

1. Train a simple linear regressing model on dataset and predict the output.

Train dataset :- https://docs.google.com/spreadsheets/d/e/2PACX-1vRTK2NvcndgPX41Czu6Ft2Ho_nE-z50BgTqdzFW0rsJ2nvyNLe2DolG1COzUbgw80oaRBjfy5-WtFk/pubhtml

```
import pandas as pd
```

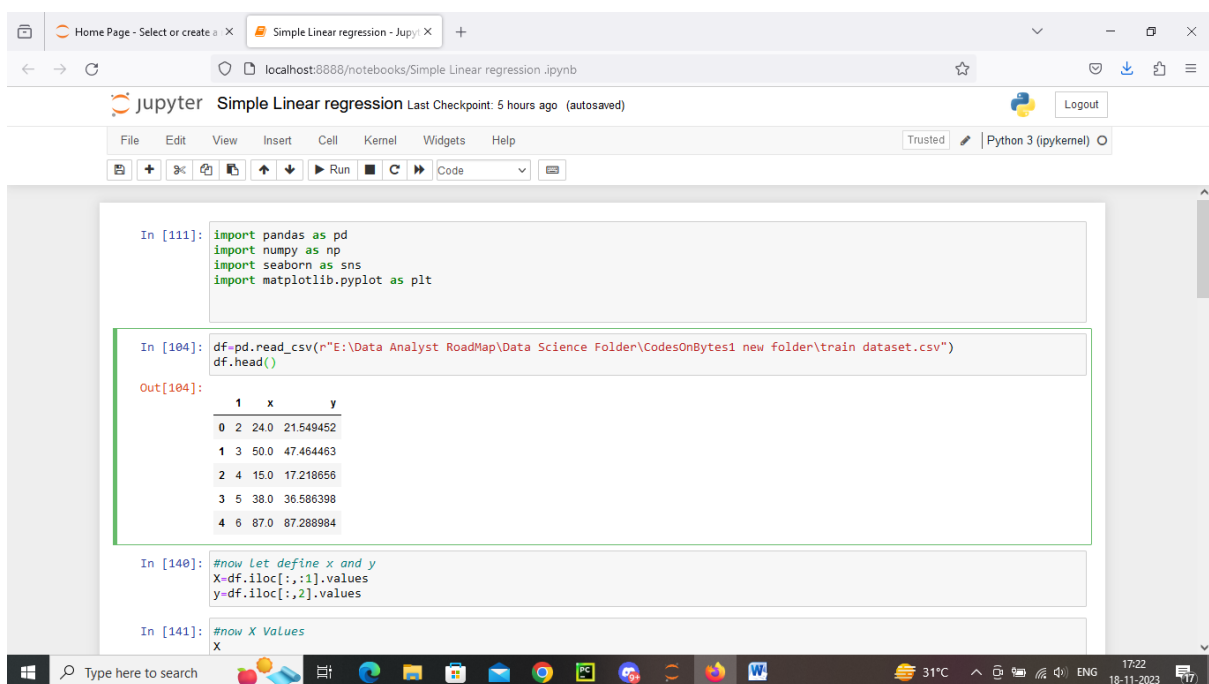
```
import numpy as np
```

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
df=pd.read_csv(r"E:\Data Analyst RoadMap\Data Science Folder\CodesOnBytes1 new folder\train dataset.csv")
```

```
df.head()
```



The screenshot shows a Jupyter Notebook window titled "Simple Linear regression". The code in the notebook is as follows:

```
In [111]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

In [104]: df=pd.read_csv(r"E:\Data Analyst RoadMap\Data Science Folder\CodesOnBytes1 new folder\train dataset.csv")
df.head()

Out[104]:
```

	x	y
0	24.0	21.549452
1	50.0	47.464463
2	15.0	17.218656
3	38.0	36.586398
4	87.0	87.288984

```
In [140]: #now let define x and y
X=df.iloc[:,1].values
y=df.iloc[:,2].values

In [141]: #now X Values
X
```

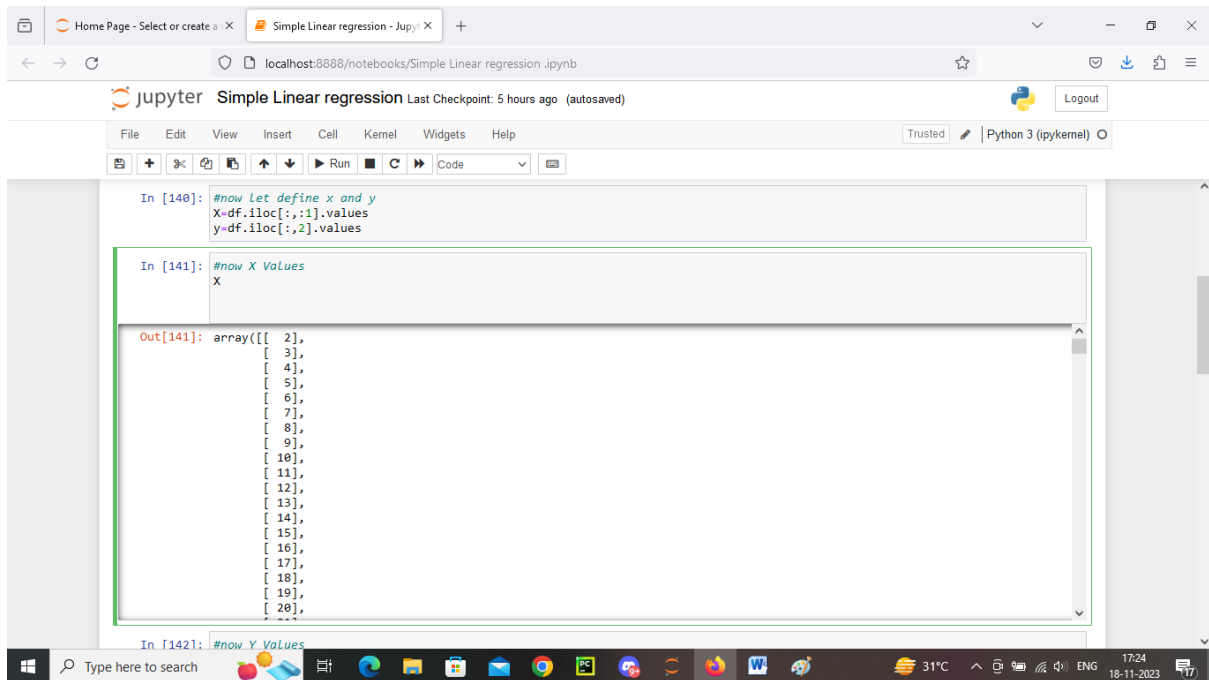
```
#now let define x and y
```

```
X=df.iloc[:,1].values
```

```
y=df.iloc[:,2].values
```

#now X Values

X



A screenshot of a Jupyter Notebook interface. The browser address bar shows 'localhost:8888/notebooks/Simple Linear regression .ipynb'. The notebook title is 'Simple Linear regression' with a 'Last Checkpoint: 5 hours ago (autosaved)' status. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The code area contains two cells. The first cell, labeled 'In [140]:', defines variables x and y from a DataFrame df: `#now Let define x and y`, `x=df.iloc[:,1].values`, and `y=df.iloc[:,2].values`. The second cell, labeled 'In [141]:', is titled '#now X Values' and contains the variable `x`. The output area, labeled 'Out[141]:', displays a 1D array of 20 values: `array([[2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]])`. The Windows taskbar at the bottom shows the search bar, application icons, system tray with temperature (31°C), and date/time (17:25, 18-11-2023).

