😂😂😂😂😂😂

In [8]:
```
df_copy = titanic
df_copy.groupby('sex')['survived'].value_counts()
```

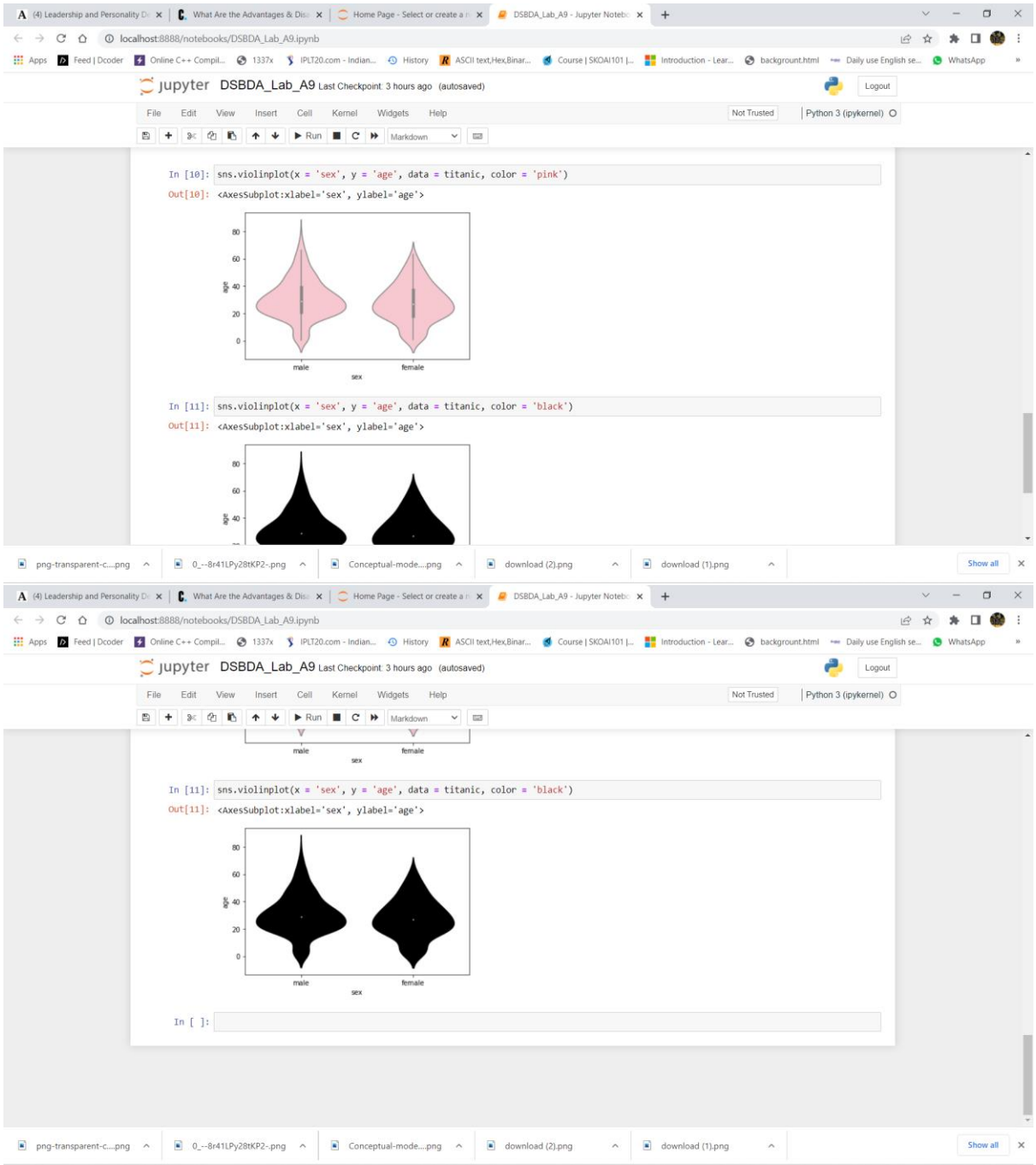
Out[8]:
```
sex     survived
female  1           233
        0            81
male    0           468
        1           109
Name: survived, dtype: int64
```

In [9]:
```
sns.violinplot(x = 'sex', y = 'age', hue = 'survived', data = titanic, color = 'pink')
```

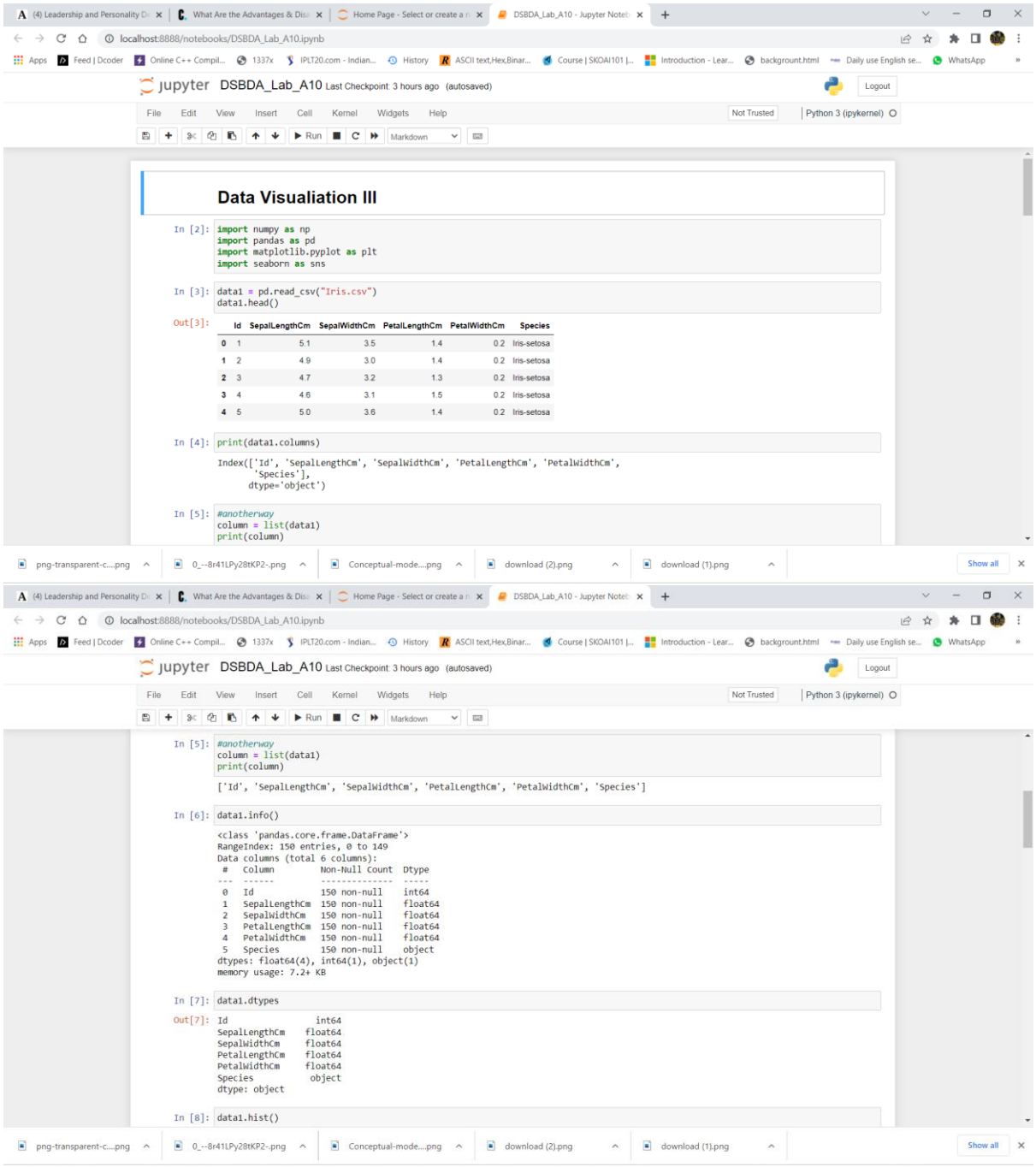
Out[9]: <AxesSubplot:xlabel='sex', ylabel='age'>



png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all


jupyter DSBDA_Lab_A9 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [10]: sns.violinplot(x = 'sex', y = 'age', data = titanic, color = 'pink')
Out[10]: <AxesSubplot:xlabel='sex', ylabel='age'>
age
sex
male
female
In [11]: sns.violinplot(x = 'sex', y = 'age', data = titanic, color = 'black')
Out[11]: <AxesSubplot:xlabel='sex', ylabel='age'>
Show all

jupyter DSBDA_Lab_A9 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [11]: sns.violinplot(x = 'sex', y = 'age', data = titanic, color = 'black')
Out[11]: <AxesSubplot:xlabel='sex', ylabel='age'>
age
sex
male
female
In [ ]:
Show all


DSBDA_Lab_A10 - Jupyter Notebook
localhost:8888/notebooks/DSBDA_Lab_A10.ipynb
jupyter DSBDA_Lab_A10 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
Data Visualiation III
In [2]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
In [3]:
data1 = pd.read_csv("Iris.csv")
data1.head()
Out[3]:
Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Species
0 1 5.1 3.5 1.4 0.2 Iris-setosa
1 2 4.9 3.0 1.4 0.2 Iris-setosa
2 3 4.7 3.2 1.3 0.2 Iris-setosa
3 4 4.6 3.1 1.5 0.2 Iris-setosa
4 5 5.0 3.6 1.4 0.2 Iris-setosa
In [4]:
print(data1.columns)
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
'Species'],
dtype='object')
In [5]:
#anotherway
column = list(data1)
print(column)
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

DSBDA_Lab_A10 - Jupyter Notebook
localhost:8888/notebooks/DSBDA_Lab_A10.ipynb
jupyter DSBDA_Lab_A10 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [5]:
#anotherway
column = list(data1)
print(column)
['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm', 'Species']
In [6]:
data1.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 Id 150 non-null int64
1 SepalLengthCm 150 non-null float64
2 SepalWidthCm 150 non-null float64
3 PetalLengthCm 150 non-null float64
4 PetalWidthCm 150 non-null float64
5 Species 150 non-null object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
In [7]:
data1.dtypes
Out[7]:
Id int64
SepalLengthCm float64
SepalWidthCm float64
PetalLengthCm float64
PetalWidthCm float64
Species object
dtype: object
In [8]:
data1.hist()
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a n... | DSBDA_Lab_A10 - Jupyter Noteb...

localhost:8888/notebooks/DSBDA_Lab_A10.ipynb

Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | background.html | Daily use English se... | WhatsApp

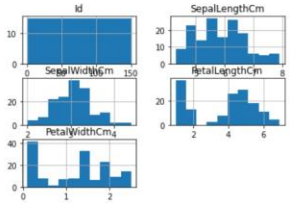
jupyter **DSBDA_Lab_A10** Last Checkpoint: 3 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Run | Markdown