```
In [140]: #now Let define x and y
x=df.iloc[:,1].values
y=df.iloc[:,2].values

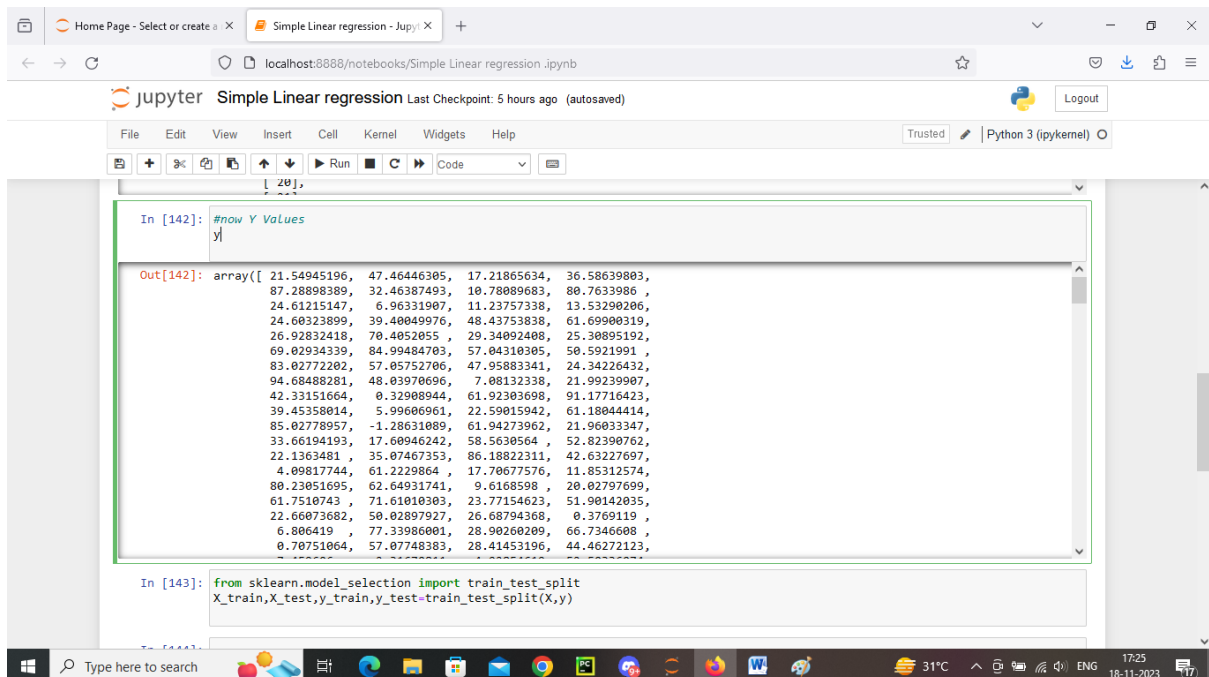
In [141]: #now X Values
x

Out[141]: array([[ 2],
 [ 3],
 [ 4],
 [ 5],
 [ 6],
 [ 7],
 [ 8],
 [ 9],
 [10],
 [11],
 [12],
 [13],
 [14],
 [15],
 [16],
 [17],
 [18],
 [19],
 [20],
 [21]])

In [142]: #now Y Values
y
```

#now Y Values

y



A screenshot of a Jupyter Notebook interface, continuing from the previous one. The code area contains two cells. The first cell, labeled 'In [142]:', is titled '#now Y Values' and contains the variable `y`. The output area, labeled 'Out[142]:', displays a 2D array of 20 rows and 4 columns of numerical values. The second cell, labeled 'In [143]:', imports `train_test_split` from `sklearn.model_selection` and performs the split: `X_train,X_test,y_train,y_test=train_test_split(X,y)`. The Windows taskbar at the bottom is identical to the previous screenshot.

```
In [142]: #now Y Values
y

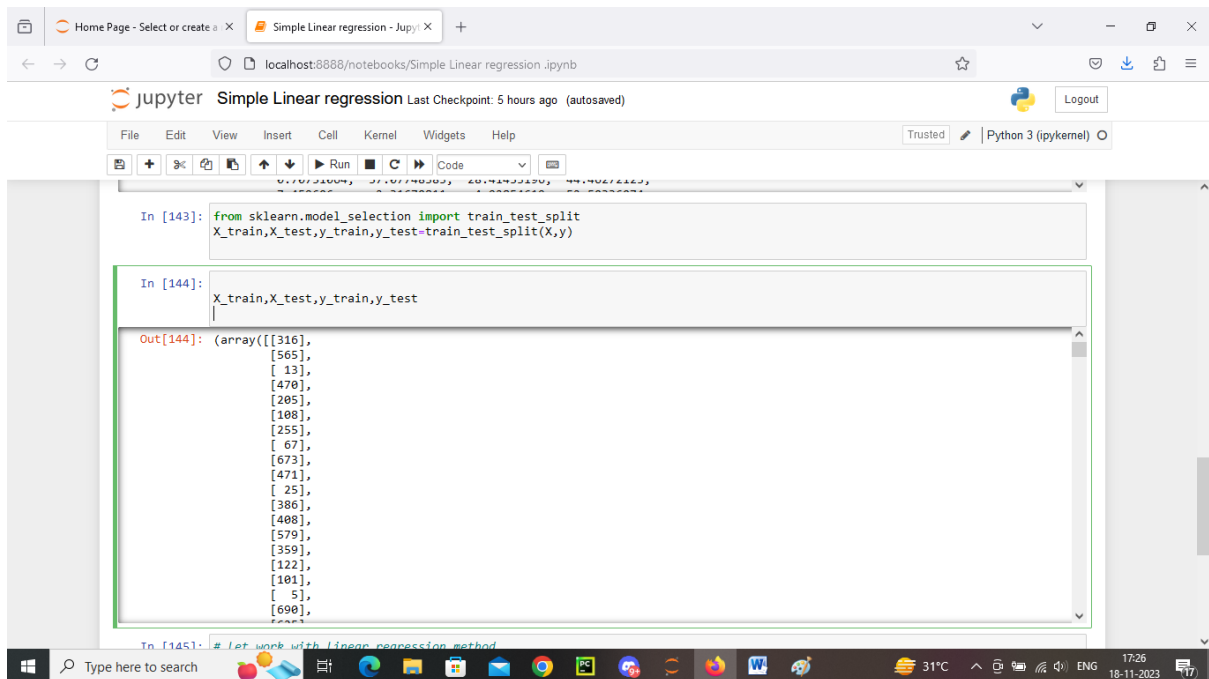
Out[142]: array([[ 21.54945196,  47.46446305,  17.21865634,  36.58639803,
  87.28898389,  32.46387493,  10.78089683,  80.7633986 ,
  24.61215147,  6.96331907,  11.23757338,  13.53290206,
  24.60323899,  39.40049976,  48.43753838,  61.69900319,
  26.92832418,  70.4052055 ,  29.34092408,  25.30895192,
  69.02934339,  84.99484703,  57.04310305,  50.5921991 ,
  83.02772202,  57.05752706,  47.95883341,  24.34226432,
  94.68488281,  48.03970696,  7.08132338,  21.99239907,
  42.33151664,  0.32908944,  61.92303698,  91.17716423,
  39.45358014,  5.99606961,  22.59015942,  61.18044414,
  85.02778957, -1.28631089,  61.94273962,  21.96033347,
  33.66194193,  17.60946242,  58.5630564 ,  52.82390762,
  22.1363481 ,  35.07467353,  86.18822311,  42.63227697,
  4.09817744,  61.2229864 ,  17.70677576,  11.85312574,
  80.23051695,  62.64931741,  9.6168598 ,  20.02797699,
  61.7510743 ,  71.61010303,  23.77154623,  51.90142035,
  22.66073682,  50.02897927,  26.68794368,  0.3769119 ,
  6.806419 ,  77.33986001,  28.90260209,  66.7346608 ,
  0.70751064,  57.07748383,  28.41453196,  44.46272123,
  1.14444444,  1.14444444,  1.14444444,  1.14444444])

In [143]: from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y)
```

from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test=train_test_split(X,y)

X_train,X_test,y_train,y_test



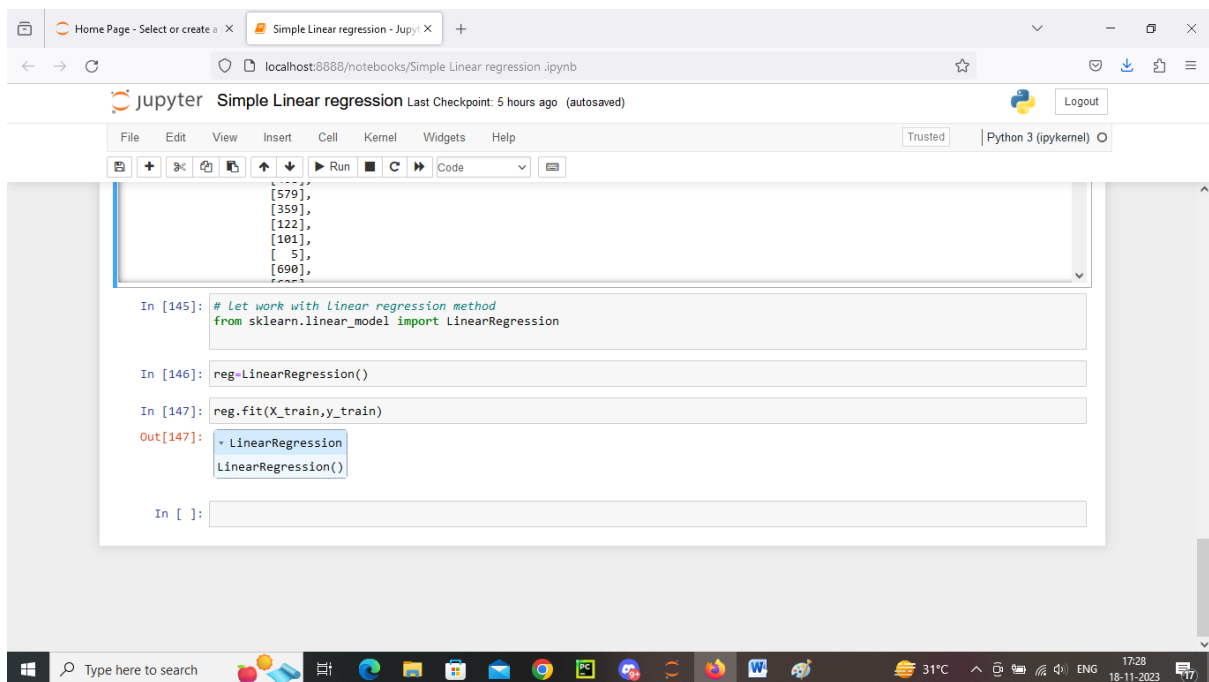
A screenshot of a Jupyter Notebook titled "Simple Linear regression". The notebook is running on a local host (localhost:8888). The interface shows a menu bar with options like File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu bar, there are tabs for "Simple Linear regression - Jupyter" and "Simple Linear regression - Jupyter". The notebook content shows two code cells. The first cell (In [143]) contains the code: `from sklearn.model_selection import train_test_split` and `X_train,X_test,y_train,y_test=train_test_split(X,y)`. The second cell (In [144]) contains the code: `X_train,X_test,y_train,y_test`. The output of the second cell (Out[144]) shows a 2D array of data points: `(array([[316], [565], [13], [470], [205], [108], [255], [67], [673], [471], [25], [386], [408], [579], [359], [122], [101], [5], [690], [222]]))`. The notebook is running on Python 3 (ipykernel).

let work with linear regression method

from sklearn.linear_model import LinearRegression

reg=LinearRegression()

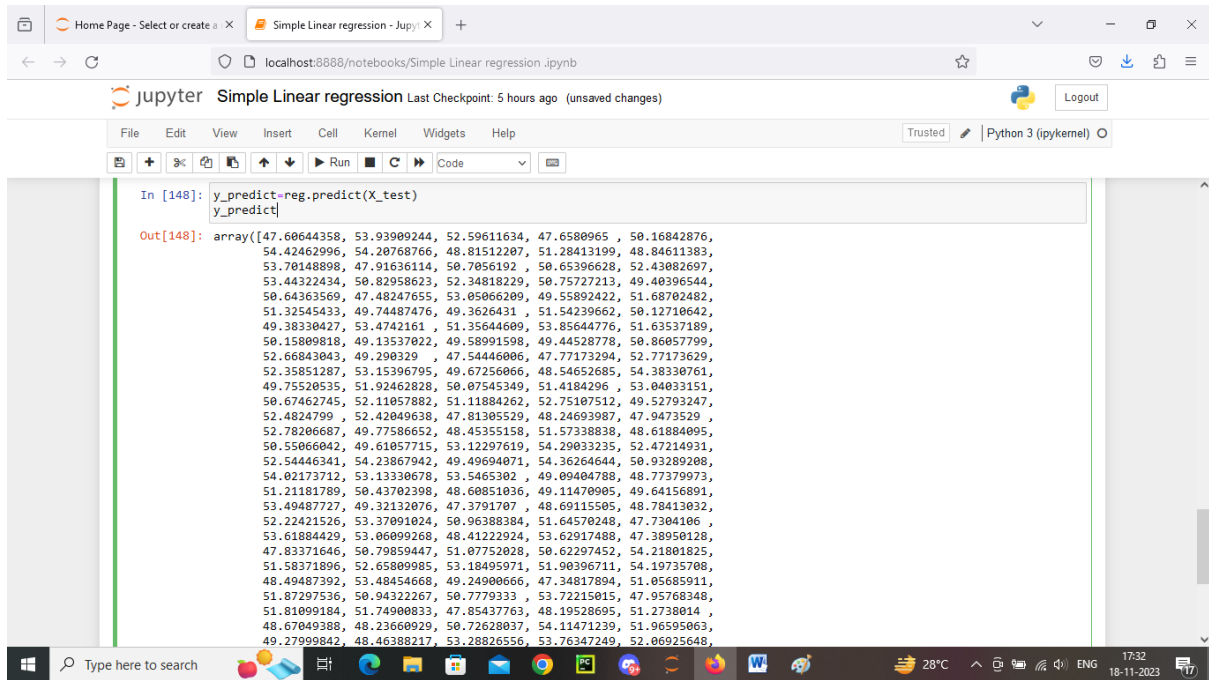
reg.fit(X_train,y_train)



A screenshot of a Jupyter Notebook titled "Simple Linear regression". The notebook is running on a local host (localhost:8888). The interface shows a menu bar with options like File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu bar, there are tabs for "Simple Linear regression - Jupyter" and "Simple Linear regression - Jupyter". The notebook content shows four code cells. The first cell (In [145]) contains the code: `# Let work with linear regression method` and `from sklearn.linear_model import LinearRegression`. The second cell (In [146]) contains the code: `reg=LinearRegression()`. The third cell (In [147]) contains the code: `reg.fit(X_train,y_train)`. The output of the third cell (Out[147]) shows a LinearRegression object: `LinearRegression()`. The notebook is running on Python 3 (ipykernel).

y_predict=reg.predict(X_test)

y_predict



A screenshot of a Jupyter Notebook interface. The browser address bar shows 'localhost:8888/notebooks/Simple Linear regression .ipynb'. The notebook title is 'Simple Linear regression' with a 'Last Checkpoint: 5 hours ago (unsaved changes)' note. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar shows icons for file operations, running, and code execution. The code cell contains:

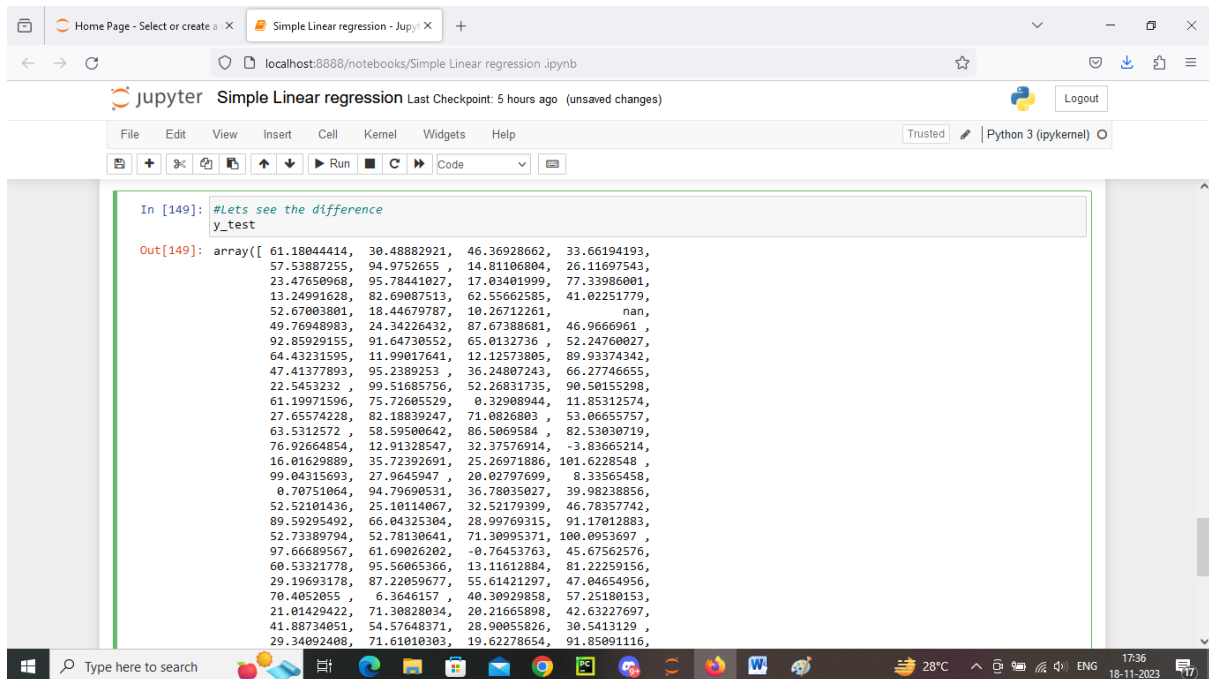
```
In [148]: y_predict=reg.predict(X_test)
y_predict
```

 The output cell shows a long array of predicted values:

```
Out[148]: array([47.60644358, 53.93989244, 52.59611634, 47.6580965 , 50.16842876,
54.42462996, 54.20768766, 48.81512207, 51.28413199, 48.84611383,
53.70148898, 47.91636114, 50.7056192 , 50.65396628, 52.43082697,
53.44322434, 50.82958623, 52.34818229, 50.75727213, 49.40396544,
50.64363569, 47.48247655, 53.05065209, 49.55892422, 51.68702482,
51.32545433, 49.74487476, 49.3626431 , 51.54239662, 50.12710642,
49.38330427, 53.4742161 , 51.35644609, 53.85644776, 51.63537189,
50.15809818, 49.13537022, 49.58991598, 49.44528778, 50.86057799,
52.66843043, 49.290329 , 47.54446006, 47.77173294, 52.77173629,
52.35851287, 53.15396795, 49.67256066, 48.54652685, 54.38330761,
49.75520535, 51.92462828, 50.07545349, 51.4184296 , 53.04033151,
50.67462745, 52.11057882, 51.11884262, 52.75107512, 49.52793247,
52.4824799 , 52.42049638, 47.81305529, 48.24693987, 47.9473529 ,
52.78206687, 49.77586652, 48.45355158, 51.57338838, 48.61804095,
50.55066042, 49.61057715, 53.12297619, 54.29033235, 52.47214931,
52.54446341, 54.23867942, 49.49694071, 54.36264644, 50.93289208,
54.02173712, 53.13330678, 53.5465302 , 49.09404788, 48.77379973,
51.21181789, 50.43702398, 48.60851036, 49.11470905, 49.64156891,
53.49487727, 49.32132076, 47.3791707 , 48.69115505, 48.78413032,
52.22421526, 53.37901024, 50.96388304, 51.64570248, 47.7304106 ,
53.61804429, 53.06099260, 48.41229204, 53.62917488, 47.38050128,
47.83371646, 50.79859447, 51.07752028, 50.62297452, 54.21801825,
51.58371896, 52.65809985, 53.18495971, 51.90396711, 54.19735708,
48.49487392, 53.48454668, 49.24900666, 47.34817894, 51.05685911,
51.87207536, 50.94322267, 50.7779333 , 53.72215015, 47.95768348,
51.81099184, 51.74900833, 47.85437763, 48.19528695, 51.2738014 ,
48.67049388, 48.23660929, 50.72628037, 54.11471239, 51.96595063,
49.27999842, 48.46388217, 53.28826556, 53.76347249, 52.06925648,
```

#Lets see the difference

y_test



A screenshot of a Jupyter Notebook interface. The browser address bar shows 'localhost:8888/notebooks/Simple Linear regression .ipynb'. The notebook title is 'Simple Linear regression' with a 'Last Checkpoint: 5 hours ago (unsaved changes)' note. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar shows icons for file operations, running, and code execution. The code cell contains:

```
In [149]: #Lets see the difference
y_test
```

 The output cell shows a long array of test values:

```
Out[149]: array([ 61.18044414, 30.48882921, 46.36928662, 33.66194193,
57.53887255, 94.9752655 , 14.81106004, 26.11697543,
23.47650968, 95.78441027, 17.03401999, 77.33086001,
13.24991628, 82.69087513, 62.55662585, 41.02251779,
52.67003801, 18.44679787, 10.26712261, nan,
49.76948983, 24.34226432, 87.67388681, 46.9666961 ,
92.85929155, 91.64730552, 65.0132736 , 52.24760027,
64.43231595, 11.99017641, 12.12573805, 89.93374342,
47.41377893, 95.2389253 , 36.24807243, 66.27746655,
22.5453232 , 99.51685756, 52.26831735, 90.50155298,
61.19971596, 75.72605529, 0.32908944, 11.85312574,
27.65574228, 82.18839247, 71.08268003, 53.06655757,
63.5312572 , 58.59500642, 86.5069584 , 82.53030719,
76.92664854, 12.91328547, 32.37576914, -3.83665214,
16.01629889, 35.72392691, 25.26971886, 101.6228548 ,
99.04315693, 27.9645947 , 20.02797699, 8.33565458,
0.70751064, 94.79690531, 36.78035827, 39.98238856,
52.52101436, 25.10114067, 32.52179399, 46.78357742,
89.59295492, 66.04325304, 28.99769315, 91.17012883,
52.73389794, 52.78130641, 71.30995371, 100.0953697 ,
97.66689567, 61.69026202, -0.76453763, 45.67562576,
60.53321778, 95.56065366, 13.11612884, 81.22259156,
29.19693178, 87.22059677, 55.61421297, 47.04654956,
70.4052055 , 6.3646157 , 40.30929858, 57.25180153,
21.01429422, 71.30828034, 20.21665898, 42.63227697,
41.88734051, 54.57648371, 28.90055826, 30.5413129 ,
29.34092488, 71.61010303, 19.62278654, 91.85091116,
```

```
plt.scatter(X_train,y_train,color='blue')

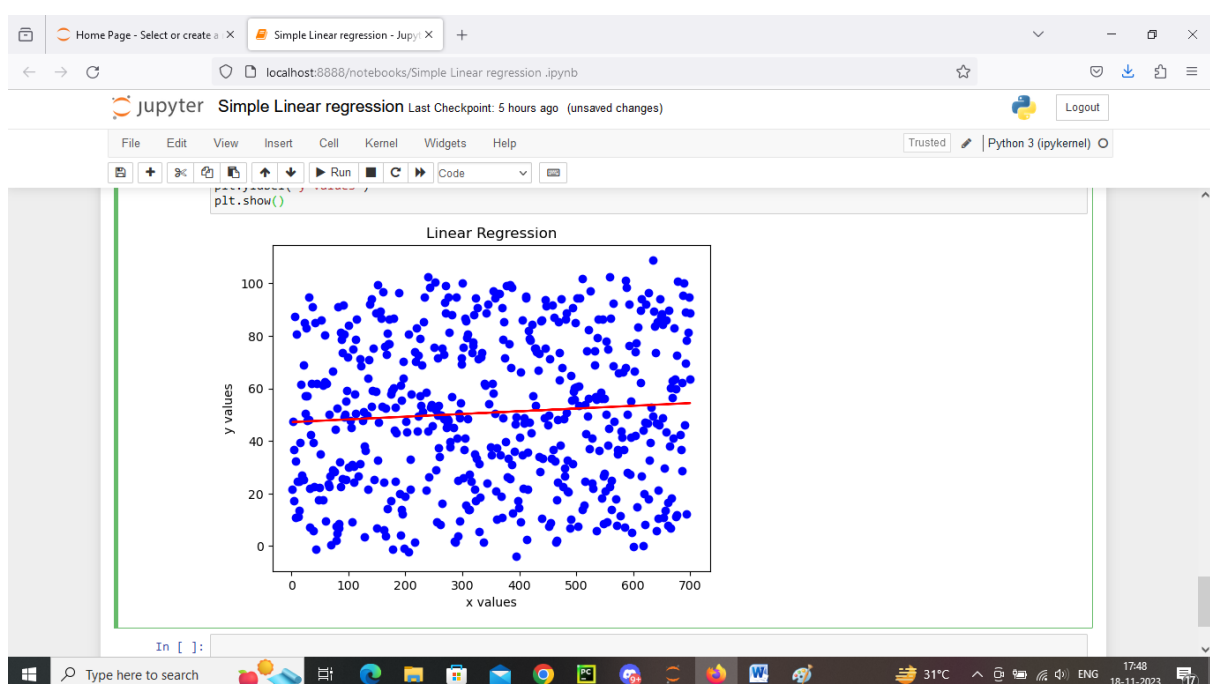
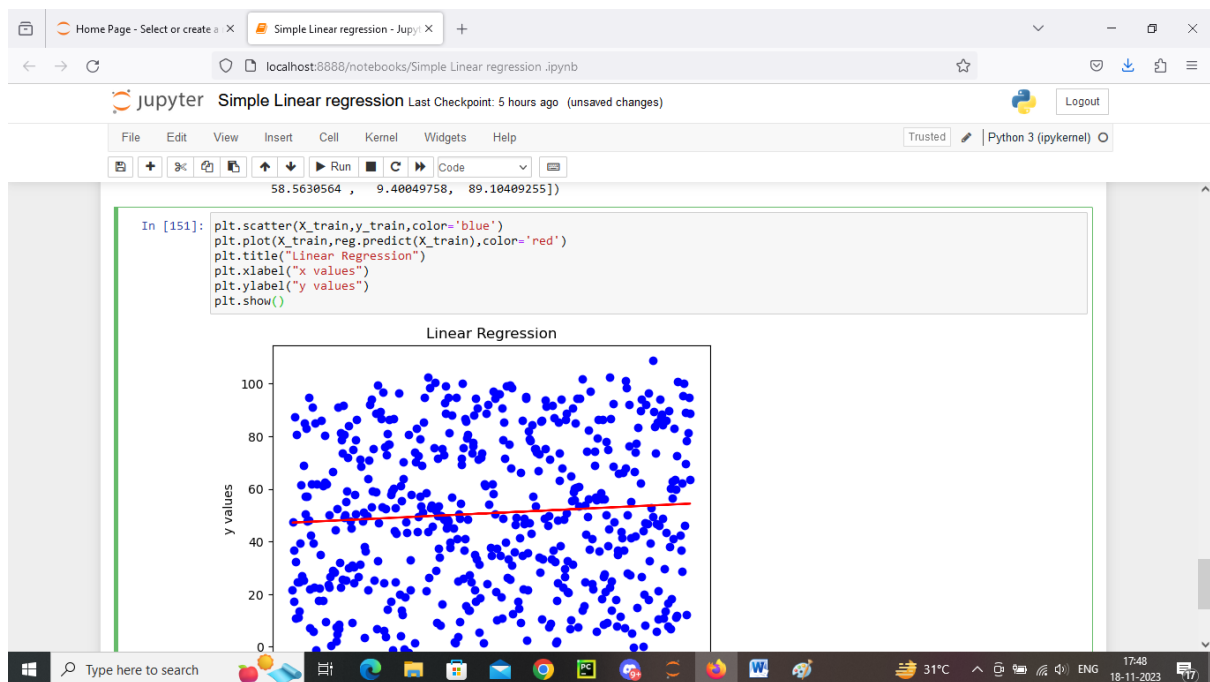
plt.plot(X_train,reg.predict(X_train),color='red')