In [8]:
```
data1.hist()
```

Out[8]:
```
array([[<AxesSubplot:title={'center':'Id'}>,
        <AxesSubplot:title={'center':'SepalLengthCm'}>],
       [<AxesSubplot:title={'center':'SepalWidthCm'}>,
        <AxesSubplot:title={'center':'PetalLengthCm'}>],
       [<AxesSubplot:title={'center':'PetalWidthCm'}>, <AxesSubplot:>]],
      dtype=object)
```



In [9]:
```
fig, axes = plt.subplots(2, 2, figsize=(16, 8))

axes[0,0].set_title("Distribution of First Column")
axes[0,0].hist(data1["SepalLengthCm"]);

axes[0,1].set_title("Distribution of Second Column")
axes[0,1].hist(data1["SepalWidthCm"]);
```

png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a n... | DSBDA_Lab_A10 - Jupyter Noteb...

localhost:8888/notebooks/DSBDA_Lab_A10.ipynb

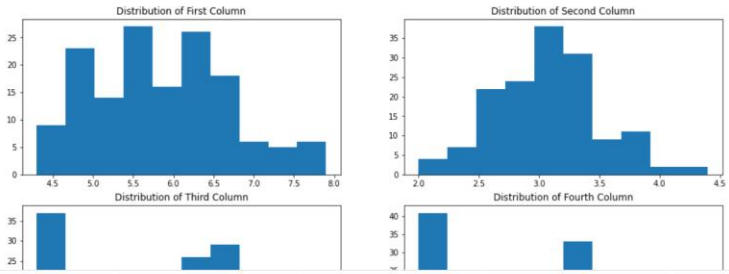
Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | background.html | Daily use English se... | WhatsApp

jupyter **DSBDA_Lab_A10** Last Checkpoint: 3 hours ago (autosaved) Logout

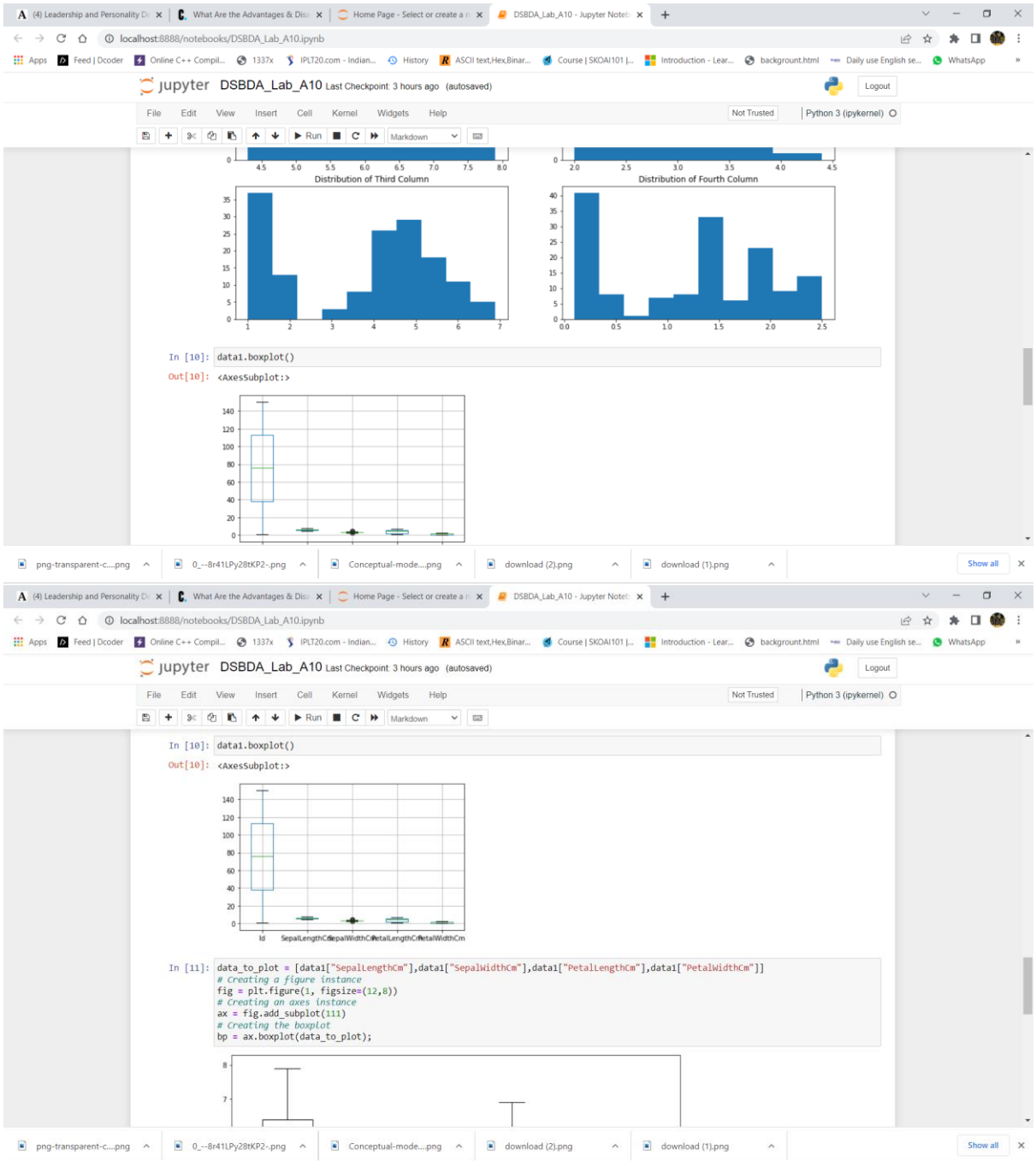
File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Run | Markdown

In [9]:
```
fig, axes = plt.subplots(2, 2, figsize=(16, 8))

axes[0,0].set_title("Distribution of First Column")
axes[0,0].hist(data1["SepalLengthCm"]);

axes[0,1].set_title("Distribution of Second Column")
axes[0,1].hist(data1["SepalWidthCm"]);

axes[1,0].set_title("Distribution of Third Column")
axes[1,0].hist(data1["PetalLengthCm"]);

axes[1,1].set_title("Distribution of Fourth Column")
axes[1,1].hist(data1["PetalWidthCm"]);
```



png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all


jupyter DSBDA_Lab_A10 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
Distribution of Third Column
Distribution of Fourth Column
In [10]: data1.boxplot()
Out[10]: <AxesSubplot:>
In [11]: data_to_plot = [data1["SepalLengthCm"],data1["SepalWidthCm"],data1["PetalLengthCm"],data1["PetalWidthCm"]]
# Creating a figure instance
fig = plt.figure(1, figsize=(12,8))
# Creating an axes instance
ax = fig.add_subplot(111)
# Creating the boxplot
bp = ax.boxplot(data_to_plot);
Id
SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

localhost:8888/notebooks/DSBDA_Lab_A10.ipynb

jupyter **DSBDA_Lab_A10** Last Checkpoint: 3 hours ago (autosaved)

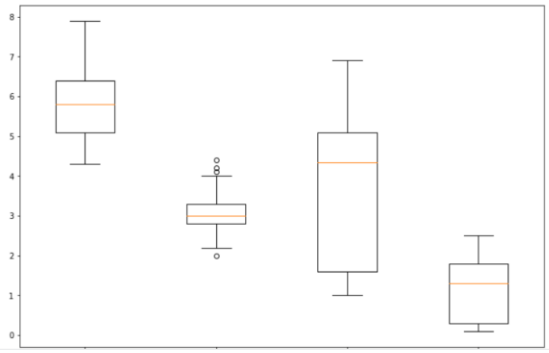
Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run | Markdown

In [11]:
```
data_to_plot = [data1["SepalLengthCm"],data1["SepalWidthCm"],data1["PetalLengthCm"],data1["PetalWidthCm"]]
# Creating a figure instance
fig = plt.figure(1, figsize=(12,8))
# Creating an axes instance
ax = fig.add_subplot(111)
# Creating the boxplot
bp = ax.boxplot(data_to_plot);
```



png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

localhost:8888/notebooks/DSBDA_Lab_A10.ipynb

jupyter **DSBDA_Lab_A10** Last Checkpoint: 3 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

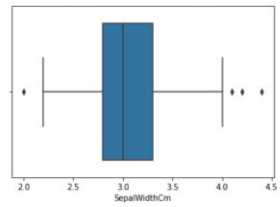
Not Trusted | Python 3 (ipykernel)

Run | Markdown

In [12]:
```
sns.boxplot(data1['SepalWidthCm'])
```

```
C:\Users\Rushi\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[12]: <AxesSubplot:xlabel='SepalWidthCm'>



In [13]:
```
print(np.where(data1['SepalWidthCm']>4.0))
```