plt.title("Linear Regression")

plt.xlabel("x values")

plt.ylabel("y values")

plt.show()
```



1. Train a simple linear regressing model on dataset and predict the output.

Test dataset :- https://docs.google.com/spreadsheets/d/e/2PACX-1vRyvZ7lknwiSghK9aen1SaTEYoN3JS40rrGLpcyrsVZy1tB2T4gn6Y_3-cdzPUFCPMmmqREWefW3kl4_/pubhtml

```
import pandas as pd
```

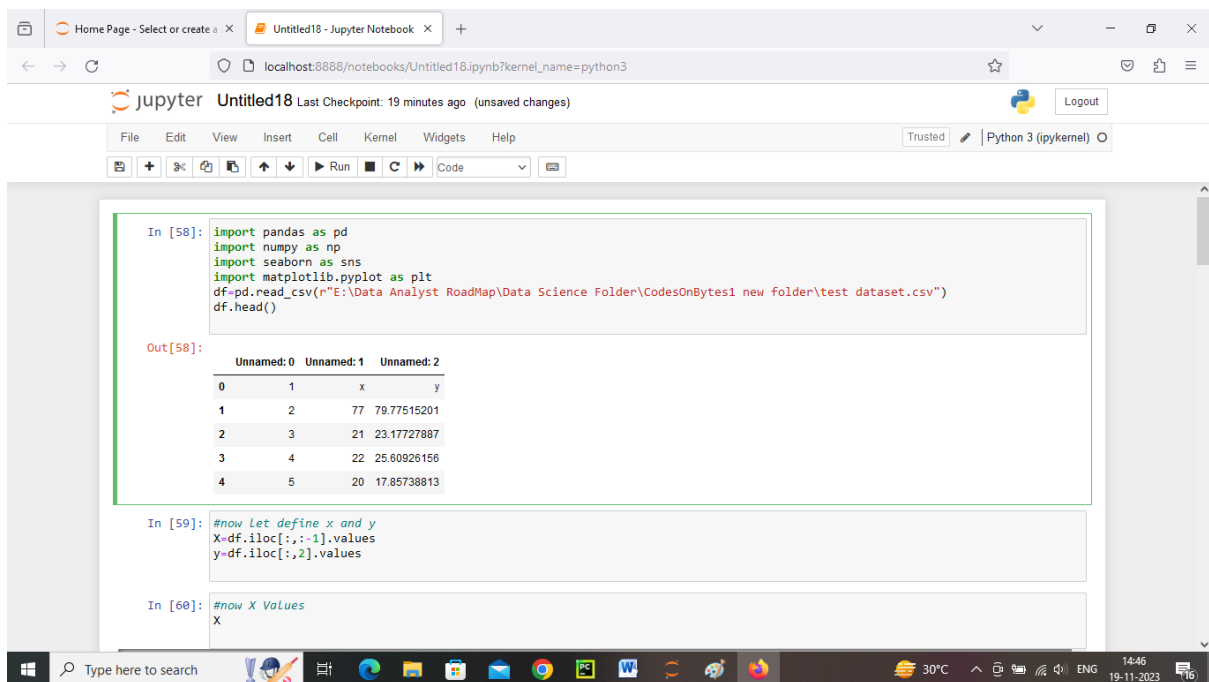
```
import numpy as np
```

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
df=pd.read_csv(r"E:\Data Analyst RoadMap\Data Science Folder\CodesOnBytes1 new folder\test dataset.csv")
```

```
df.head()
```



The screenshot shows a Jupyter Notebook window titled 'Untitled18'. The code in cell [58] imports pandas, numpy, seaborn, and matplotlib, then reads a CSV file and displays its head. The output shows a table with 5 rows and 4 columns. Below the code, there are two more cells: [59] for defining X and y, and [60] for displaying X values.

```
In [58]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv(r"E:\Data Analyst RoadMap\Data Science Folder\CodesOnBytes1 new folder\test dataset.csv")
df.head()

Out[58]:
```

	Unnamed: 0	Unnamed: 1	Unnamed: 2	
0	1	x	y	
1	2	77	79.77515201	
2	3	21	23.17727887	
3	4	22	25.60926156	
4	5	20	17.85738813	

```
In [59]: #now let define x and y
X=df.iloc[:, :-1].values
y=df.iloc[:, 2].values

In [60]: #now X Values
X
```

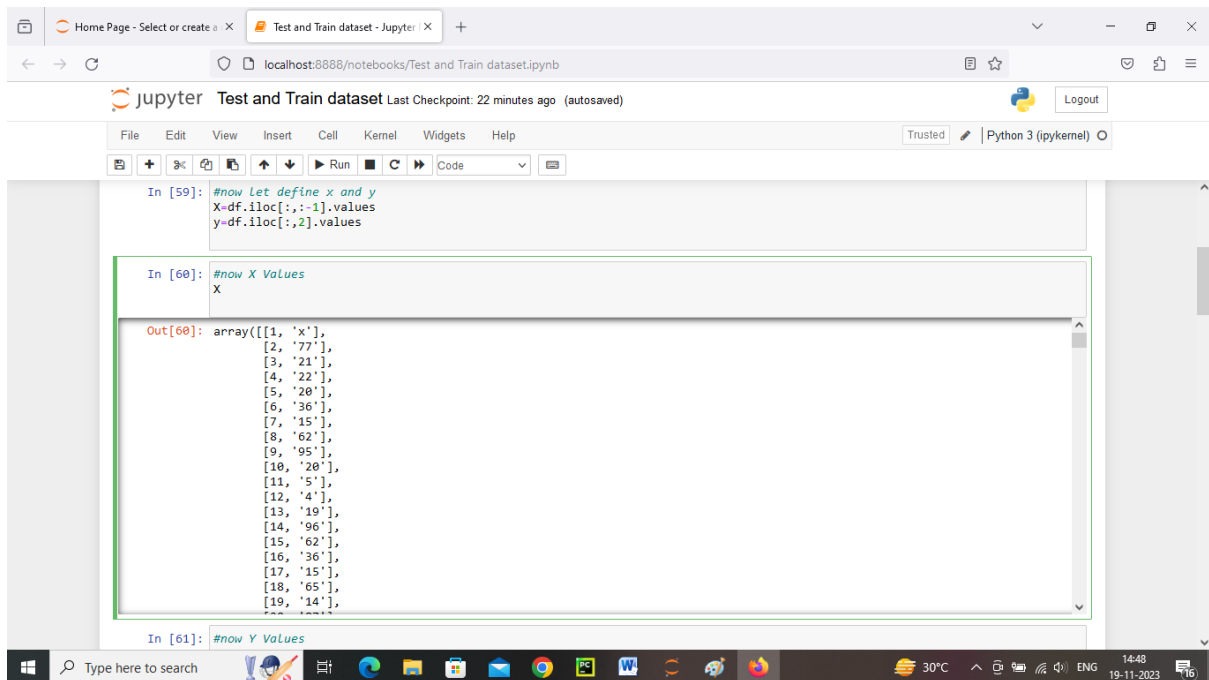
```
#now let define x and y
```

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

```
y=df.iloc[:, 2].values
```

#now X Values

X



A screenshot of a Jupyter Notebook interface. The browser address bar shows 'localhost:8888/notebooks/Test and Train dataset.ipynb'. The notebook title is 'Test and Train dataset' with a last checkpoint of 22 minutes ago. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The code area shows two input cells. The first cell (In [59]) defines variables x and y from a DataFrame df. The second cell (In [60]) is titled '#now X Values' and contains the variable X. The output of cell [60] is displayed as an array of 20 pairs, each containing an index and a string value for 'x'.

```
In [59]: #now let define x and y
X=df.iloc[:,1].values
y=df.iloc[:,2].values

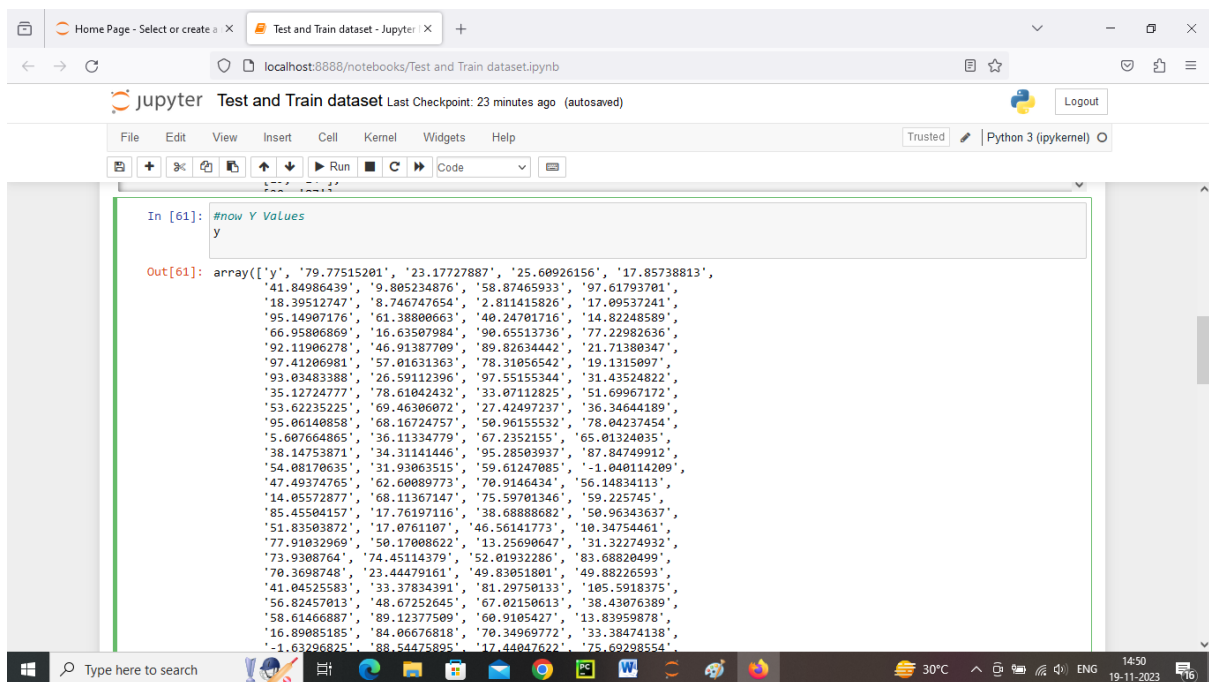
In [60]: #now X Values
X

Out[60]: array([[1, 'x'],
                [2, '77'],
                [3, '21'],
                [4, '22'],
                [5, '20'],
                [6, '36'],
                [7, '15'],
                [8, '62'],
                [9, '95'],
                [10, '20'],
                [11, '5'],
                [12, '4'],
                [13, '19'],
                [14, '96'],
                [15, '62'],
                [16, '36'],
                [17, '15'],
                [18, '65'],
                [19, '14'],
                [20, '14']])

In [61]: #now Y Values
```

#now Y Values

Y



A screenshot of a Jupyter Notebook interface, similar to the one above. The browser address bar shows 'localhost:8888/notebooks/Test and Train dataset.ipynb'. The notebook title is 'Test and Train dataset' with a last checkpoint of 23 minutes ago. The code area shows an input cell (In [61]) titled '#now Y Values' containing the variable y. The output of cell [61] is displayed as an array of 20 pairs, each containing an index and a float value for 'y'.

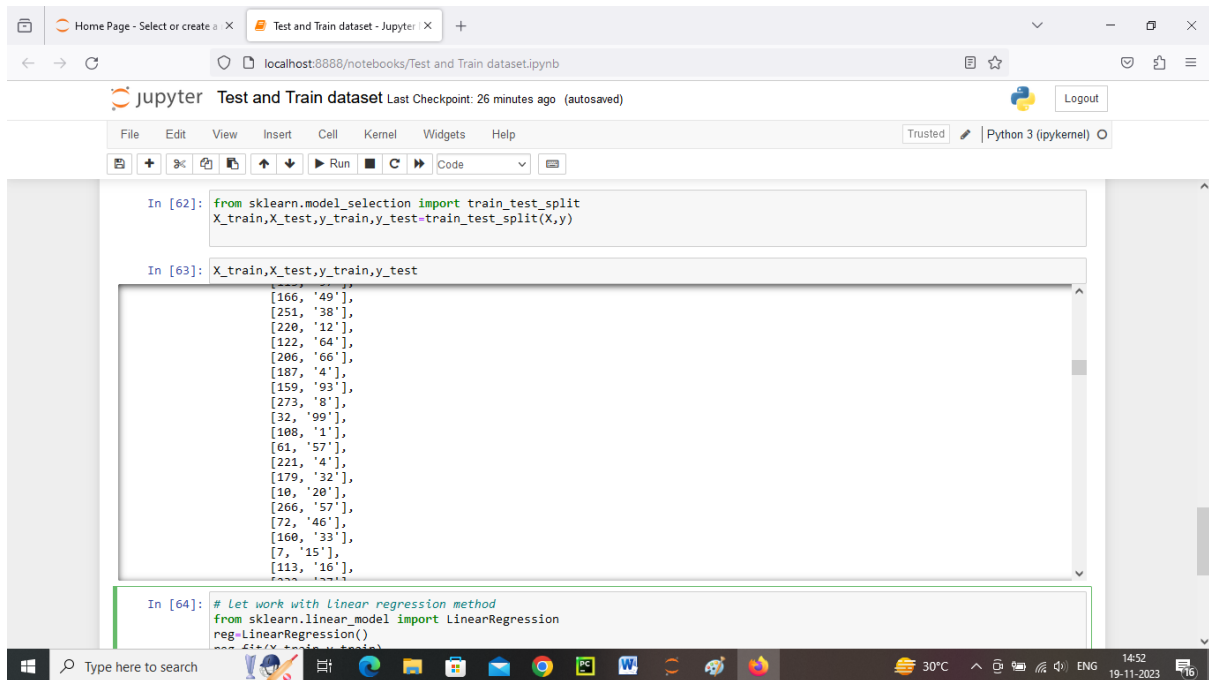
```
In [61]: #now Y Values
y

Out[61]: array(['y', '79.77515201', '23.17727887', '25.60926156', '17.85738813',
                '41.84986439', '9.885234876', '58.87465933', '97.61793781',
                '18.39512747', '8.746747654', '2.811415826', '17.09537241',
                '95.14907176', '61.38800663', '40.24701716', '14.82248589',
                '66.95806869', '16.63507984', '90.65513736', '77.22982636',
                '92.11906278', '46.91387709', '89.82634442', '21.71380347',
                '97.41206981', '57.81631363', '78.31056542', '19.1315097',
                '93.03483388', '26.59112396', '97.55155344', '31.43524822',
                '35.12724777', '78.61042432', '33.07112825', '51.69967172',
                '53.62235225', '69.46306072', '27.42497237', '36.34644189',
                '95.06140858', '68.16724757', '50.96155532', '78.04237454',
                '5.607664865', '36.11334779', '67.2352155', '65.01324035',
                '38.14753871', '34.31141446', '95.28503937', '87.84749912',
                '54.08170635', '31.93063515', '59.61247085', '-1.040114209',
                '47.49374765', '62.60089773', '70.9146434', '56.14834113',
                '14.05572877', '68.11367147', '75.59701346', '59.225745',
                '85.45504157', '17.76197116', '38.68888682', '50.96343637',
                '51.83503872', '17.0761107', '46.56141773', '10.34754461',
                '77.91032969', '50.17008622', '13.25690647', '31.32774932',
                '73.9308764', '74.45114379', '52.01932286', '83.68820499',
                '70.3690748', '23.44479161', '49.83051001', '49.88226593',
                '41.04525583', '33.37834391', '81.29750133', '105.5018375',
                '56.82457013', '48.67252645', '67.02150613', '38.43076389',
                '58.61466887', '89.12377509', '60.9105427', '13.83959878',
                '16.89085185', '84.06676818', '70.34969772', '33.38474138',
                '-1.63296825', '88.54475895', '17.44047622', '75.69298554'])
```

```
from sklearn.model_selection import train_test_split
```

```
X_train,X_test,y_train,y_test=train_test_split(X,y)
```

```
X_train,X_test,y_train,y_test
```