```
(array([15, 32, 33], dtype=int64),)
```

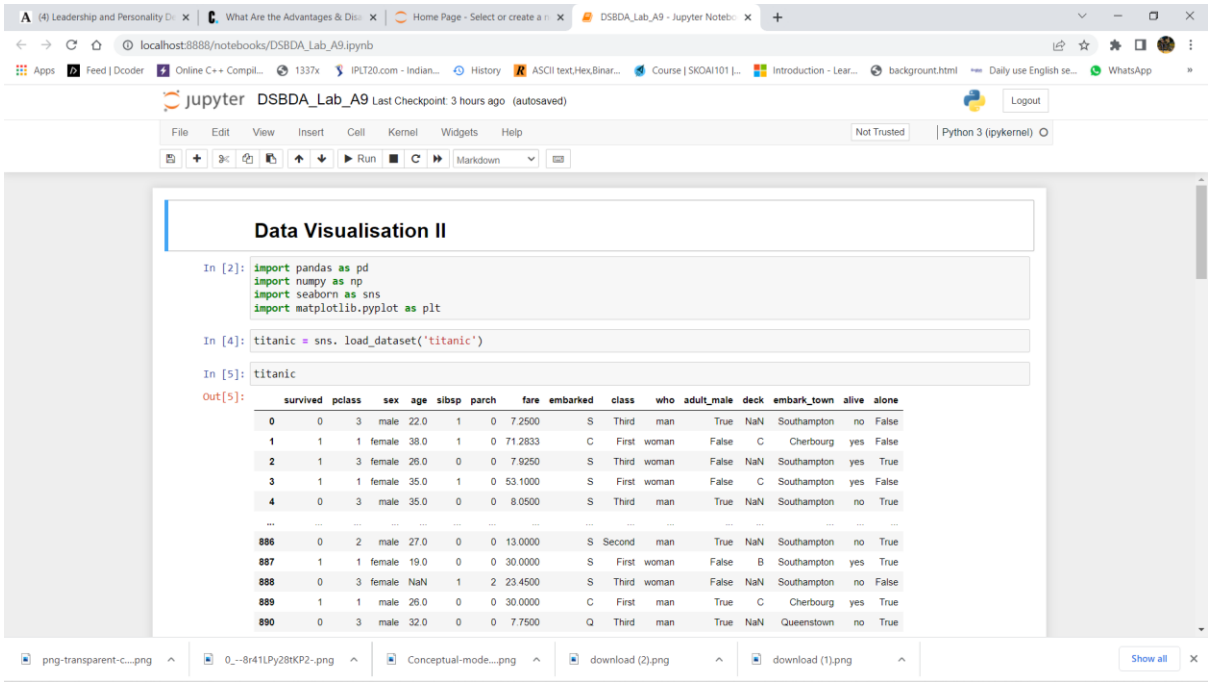
In [ ]:

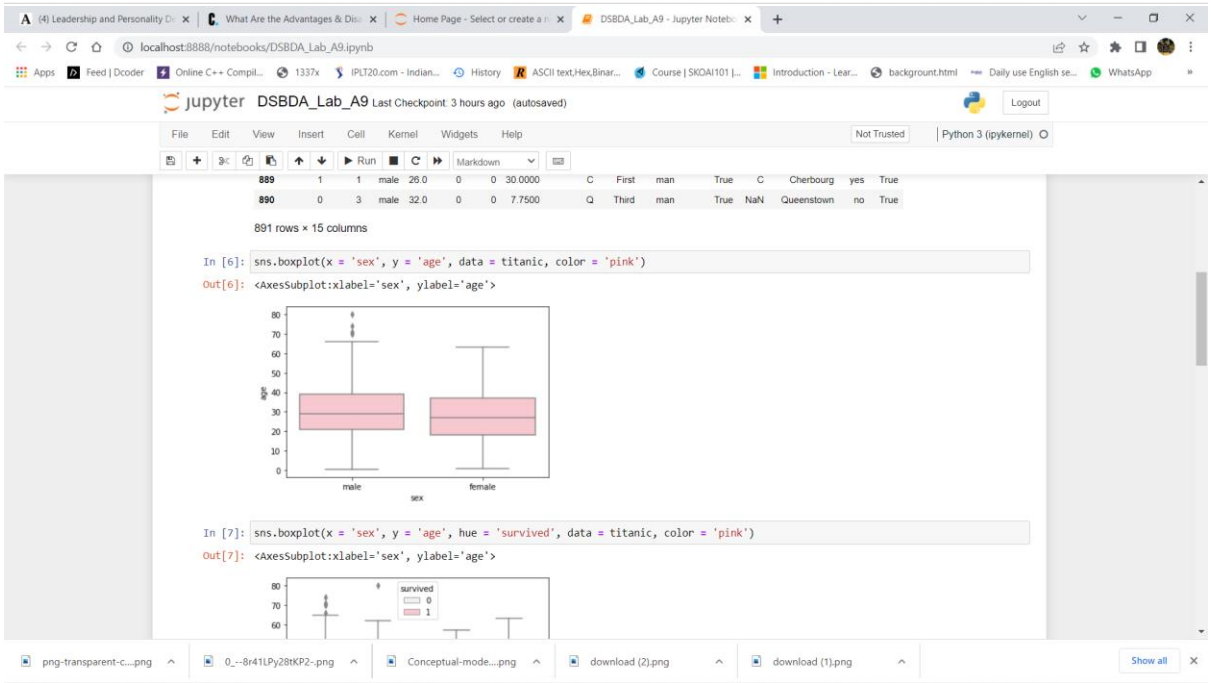
png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

Data Science and Big Data Analytics

Group : B2

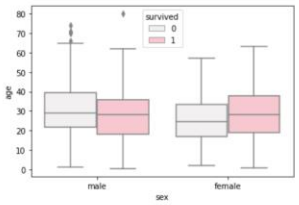
Lab Assignment : 12





In [7]:
```
sns.boxplot(x = 'sex', y = 'age', hue = 'survived', data = titanic, color = 'pink')
```

Out[7]: <AxesSubplot:xlabel='sex', ylabel='age'>



male mar gaye 😂😂😂😂😂😂

In [8]:
```
df_copy = titanic
df_copy.groupby('sex')['survived'].value_counts()
```

Out[8]:
```
sex     survived
female  1           233
        0            81
```


(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a ... | DSBDA_Lab_A9 - Jupyter Noteb...
localhost:8888/notebooks/DSBDA_Lab_A9.ipynb
Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | backgrount.html | Daily use English se... | WhatsApp
jupyter DSBDA_Lab_A9 Last Checkpoint: 3 hours ago (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3 (ipykernel)
Run | Markdown

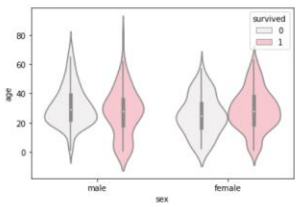
male mar gaye 😂😂😂😂😂😂

In [8]:
```
df_copy = titanic
df_copy.groupby('sex')['survived'].value_counts()
```

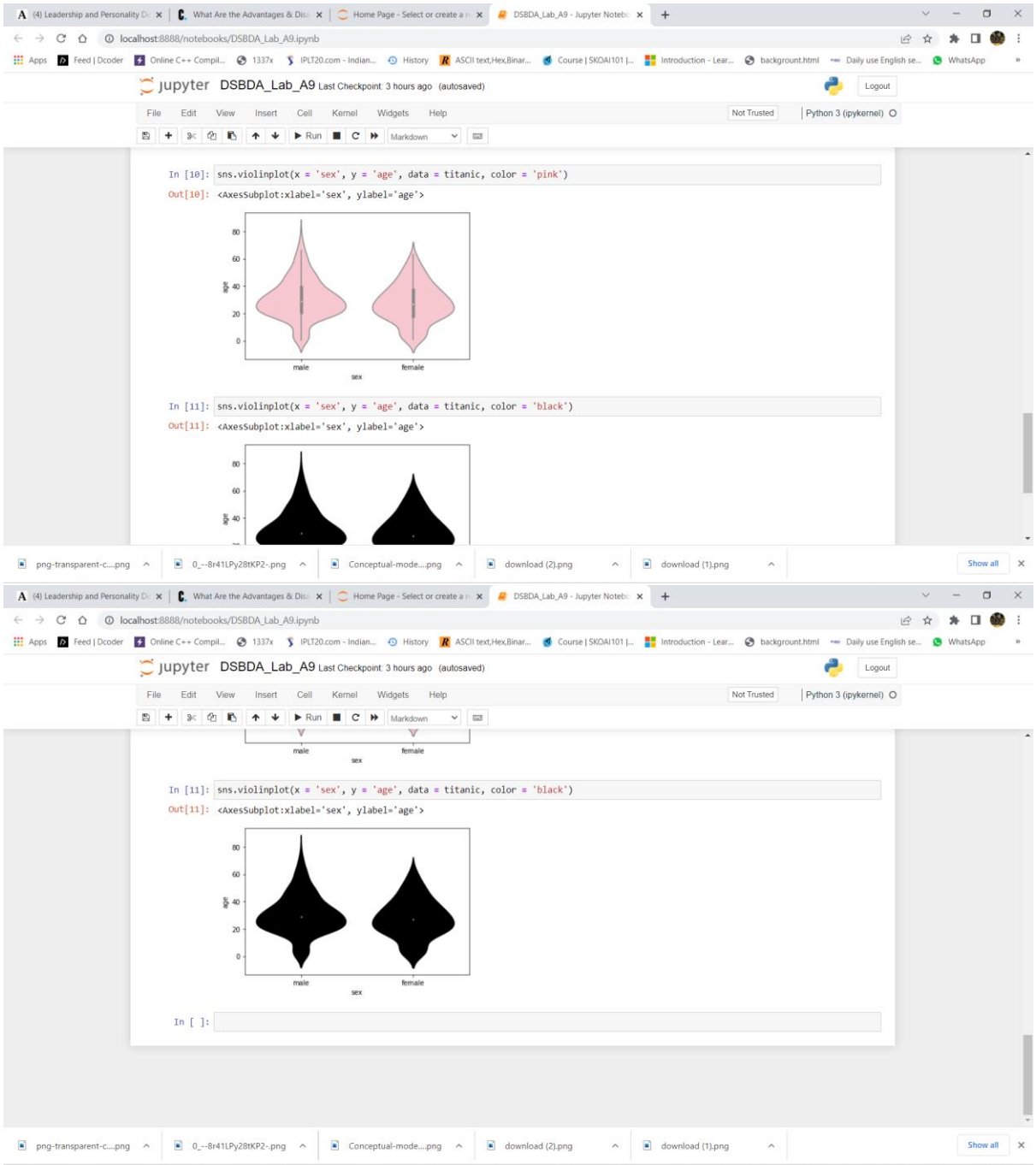
Out[8]:
```
sex     survived
female  1           233
        0            81
male    0           468
        1           109
Name: survived, dtype: int64
```

In [9]:
```
sns.violinplot(x = 'sex', y = 'age', hue = 'survived', data = titanic, color = 'pink')
```

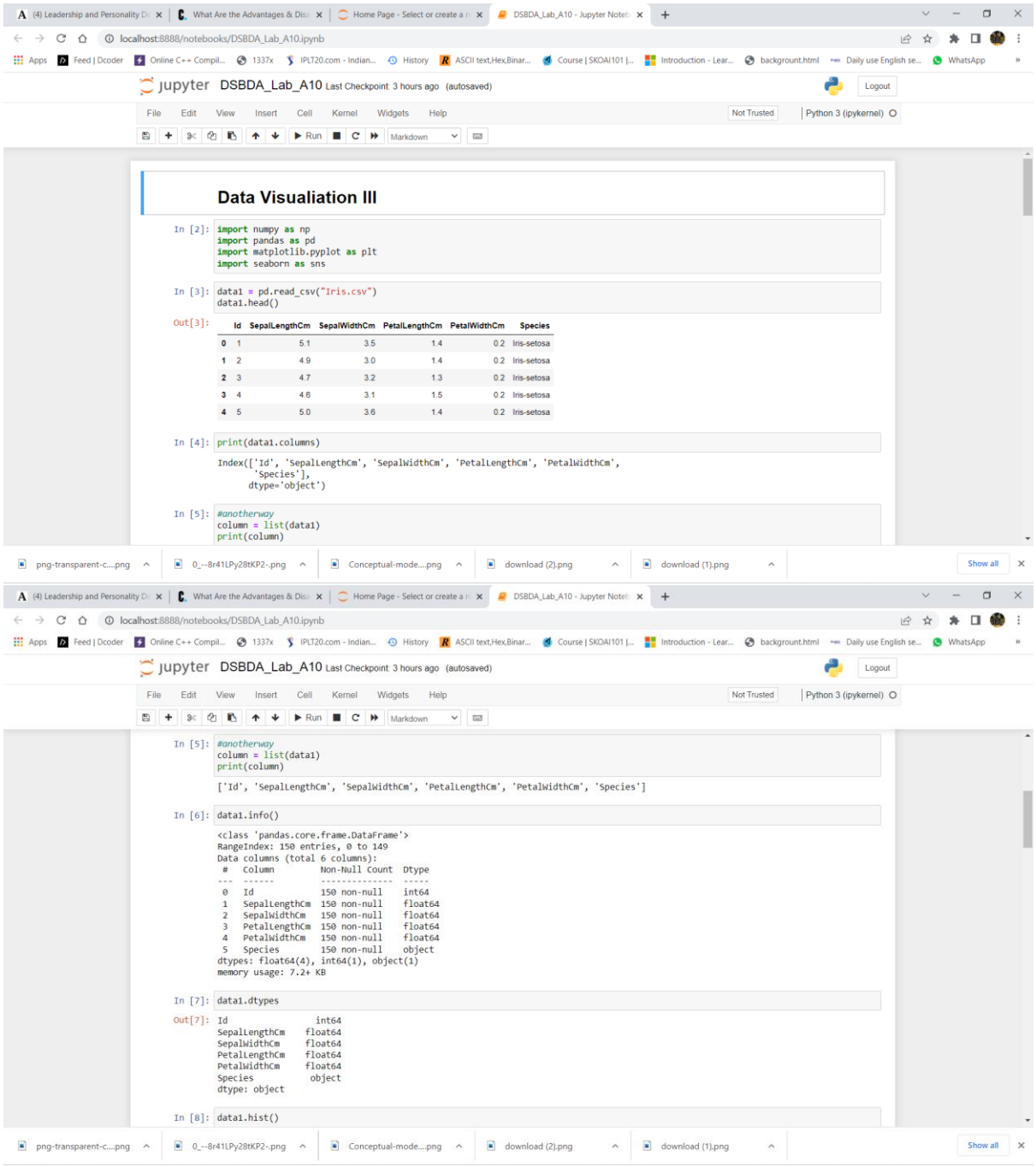
Out[9]: <AxesSubplot:xlabel='sex', ylabel='age'>



png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all


jupyter DSBDA_Lab_A9 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [10]: sns.violinplot(x = 'sex', y = 'age', data = titanic, color = 'pink')
Out[10]: <AxesSubplot:xlabel='sex', ylabel='age'>
age
sex
male
female
In [11]: sns.violinplot(x = 'sex', y = 'age', data = titanic, color = 'black')
Out[11]: <AxesSubplot:xlabel='sex', ylabel='age'>
Show all

jupyter DSBDA_Lab_A9 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [11]: sns.violinplot(x = 'sex', y = 'age', data = titanic, color = 'black')
Out[11]: <AxesSubplot:xlabel='sex', ylabel='age'>
age
sex
male
female
In [ ]:
Show all


DSBDA_Lab_A10 - Jupyter Notebook
localhost:8888/notebooks/DSBDA_Lab_A10.ipynb
jupyter DSBDA_Lab_A10 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
Data Visualiation III
In [2]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
In [3]:
data1 = pd.read_csv("Iris.csv")
data1.head()
Out[3]:
Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Species
0 1 5.1 3.5 1.4 0.2 Iris-setosa
1 2 4.9 3.0 1.4 0.2 Iris-setosa
2 3 4.7 3.2 1.3 0.2 Iris-setosa
3 4 4.6 3.1 1.5 0.2 Iris-setosa
4 5 5.0 3.6 1.4 0.2 Iris-setosa
In [4]:
print(data1.columns)
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
'Species'],
dtype='object')
In [5]:
#anotherway
column = list(data1)
print(column)
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

DSBDA_Lab_A10 - Jupyter Notebook
localhost:8888/notebooks/DSBDA_Lab_A10.ipynb
jupyter DSBDA_Lab_A10 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [5]:
#anotherway
column = list(data1)
print(column)
['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm', 'Species']
In [6]:
data1.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 Id 150 non-null int64
1 SepalLengthCm 150 non-null float64
2 SepalWidthCm 150 non-null float64
3 PetalLengthCm 150 non-null float64
4 PetalWidthCm 150 non-null float64
5 Species 150 non-null object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
In [7]:
data1.dtypes
Out[7]:
Id int64
SepalLengthCm float64
SepalWidthCm float64
PetalLengthCm float64
PetalWidthCm float64
Species object
dtype: object
In [8]:
data1.hist()
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a n... | DSBDA_Lab_A10 - Jupyter Noteb...

localhost:8888/notebooks/DSBDA_Lab_A10.ipynb

Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | background.html | Daily use English se... | WhatsApp

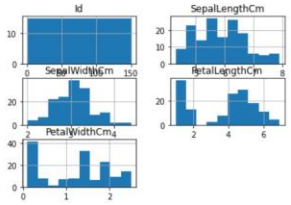
jupyter **DSBDA_Lab_A10** Last Checkpoint: 3 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Run | Markdown

In [8]:
```
data1.hist()
```

Out[8]:
```
array([[<AxesSubplot:title={'center':'Id'}>,
        <AxesSubplot:title={'center':'SepalLengthCm'}>],
       [<AxesSubplot:title={'center':'SepalWidthCm'}>,
        <AxesSubplot:title={'center':'PetalLengthCm'}>],
       [<AxesSubplot:title={'center':'PetalWidthCm'}>, <AxesSubplot:>]],
      dtype=object)
```



In [9]:
```
fig, axes = plt.subplots(2, 2, figsize=(16, 8))

axes[0,0].set_title("Distribution of First Column")
axes[0,0].hist(data1["SepalLengthCm"]);

axes[0,1].set_title("Distribution of Second Column")
axes[0,1].hist(data1["SepalWidthCm"]);
```

png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

(4) Leadership and Personality D... | What Are the Advantages & Dis... | Home Page - Select or create a n... | DSBDA_Lab_A10 - Jupyter Noteb...

localhost:8888/notebooks/DSBDA_Lab_A10.ipynb

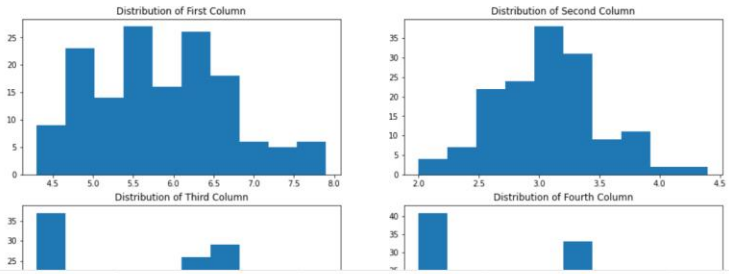
Apps | Feed | Dcoder | Online C++ Compil... | 1337x | IPLT20.com - Indian... | History | ASCII text,Hex,Binar... | Course | SKOAI101 |... | Introduction - Lear... | background.html | Daily use English se... | WhatsApp

jupyter **DSBDA_Lab_A10** Last Checkpoint: 3 hours ago (autosaved) Logout

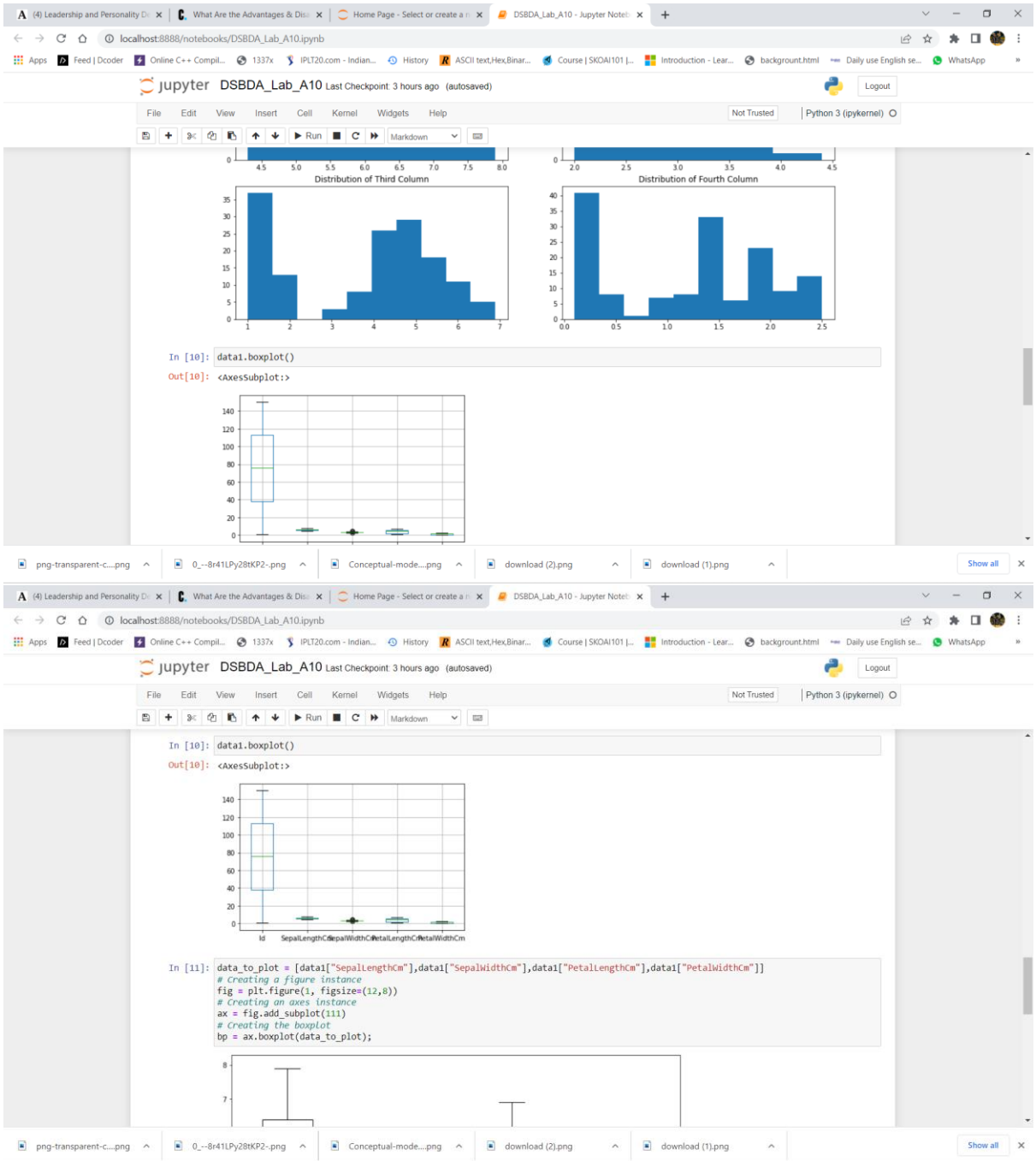
File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Run | Markdown

In [9]:
```
fig, axes = plt.subplots(2, 2, figsize=(16, 8))

axes[0,0].set_title("Distribution of First Column")
axes[0,0].hist(data1["SepalLengthCm"]);

axes[0,1].set_title("Distribution of Second Column")
axes[0,1].hist(data1["SepalWidthCm"]);

axes[1,0].set_title("Distribution of Third Column")
axes[1,0].hist(data1["PetalLengthCm"]);