The screenshot shows a Jupyter Notebook titled "Test and Train dataset" with the following code and output:

```
In [62]: from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y)
```

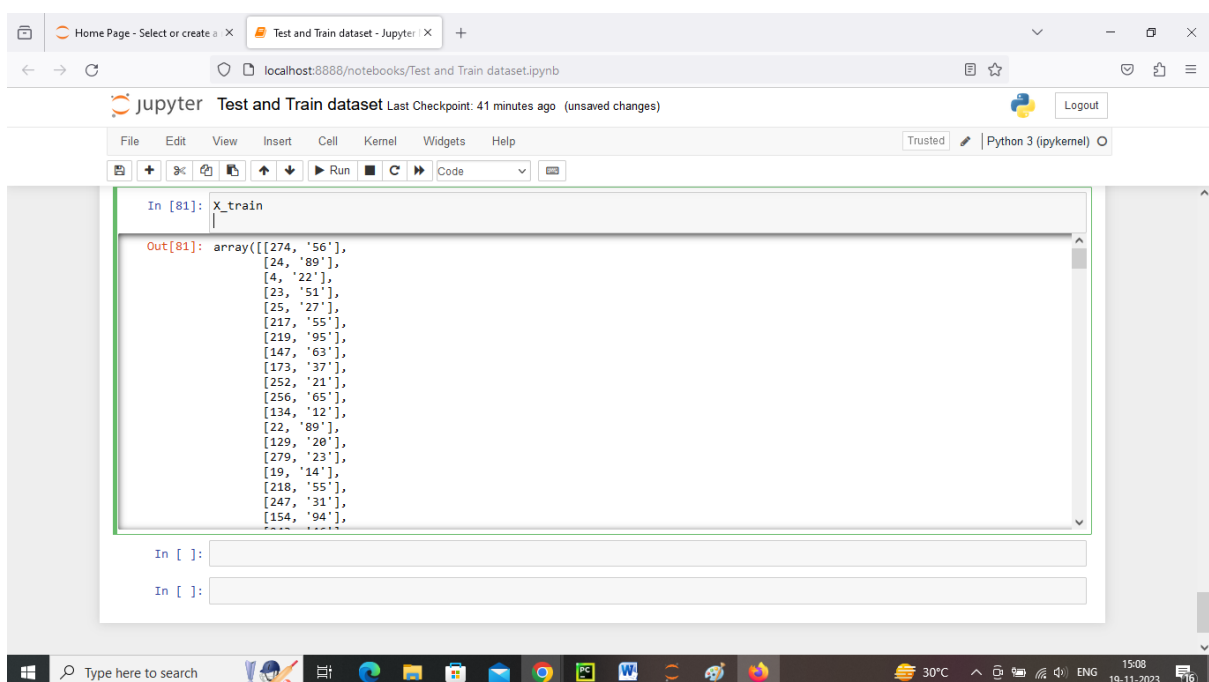
```
In [63]: X_train,X_test,y_train,y_test
```

```
Out[63]:
```

```
[[166, '49'],
 [251, '38'],
 [220, '12'],
 [122, '64'],
 [206, '66'],
 [187, '4'],
 [159, '93'],
 [273, '8'],
 [32, '99'],
 [108, '1'],
 [61, '57'],
 [221, '4'],
 [179, '32'],
 [10, '20'],
 [266, '57'],
 [72, '46'],
 [160, '33'],
 [7, '15'],
 [113, '16'],
 ...]
```

```
In [64]: # Let work with linear regression method
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(X_train,y_train)
```

```
X_train
```



The screenshot shows a Jupyter Notebook titled "Test and Train dataset" with the following code and output:

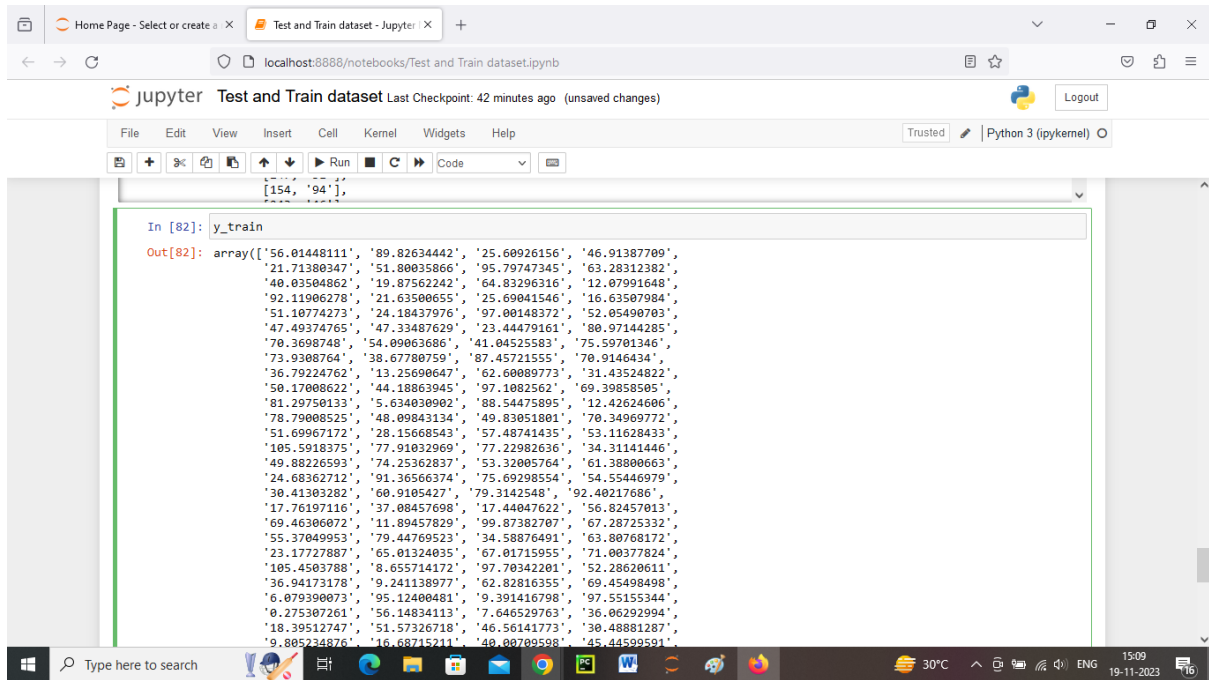
```
In [81]: X_train
```

```
Out[81]: array([[274, '56'],
 [24, '89'],
 [4, '22'],
 [23, '51'],
 [25, '27'],
 [217, '55'],
 [219, '95'],
 [147, '63'],
 [173, '37'],
 [252, '21'],
 [256, '65'],
 [134, '12'],
 [22, '89'],
 [129, '20'],
 [279, '23'],
 [19, '14'],
 [218, '55'],
 [247, '31'],
 [154, '94'],
 ...])
```

```
In [ ]:
```