axes[1,1].set_title("Distribution of Fourth Column")
axes[1,1].hist(data1["PetalWidthCm"]);
```



png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all


jupyter DSBDA_Lab_A10 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
Distribution of Third Column
Distribution of Fourth Column
In [10]: data1.boxplot()
Out[10]: <AxesSubplot:>
In [11]: data_to_plot = [data1["SepalLengthCm"],data1["SepalWidthCm"],data1["PetalLengthCm"],data1["PetalWidthCm"]]
# Creating a figure instance
fig = plt.figure(1, figsize=(12,8))
# Creating an axes instance
ax = fig.add_subplot(111)
# Creating the boxplot
bp = ax.boxplot(data_to_plot);
Id
SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

localhost:8888/notebooks/DSBDA_Lab_A10.ipynb

jupyter **DSBDA_Lab_A10** Last Checkpoint: 3 hours ago (autosaved)

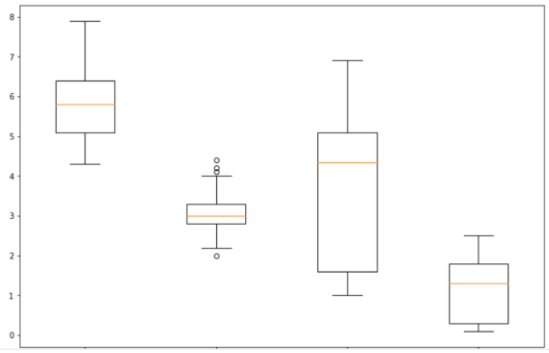
Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run | Markdown

In [11]:
```
data_to_plot = [data1["SepalLengthCm"],data1["SepalWidthCm"],data1["PetalLengthCm"],data1["PetalWidthCm"]]
# Creating a figure instance
fig = plt.figure(1, figsize=(12,8))
# Creating an axes instance
ax = fig.add_subplot(111)
# Creating the boxplot
bp = ax.boxplot(data_to_plot);
```



png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all

localhost:8888/notebooks/DSBDA_Lab_A10.ipynb

jupyter **DSBDA_Lab_A10** Last Checkpoint: 3 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

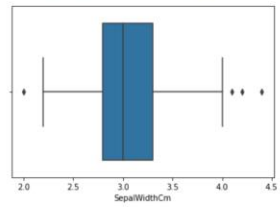
Not Trusted | Python 3 (ipykernel)

Run | Markdown

In [12]:
```
sns.boxplot(data1['SepalWidthCm'])
```

```
C:\Users\Rushi\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[12]: <AxesSubplot:xlabel='SepalWidthCm'>

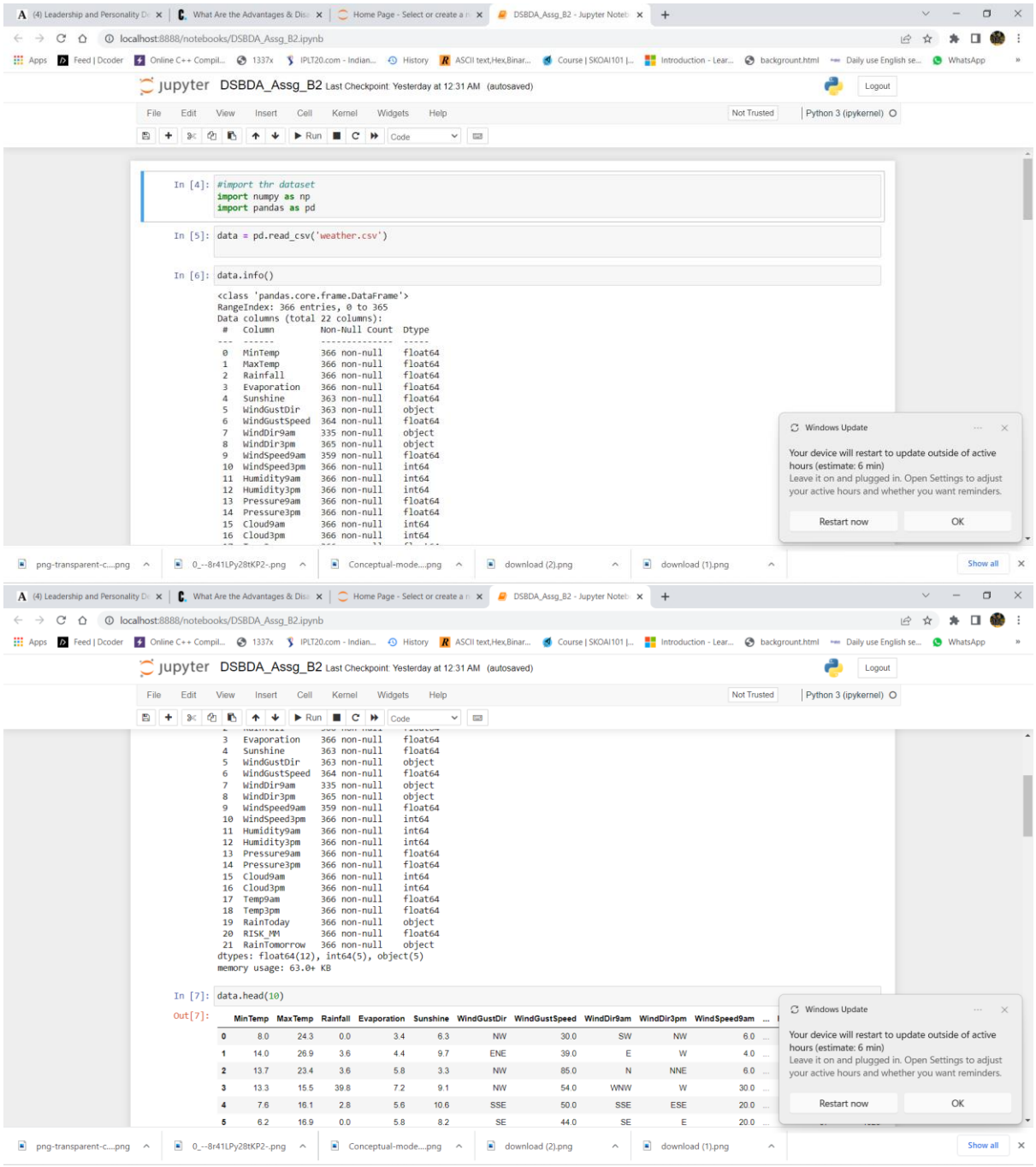


In [13]:
```
print(np.where(data1['SepalWidthCm']>4.0))
```

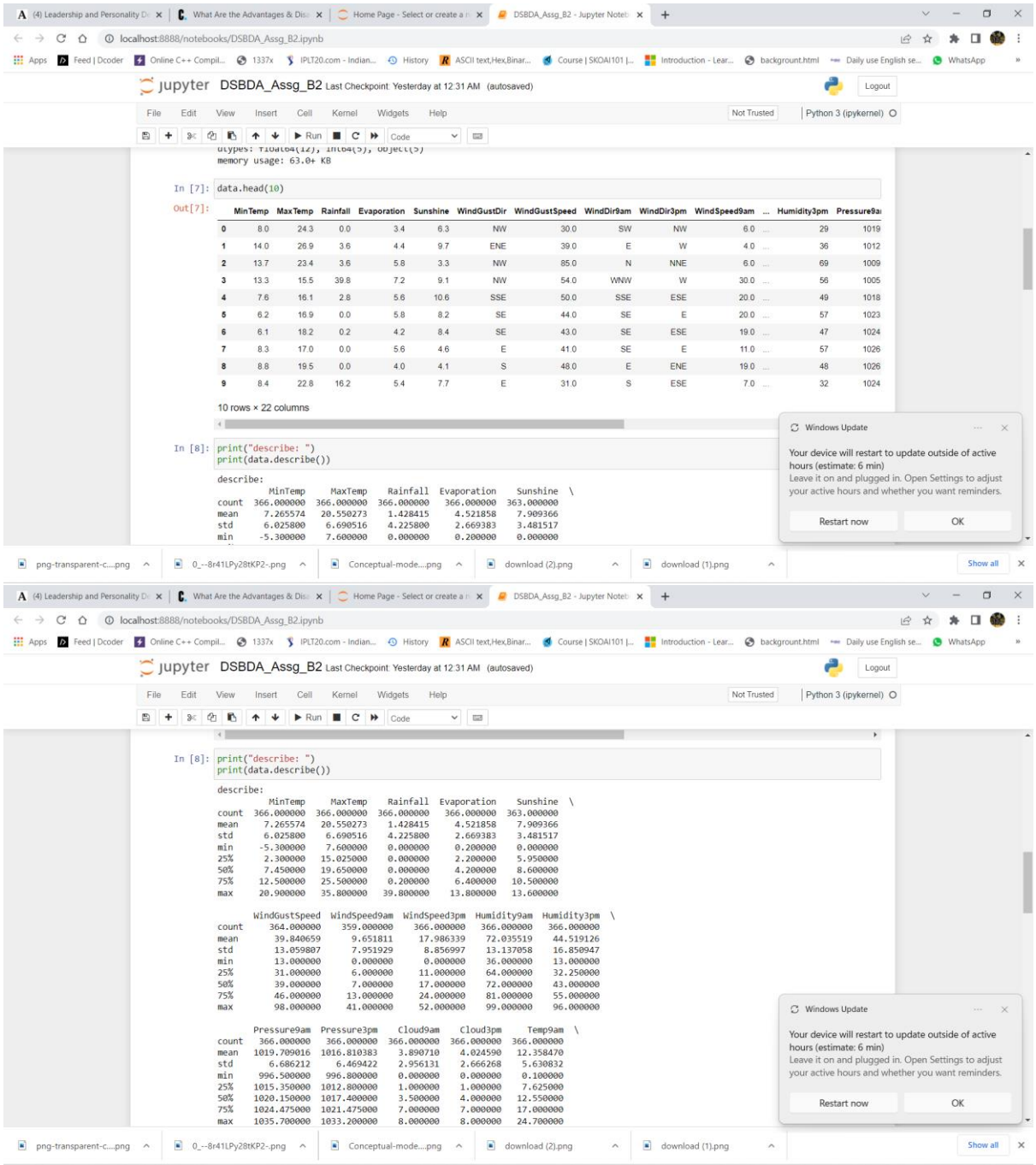
```
(array([15, 32, 33], dtype=int64),)
```

In [ ]:

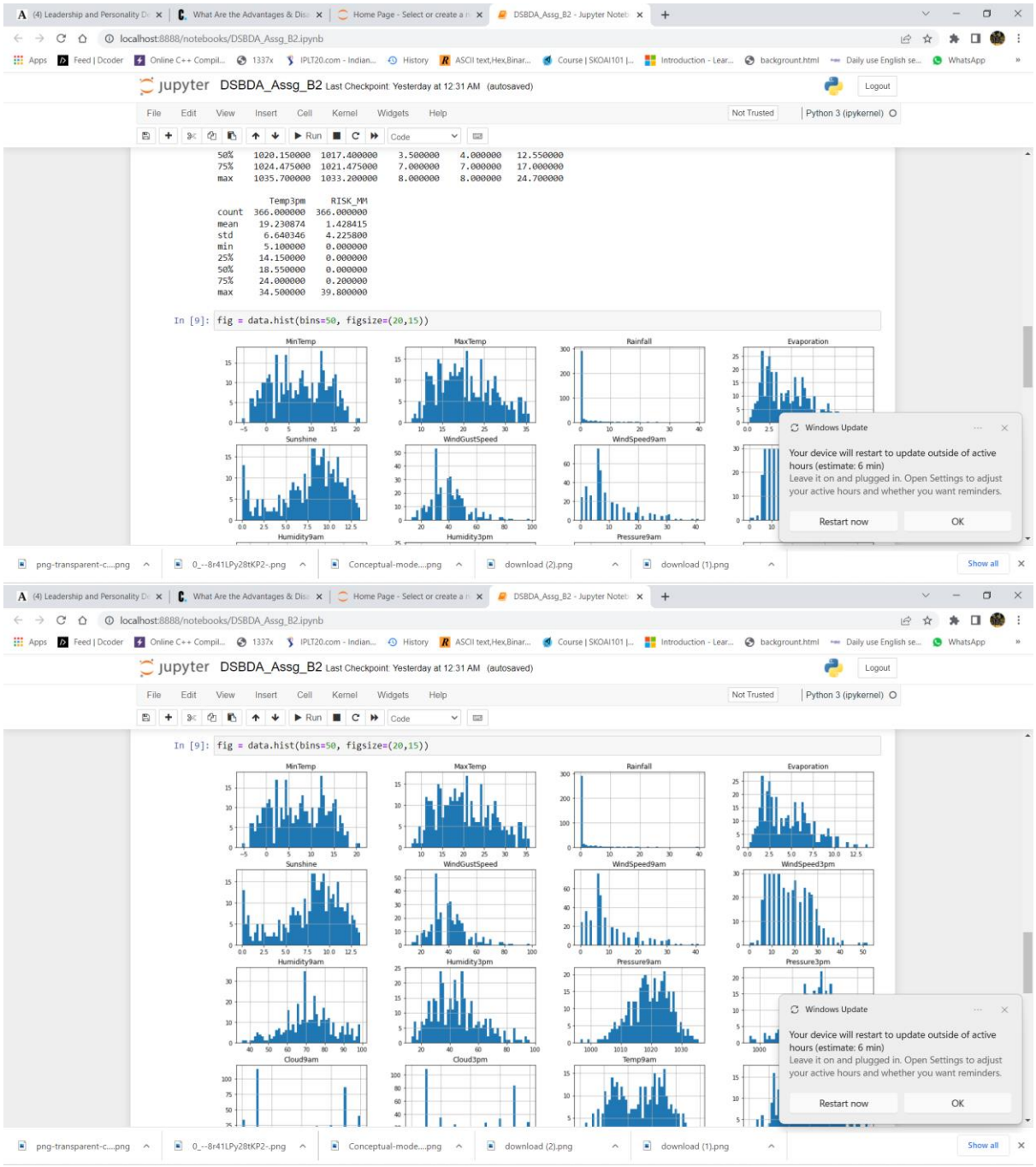
png-transparent-c....png | 0_--8r41LPy28tKP2-.png | Conceptual-mode....png | download (2).png | download (1).png | Show all


jupyter DSBDA_Assg_B2 Last Checkpoint: Yesterday at 12:31 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [4]: #import thr dataset
import numpy as np
import pandas as pd
In [5]: data = pd.read_csv('weather.csv')
In [6]: data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 366 entries, 0 to 365
Data columns (total 22 columns):
Column Non-Null Count Dtype
0 MinTemp 366 non-null float64
1 MaxTemp 366 non-null float64
2 Rainfall 366 non-null float64
3 Evaporation 366 non-null float64
4 Sunshine 363 non-null float64
5 WindGustDir 363 non-null object
6 WindGustSpeed 364 non-null float64
7 WindDir9am 335 non-null object
8 WindDir3pm 365 non-null object
9 WindSpeed9am 359 non-null float64
10 WindSpeed3pm 366 non-null int64
11 Humidity9am 366 non-null int64
12 Humidity3pm 366 non-null int64
13 Pressure9am 366 non-null float64
14 Pressure3pm 366 non-null float64
15 Cloud9am 366 non-null int64
16 Cloud3pm 366 non-null int64
Windows Update
Your device will restart to update outside of active hours (estimate: 6 min)
Leave it on and plugged in. Open Settings to adjust your active hours and whether you want reminders.
Restart now
OK
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

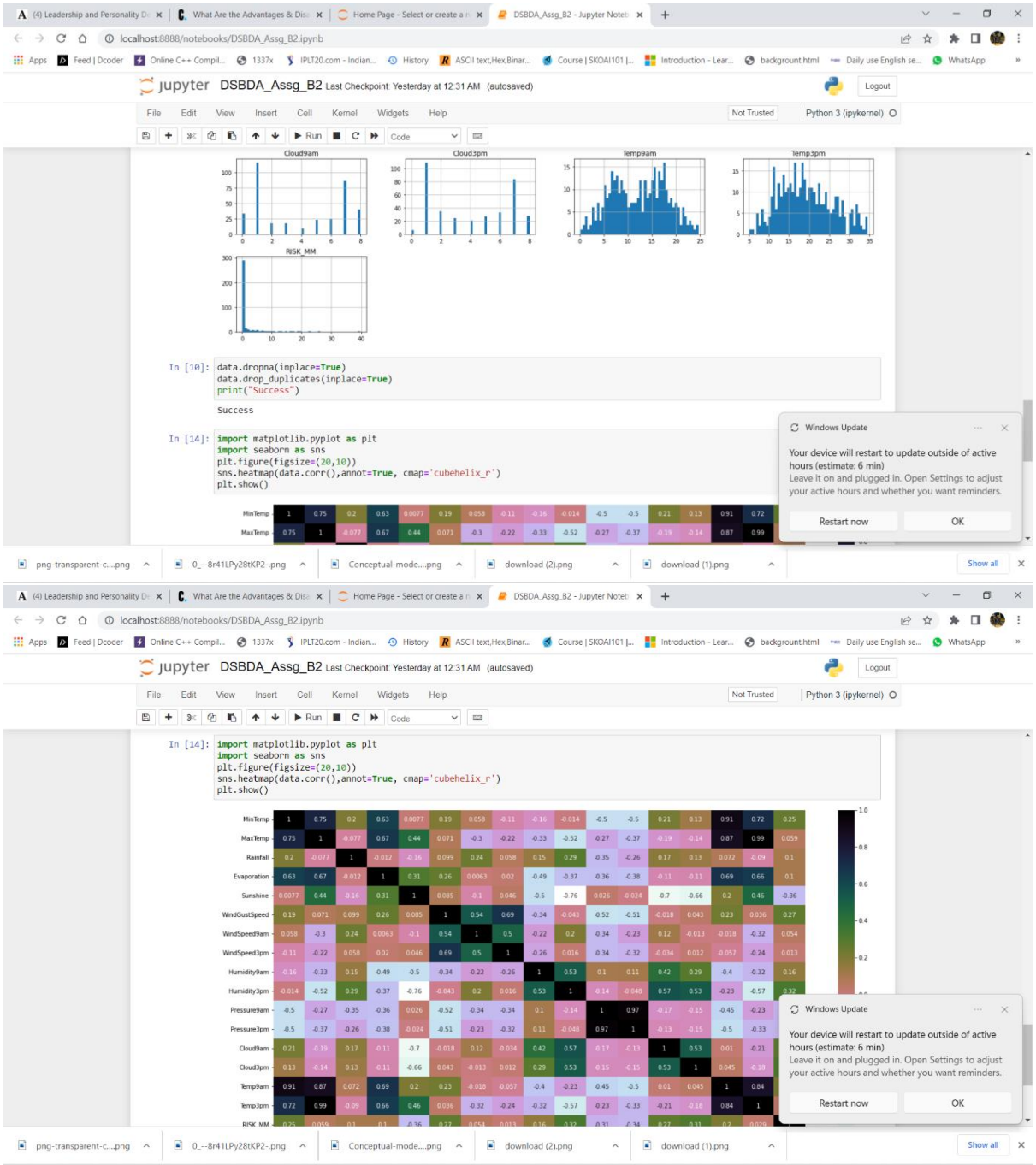
jupyter DSBDA_Assg_B2 Last Checkpoint: Yesterday at 12:31 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
3 Evaporation 366 non-null float64
4 Sunshine 363 non-null float64
5 WindGustDir 363 non-null object
6 WindGustSpeed 364 non-null float64
7 WindDir9am 335 non-null object
8 WindDir3pm 365 non-null object
9 WindSpeed9am 359 non-null float64
10 WindSpeed3pm 366 non-null int64
11 Humidity9am 366 non-null int64
12 Humidity3pm 366 non-null int64
13 Pressure9am 366 non-null float64
14 Pressure3pm 366 non-null float64
15 Cloud9am 366 non-null int64
16 Cloud3pm 366 non-null int64
17 Temp9am 366 non-null float64
18 Temp3pm 366 non-null float64
19 RainToday 366 non-null object
20 RISK_MM 366 non-null float64
21 RainTomorrow 366 non-null object
dtypes: float64(12), int64(5), object(5)
memory usage: 63.0+ KB
In [7]: data.head(10)
Out[7]:
MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir WindGustSpeed WindDir9am WindDir3pm WindSpeed9am ...
0 8.0 24.3 0.0 3.4 6.3 NW 30.0 SW NW 6.0 ...
1 14.0 26.9 3.6 4.4 9.7 ENE 39.0 E W 4.0 ...
2 13.7 23.4 3.6 5.8 3.3 NW 85.0 N NNE 6.0 ...
3 13.3 15.5 39.8 7.2 9.1 NW 54.0 WNW W 30.0 ...
4 7.6 16.1 2.8 5.6 10.6 SSE 50.0 SSE ESE 20.0 ...
5 6.2 16.9 0.0 5.8 8.2 SE 44.0 SE E 20.0 ...
Windows Update
Your device will restart to update outside of active hours (estimate: 6 min)
Leave it on and plugged in. Open Settings to adjust your active hours and whether you want reminders.
Restart now
OK
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all


jupyter DSBDA_Assg_B2 Last Checkpoint: Yesterday at 12:31 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
memory usage: 63.0+ KB
In [7]:
data.head(10)
Out[7]:
MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir WindGustSpeed WindDir9am WindDir3pm WindSpeed9am ... Humidity3pm Pressure9a
0 8.0 24.3 0.0 3.4 6.3 NW 30.0 SW NW 6.0 ... 29 1019
1 14.0 26.9 3.6 4.4 9.7 ENE 39.0 E W 4.0 ... 36 1012
2 13.7 23.4 3.6 5.8 3.3 NW 85.0 N NNE 6.0 ... 69 1009
3 13.3 15.5 39.8 7.2 9.1 NW 54.0 WNW W 30.0 ... 56 1005
4 7.6 16.1 2.8 5.6 10.6 SSE 50.0 SSE ESE 20.0 ... 49 1018
5 6.2 16.9 0.0 5.8 8.2 SE 44.0 SE E 20.0 ... 57 1023
6 6.1 18.2 0.2 4.2 8.4 SE 43.0 SE ESE 19.0 ... 47 1024
7 8.3 17.0 0.0 5.6 4.6 E 41.0 SE E 11.0 ... 57 1026
8 8.8 19.5 0.0 4.0 4.1 S 48.0 E ENE 19.0 ... 48 1026
9 8.4 22.8 16.2 5.4 7.7 E 31.0 S ESE 7.0 ... 32 1024
10 rows × 22 columns
In [8]:
print("describe: ")
print(data.describe())
describe:
MinTemp MaxTemp Rainfall Evaporation Sunshine \
count 366.000000 366.000000 366.000000 366.000000 363.000000
mean 7.265574 20.550273 1.428415 4.521858 7.909366
std 6.025800 6.690516 4.225800 2.669383 3.481517
min -5.300000 7.600000 0.000000 0.200000 0.000000
Windows Update
Your device will restart to update outside of active hours (estimate: 6 min)
Leave it on and plugged in. Open Settings to adjust your active hours and whether you want reminders.
Restart now
OK
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

jupyter DSBDA_Assg_B2 Last Checkpoint: Yesterday at 12:31 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [8]:
print("describe: ")
print(data.describe())
describe:
MinTemp MaxTemp Rainfall Evaporation Sunshine \
count 366.000000 366.000000 366.000000 366.000000 363.000000
mean 7.265574 20.550273 1.428415 4.521858 7.909366
std 6.025800 6.690516 4.225800 2.669383 3.481517
min -5.300000 7.600000 0.000000 0.200000 0.000000
25% 2.300000 15.025000 0.000000 2.200000 5.950000
50% 7.450000 19.650000 0.000000 4.200000 8.600000
75% 12.500000 25.500000 0.200000 6.400000 10.500000
max 20.900000 35.800000 39.800000 13.800000 13.600000
WindGustSpeed WindSpeed9am WindSpeed3pm Humidity9am Humidity3pm \
count 364.000000 359.000000 366.000000 366.000000 366.000000
mean 39.840659 9.651811 17.986339 72.035519 44.519126
std 13.059807 7.951929 8.856997 13.137058 16.850947
min 13.000000 0.000000 0.000000 36.000000 13.000000
25% 31.000000 6.000000 11.000000 64.000000 32.250000
50% 39.000000 7.000000 17.000000 72.000000 43.000000
75% 46.000000 13.000000 24.000000 81.000000 55.000000
max 98.000000 41.000000 52.000000 99.000000 96.000000
Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am \
count 366.000000 366.000000 366.000000 366.000000 366.000000
mean 1019.709016 1016.810383 3.890710 4.024590 12.358470
std 6.686212 6.469422 2.956131 2.666268 5.630832
min 996.500000 996.800000 0.000000 0.000000 0.100000
25% 1015.350000 1012.800000 1.000000 1.000000 7.625000
50% 1020.150000 1017.400000 3.500000 4.000000 12.550000
75% 1024.475000 1021.475000 7.000000 7.000000 17.000000
max 1035.700000 1033.200000 8.000000 8.000000 24.700000
Windows Update
Your device will restart to update outside of active hours (estimate: 6 min)
Leave it on and plugged in. Open Settings to adjust your active hours and whether you want reminders.
Restart now
OK
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all


jupyter DSBDA_Assg_B2 Last Checkpoint: Yesterday at 12:31 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
50% 1020.150000 1017.400000 3.500000 4.000000 12.550000
75% 1024.475000 1021.475000 7.000000 7.000000 17.000000
max 1035.700000 1033.200000 8.000000 8.000000 24.700000
Temp3pm RISK_MM
count 366.000000 366.000000
mean 19.230874 1.428415
std 6.640346 4.225800
min 5.100000 0.000000
25% 14.150000 0.000000
50% 18.550000 0.000000
75% 24.000000 0.200000
max 34.500000 39.800000
In [9]: fig = data.hist(bins=50, figsize=(20,15))
MinTemp
MaxTemp
Rainfall
Evaporation
Sunshine
WindGustSpeed
WindSpeed9am
Humidity9am
Humidity3pm
Pressure9am
Windows Update
Your device will restart to update outside of active hours (estimate: 6 min)
Leave it on and plugged in. Open Settings to adjust your active hours and whether you want reminders.
Restart now
OK
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

jupyter DSBDA_Assg_B2 Last Checkpoint: Yesterday at 12:31 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [9]: fig = data.hist(bins=50, figsize=(20,15))
MinTemp
MaxTemp
Rainfall
Evaporation
Sunshine
WindGustSpeed
WindSpeed9am
WindSpeed3pm
Humidity9am
Humidity3pm
Pressure9am
Pressure3pm
Cloud9am
Cloud3pm
Temp9am
Windows Update
Your device will restart to update outside of active hours (estimate: 6 min)
Leave it on and plugged in. Open Settings to adjust your active hours and whether you want reminders.
Restart now
OK
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all


jupyter DSBDA_Assg_B2 Last Checkpoint: Yesterday at 12:31 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Cloud9am
Cloud3pm
Temp9am
Temp3pm
RISK_MM
In [10]:
data.dropna(inplace=True)
data.drop_duplicates(inplace=True)
print("Success")
Success
In [14]:
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(20,10))
sns.heatmap(data.corr(),annot=True, cmap='cubehelix_r')
plt.show()
Windows Update
Your device will restart to update outside of active hours (estimate: 6 min)
Leave it on and plugged in. Open Settings to adjust your active hours and whether you want reminders.
Restart now
OK
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

jupyter DSBDA_Assg_B2 Last Checkpoint: Yesterday at 12:31 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [14]:
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(20,10))
sns.heatmap(data.corr(),annot=True, cmap='cubehelix_r')
plt.show()
MinTemp
MaxTemp
Rainfall
Evaporation
Sunshine
WindGustSpeed
WindSpeed9am
WindSpeed3pm
Humidity9am
Humidity3pm
Pressure9am
Pressure3pm
Cloud9am
Cloud3pm
Temp9am
Temp3pm
RISK_MM
Windows Update
Your device will restart to update outside of active hours (estimate: 6 min)
Leave it on and plugged in. Open Settings to adjust your active hours and whether you want reminders.
Restart now
OK
png-transparent-c....png
0_--8r41LPy28tKP2-.png
Conceptual-mode....png
download (2).png
download (1).png
Show all

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [6]:

```
df = pd.read_csv('data_science.csv')
df.head()
```

```
C:\Users\Acer\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3165: DtypeWar
ning: Columns (9) have mixed types.Specify dtype option on import or set low_memory=False
.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

Out[6]:

	id	conversation_id	created_at	date	time	timezone	user_id	username
0	1406400408545804288	1406400396264943616	2021-06-20 05:26:01 IST	2021-06-20	05:26:01	530	1113747629282930688	ballouxfrancois
1	1406390341176016897	1406390341176016897	2021-06-20 04:46:01 IST	2021-06-20	04:46:01	530	788898706586275840	tdatascience
2	1406386311481774083	1406386311481774083	2021-06-20 04:30:00 IST	2021-06-20	04:30:00	530	19402238	sciencenews
3	1406383545153638402	1406383545153638402	2021-06-20 04:19:01 IST	2021-06-20	04:19:01	530	788898706586275840	tdatascience
4	1406358632648818689	1406358632648818689	2021-06-20 02:40:01 IST	2021-06-20	02:40:01	530	788898706586275840	tdatascience

5 rows × 36 columns

In [7]:

```
df.shape
```

Out[7]:

```
(241386, 36)
```

In [8]:

```
df.isnull().sum()
```

Out[8]:

```
id                  0
conversation_id     0
created_at          0
date                0
time                0
timezone            0
user_id             0
username            0
name                0
```

```
place              241032
tweet                   0
language                0
mentions                0
urls                    0
photos                  0
replies_count           0
retweets_count          0
likes_count             0
hashtags                0
cashtags                0
link                    0
retweet                 0
quote_url          231065
video                   0
thumbnail          131048
near               241386
geo                241386
source             241386
user_rt_id         241386
user_rt            241386
retweet_id         241386
reply_to                0
retweet_date       241386
translate          241386
trans_src          241386
trans_dest         241386
dtype: int64
```

In [9]:

```
#data cleaning
```

In [10]:

```
columns_to_drop = list(df.columns[27:])
df.drop(columns = columns_to_drop, axis = 1, inplace = True)
```

In [11]:

```
df.isnull().sum()
```

Out[11]:

```
id                      0
conversation_id         0
created_at              0
date                    0
time                    0
timezone                0
user_id                 0
username                0
name                    0
place              241032
tweet                   0
language                0
mentions                0
urls                    0
photos                  0
replies_count           0
retweets_count          0
likes_count             0
hashtags                0
cashtags                0
link                    0
retweet                 0
quote_url          231065
video                   0
thumbnail          131048
near               241386
geo                241386
dtype: int64
```

In [12]:

```
df.drop(columns = ['place', 'quote_url', 'thumbnail', 'near', 'geo'], axis = 1, inplace
= True)
```

In [13]:

```
df.isnull().sum()
```

Out[13]:

```
id                  0
conversation_id     0
created_at          0
date                0
time                0
timezone            0
user_id             0
username            0
name                0
tweet               0
language            0
mentions            0
urls                0
photos              0
replies_count       0
retweets_count      0
likes_count         0
hashtags            0
cashtags            0
link                0
retweet             0
video               0
dtype: int64
```

In [14]:

```
pd.set_option('display.max_columns', None)
df.head()
```

Out[14]:

	id	conversation_id	created_at	date	time	timezone	user_id	username
0	1406400408545804288	1406400396264943616	2021-06-20 05:26:01 IST	2021-06-20	05:26:01	530	1113747629282930688	ballouxfrancois
1	1406390341176016897	1406390341176016897	2021-06-20 04:46:01 IST	2021-06-20	04:46:01	530	788898706586275840	tdatascience
2	1406386311481774083	1406386311481774083	2021-06-20 04:30:00 IST	2021-06-20	04:30:00	530	19402238	sciencenews
3	1406383545153638402	1406383545153638402	2021-06-20 04:19:01 IST	2021-06-20	04:19:01	530	788898706586275840	tdatascience
4	1406358632648818689	1406358632648818689	2021-06-20 02:40:01 IST	2021-06-20	02:40:01	530	788898706586275840	tdatascience

In [15]:

```
#Dropping cashtags column
```

```
df.drop(columns = ['cashtags'], axis = 1, inplace= True)
```

In [16]:

```
df.head()
```

Out[16]:

	id	conversation_id	created_at	date	time	timezone	user_id	username
0	1406400408545804288	1406400396264943616	2021-06-20 05:26:01 IST	2021-06-20	05:26:01	530	1113747629282930688	ballouxfrancois
1	1406390341176016897	1406390341176016897	2021-06-20 04:46:01 IST	2021-06-20	04:46:01	530	788898706586275840	tdatascience
2	1406386311481774083	1406386311481774083	2021-06-20 04:30:00 IST	2021-06-20	04:30:00	530	19402238	sciencenews
3	1406383545153638402	1406383545153638402	2021-06-20 04:19:01 IST	2021-06-20	04:19:01	530	788898706586275840	tdatascience
4	1406358632648818689	1406358632648818689	2021-06-20 02:40:01 IST	2021-06-20	02:40:01	530	788898706586275840	tdatascience

In [17]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 241386 entries, 0 to 241385
Data columns (total 21 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   id                241386 non-null  int64 
 1   conversation_id   241386 non-null  int64 
 2   created_at        241386 non-null  object
 3   date              241386 non-null  object
 4   time              241386 non-null  object
 5   timezone          241386 non-null  int64 
 6   user_id           241386 non-null  int64 
 7   username          241386 non-null  object
 8   name              241386 non-null  object
 9   tweet             241386 non-null  object
 10  language          241386 non-null  object
 11  mentions          241386 non-null  object
 12  urls              241386 non-null  object
 13  photos            241386 non-null  object
 14  replies_count     241386 non-null  int64 
 15  retweets_count    241386 non-null  int64 
 16  likes_count       241386 non-null  int64 
 17  hashtags          241386 non-null  object
 18  link              241386 non-null  object
 19  retweet           241386 non-null  bool  
 20  video             241386 non-null  int64 
dtypes: bool(1), int64(8), object(12)
memory usage: 37.1+ MB
```

In [18]:

```
#Removing IST from created_at column
df['created_at'] = df['created_at'].apply(lambda tweet: tweet.replace('IST', '').strip()
```

```
)
```

In [19]:

```
df['created_at'] = pd.to_datetime(df['created_at'])
df['date'] = pd.to_datetime(df['date'])
df['time'] = pd.to_datetime(df['time'])
```

In [20]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 241386 entries, 0 to 241385
Data columns (total 21 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   id               241386 non-null  int64
 1   conversation_id  241386 non-null  int64
 2   created_at       241386 non-null  datetime64[ns]
 3   date             241386 non-null  datetime64[ns]
 4   time             241386 non-null  datetime64[ns]
 5   timezone         241386 non-null  int64
 6   user_id          241386 non-null  int64
 7   username         241386 non-null  object
 8   name             241386 non-null  object
 9   tweet            241386 non-null  object
 10  language         241386 non-null  object
 11  mentions         241386 non-null  object
 12  urls             241386 non-null  object
 13  photos           241386 non-null  object
 14  replies_count    241386 non-null  int64
 15  retweets_count   241386 non-null  int64
 16  likes_count      241386 non-null  int64
 17  hashtags         241386 non-null  object
 18  link             241386 non-null  object
 19  retweet          241386 non-null  bool
 20  video            241386 non-null  int64
dtypes: bool(1), datetime64[ns](3), int64(8), object(9)
memory usage: 37.1+ MB
```

In [21]:

```
df = df.astype({'username':'category', 'name':'category', 'tweet':'category', 'language'
:'category', 'link':'category', 'urls':'category'})
```

In [22]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 241386 entries, 0 to 241385
Data columns (total 21 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   id               241386 non-null  int64
 1   conversation_id  241386 non-null  int64
 2   created_at       241386 non-null  datetime64[ns]
 3   date             241386 non-null  datetime64[ns]
 4   time             241386 non-null  datetime64[ns]
 5   timezone         241386 non-null  int64
 6   user_id          241386 non-null  int64
 7   username         241386 non-null  category
 8   name             241386 non-null  category
 9   tweet            241386 non-null  category
 10  language         241386 non-null  category
 11  mentions         241386 non-null  object
 12  urls             241386 non-null  category
 13  photos           241386 non-null  object
 14  replies_count    241386 non-null  int64
 15  retweets_count   241386 non-null  int64
 16  likes_count      241386 non-null  int64
```

```
 17  hashtags          241386 non-null  object
 18  link              241386 non-null  category
 19  retweet           241386 non-null  bool
 20  video             241386 non-null  int64
dtypes: bool(1), category(6), datetime64[ns](3), int64(8), object(3)
memory usage: 56.1+ MB
```

In [23]:

```
#splitting dataframe into 4 parts
```

In [24]:

```
parts = np.array_split(df,4)
```

In [25]:

```
parts[0].head()
```

Out[25]:

	id	conversation_id	created_at	date	time	timezone	user_id	username
0	1406400408545804288	1406400396264943616	2021-06-20 05:26:01	2021-06-20	2022-05-04 05:26:01	530	1113747629282930688	ballouxfrancois
1	1406390341176016897	1406390341176016897	2021-06-20 04:46:01	2021-06-20	2022-05-04 04:46:01	530	788898706586275840	tdatascience
2	1406386311481774083	1406386311481774083	2021-06-20 04:30:00	2021-06-20	2022-05-04 04:30:00	530	19402238	sciencenews
3	1406383545153638402	1406383545153638402	2021-06-20 04:19:01	2021-06-20	2022-05-04 04:19:01	530	788898706586275840	tdatascience
4	1406358632648818689	1406358632648818689	2021-06-20 02:40:01	2021-06-20	2022-05-04 02:40:01	530	788898706586275840	tdatascience

In [26]:

```
#Converting to lowercase
def to_lowercase(text):
    return text.lower()
```

In [27]:

```
for i in range(4):
    parts[i]['tweet'] = parts[i]['tweet'].apply(lambda tweet: to_lowercase(tweet))
```

In [28]:

```
for i in range(4):
    print(parts[i]['tweet'].head())
```

```
0    what can be done?  - never blindly trust an ab...
1    "we need a paradigm shift from model-centric t...
2    using high-resolution satellite data and compu...
3    .@stephenson_data shares four steps that will ...
4    "curricula is inherently brittle in a world wh...
Name: tweet, dtype: object
60347    the world is digital — learn about the power o...
60348    build your data brilliance in 2020 — explore y...
60349    the end is near, but there is still time to ea...
60350    how to make the transition from analyst to #da...
60351    that time when the australian worked with clim...
Name: tweet, dtype: object
120694    checking out "22 tips for better data science"
```

```
120694    checking out  22 tips for better data science ...
120695    10 questions every data decision-maker should ...
120696    tired of data silos? join our webinar on may 2...
120697    data science is helping people in india with t...
120698    data science master's students visit accenture...
Name: tweet, dtype: object
181040    join the #datascience movement. immerse yourse...
181041    don't forget to vote for @rapidminer in @kdnug...
181042    want to learn more about our #datascience and ...
181043    @andrewchen @mikeschiemer @dberkowitz @rwang0 ...
181044    data are data and our field is exceptionally g...
Name: tweet, dtype: object
```

In [29]:

```
#removing special character ,html tags and numbers
```

In [30]:

```
from bs4 import BeautifulSoup
import re
```

In [31]:

```
def clean_text(text):
    soup = BeautifulSoup(text, 'html.parser')
    cleaned_text = soup.get_text()
    cleaned_text = re.sub('[^A-Za-z0-9]+', ' ', cleaned_text)
    return cleaned_text
```

In [32]:

```
for i in range(4):
    parts[i]['tweet'] = parts[i]['tweet'].apply(lambda tweet: clean_text(tweet))
```

In [33]:

```
#tokenization
```

In [41]:

```
from nltk.tokenize import word_tokenize
```

In [42]:

```
def tokenization(text):
    tokens = word_tokenize(text)
    return tokens
```

In [ ]:

```
for i in range(4):
    parts[i]['tweet'] = parts[i]['tweet'].apply(lambda tweet: tokenization(tweet))
```

In [ ]:

```
#removing stopwords
```

In [44]:

```
from nltk.corpus import stopwords
```

In [ ]:

```
for i in range(4):
    stop_words = set(stopwords.words('english'))
    parts[i]['tweet'] = [[word for word in tweet if word not in stop_words] for tweet in
parts[i]['tweet']]
```

In [46]:

```
for i in range(4):
    print(parts[i]['tweet'].head())
```

```
0    what can be done never blindly trust an abstra...
1     we need a paradigm shift from model centric t...
2    using high resolution satellite data and compu...
3     stephenson data shares four steps that will h...
4     curricula is inherently brittle in a world wh...
Name: tweet, dtype: object
60347    the world is digital learn about the power of ...
60348    build your data brilliance in 2020 explore you...
60349    the end is near but there is still time to ear...
60350    how to make the transition from analyst to dat...
60351    that time when the australian worked with clim...
Name: tweet, dtype: object
120694    checking out 22 tips for better data science d...
120695    10 questions every data decision maker should ...
120696    tired of data silos join our webinar on may 24...
120697    data science is helping people in india with t...
120698    data science master s students visit accenture...
Name: tweet, dtype: object
181040    join the datascience movement immerse yourself...
181041    don t forget to vote for rapidminer in kdnugge...
181042    want to learn more about our datascience and w...
181043     andrewchen mikeschiemer dberkowitz rwang0 our...
181044    data are data and our field is exceptionally g...
Name: tweet, dtype: object
```

In [47]:

```
#Stemming and Lemmatization
```

In [48]:

```
from nltk.stem import PorterStemmer
porter = PorterStemmer()
```

In [49]:

```
for i in range(4):
    parts[i]['tweet'] = [[porter.stem(word) for word in tweet ] for tweet in parts[i]['t
weet']]
```

In [50]:

```
#Sentiment Analysis
```

In [ ]:

```
from textblob import TextBlob
```

In [ ]:

```
def get_sentiment(tweet):
    text = ' '.join(word for word in tweet)
    analysis = TextBlob(text)
    if analysis.sentiment.polarity > 0:
            return 'positive'
    elif analysis.sentiment.polarity == 0:
        return 'neutral'
    else:
        return 'negative'
```

In [ ]:

```
for i in range(4):
    parts[i]['sentiment'] = parts[i]['tweet'].apply(lambda tweet : get_sentiment(tweet))
```

In [ ]:

```
for i in range(4):
    print(parts[i][['tweet', 'sentiment']].head())
```

In [52]:

```
#joining all in one dataframe
```

In [53]:

```
df_sentiment = parts[0].append([parts[1], parts[2], parts[3]])
```

In [54]:

```
df_sentiment.head()
```

Out[54]:

	id	conversation_id	created_at	date	time	timezone	user_id	username
0	1406400408545804288	1406400396264943616	2021-06-20 05:26:01	2021-06-20	2022-05-04 05:26:01	530	1113747629282930688	ballouxfrancois
1	1406390341176016897	1406390341176016897	2021-06-20 04:46:01	2021-06-20	2022-05-04 04:46:01	530	788898706586275840	tdatascience
2	1406386311481774083	1406386311481774083	2021-06-20 04:30:00	2021-06-20	2022-05-04 04:30:00	530	19402238	sciencenews
3	1406383545153638402	1406383545153638402	2021-06-20 04:19:01	2021-06-20	2022-05-04 04:19:01	530	788898706586275840	tdatascience
4	1406358632648818689	1406358632648818689	2021-06-20 02:40:01	2021-06-20	2022-05-04 02:40:01	530	788898706586275840	tdatascience

In [ ]:

```

```