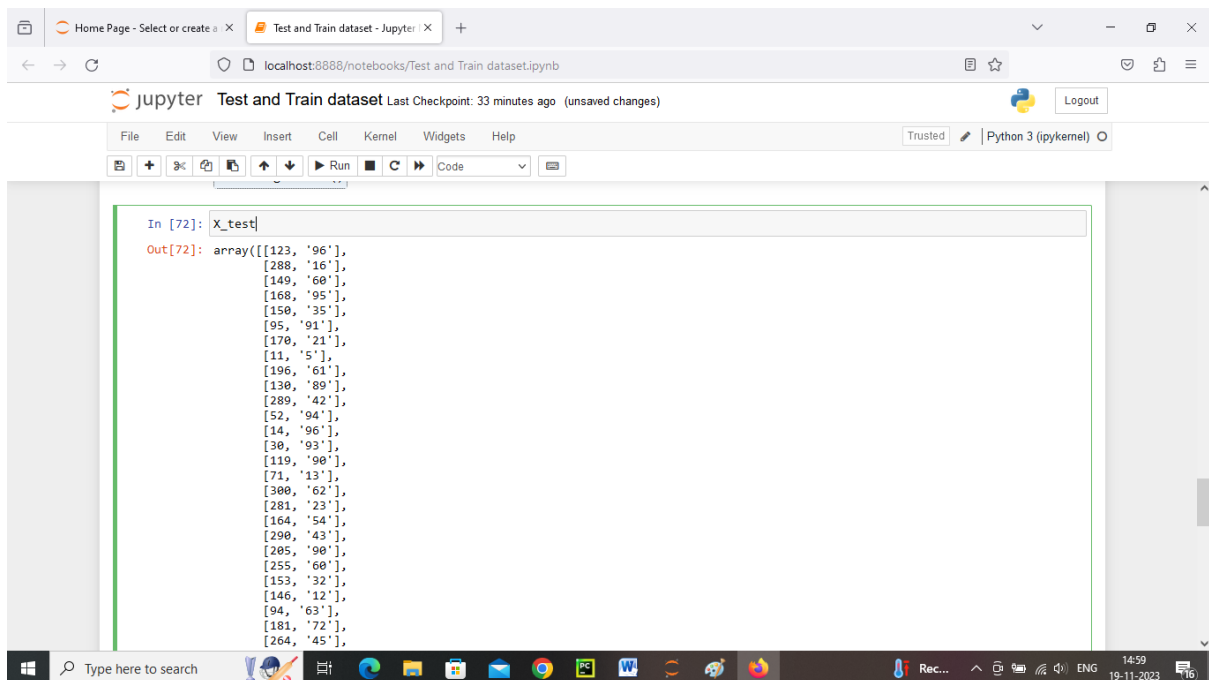
```
In [ ]:
```


y_train



```
In [82]: y_train
Out[82]: array(['56.01448111', '89.82634442', '25.60926156', '46.91387789',
                '21.71380347', '51.80035866', '95.79747345', '63.28312382',
                '40.03504862', '19.87562242', '64.83296316', '12.07991648',
                '92.11906278', '21.63500655', '25.69041546', '16.63507984',
                '51.10774273', '24.18437976', '97.00148372', '52.05490783',
                '47.49374765', '47.33487629', '23.44479161', '80.97144285',
                '70.3698748', '54.09063686', '41.04525583', '75.59701346',
                '73.9308764', '38.67780759', '87.45721555', '70.9146434',
                '36.79224762', '13.25690647', '62.60089773', '31.43524822',
                '50.17008622', '44.18863945', '97.1082562', '69.39858905',
                '81.29750133', '5.634030902', '88.54475895', '12.42624606',
                '78.79008525', '48.09843134', '49.83051801', '70.34969772',
                '51.69967172', '28.15668543', '57.48741435', '53.11628433',
                '105.5918375', '77.91032969', '77.22982636', '34.31141446',
                '49.88226593', '74.25362837', '53.32005764', '61.38000663',
                '24.68362712', '91.36566374', '75.69298554', '54.55446979',
                '30.41303282', '60.9105427', '79.3142548', '92.40217686',
                '17.76197116', '37.08457698', '17.44047622', '56.82457013',
                '69.46300072', '11.89457829', '99.87382707', '67.28725332',
                '55.37049953', '79.44769523', '34.58876491', '63.80768172',
                '23.17272807', '65.01324035', '67.01715955', '71.00377824',
                '105.4569780', '8.655714172', '97.70342201', '52.28620611',
                '36.94137178', '9.241138977', '62.82816355', '69.45498498',
                '6.079390073', '95.12400481', '9.391416798', '97.55155344',
                '0.275307261', '56.14834113', '7.646529763', '36.06292994',
                '18.39512747', '51.57326718', '46.56141773', '30.48881287',
                '9.805234876', '16.68715211', '40.00709598', '45.44599591',
```

X_test



```
In [72]: X_test
Out[72]: array([[123, '96'],
                [288, '16'],
                [149, '60'],
                [168, '95'],
                [150, '35'],
                [95, '91'],
                [170, '21'],
                [11, '5'],
                [196, '61'],
                [130, '89'],
                [289, '42'],
                [52, '94'],
                [14, '96'],
                [30, '93'],
                [119, '90'],
                [71, '13'],
                [300, '62'],
                [281, '23'],
                [164, '54'],
                [290, '43'],
                [205, '90'],
                [255, '60'],
                [153, '32'],
                [146, '12'],
                [94, '63'],
                [181, '72'],
                [264, '45'],
```

y_test

Home Page - Select or create a x Test and Train dataset - Jupyter x +

localhost:8888/notebooks/Test and Train dataset.ipynb

jupyter Test and Train dataset Last Checkpoint: 34 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (pykernel)

Type Markdown and LaTeX: α^2

```
In [73]: y_test
Out[73]: array(['97.27405461', '16.74923937', '63.19354354', '97.13832018',
               '34.8534823', '89.12377509', '18.98322306', '8.746747654',
               '60.07137496', '91.59548851', '41.46923883', '95.28503937',
               '95.14907176', '93.03483388', '95.15647284', '17.0761107',
               '63.29717058', '18.31396758', '52.45103628', '42.84526108',
               '92.88772282', '60.11134958', '29.38505024', '12.11219941',
               '58.61466887', '74.68953276', '44.68913433', '89.73951993',
               '81.40381769', '39.3294153', '73.2300846', '21.42637785',
               '21.16523945', '-2.819913974', '29.31770045', '67.38654703',
               '78.61042432', '31.32274932', '72.86282528', '68.5458879',
               '59.8313966', '16.89085185', '68.94699774', '67.2352155',
               '66.95806869', '59.10598995', '79.50341495', '26.48713683',
               '97.41206981', '62.22635684', '41.84986439', '78.04237454',
               '35.12724777', 'y', '37.80182795', '2.576625376', '26.49487961',
               '41.90360917', '-2.761182595', '98.03017721', '12.59244741',
               '21.32273728', '25.75612514', '26.59112396', '36.34644189',
               '10.34754461', '46.69013968', '33.07112825', '19.87846479',
               '36.11334779', '11.21710477', '30.49412933', '85.45504157',
               '59.225745', '59.07247174', '87.84749912'], dtype=object)
```

In []:

Type here to search 30°C 15:01 19-11-2023

reg.coef_

Home Page - Select or create a x Test and Train dataset - Jupyter x +

localhost:8888/notebooks/Test and Train dataset.ipynb

jupyter Test and Train dataset Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (pykernel)

```
'33.37834391'], dtype=object)

In [91]: reg.coef_
Out[91]: array([9.77147252e-04, 1.01337027e+00])

In [92]: reg.intercept_
Out[92]: -0.5979316663393348

In [99]:

-----
ValueError                                Traceback (most recent call last)
Cell In[99], line 1
----> 1 reg.predict(X_test)

File ~\anaconda3\Lib\site-packages\sklearn\linear_model\_base.py:386, in LinearModel.predict(self, X)
    372 def predict(self, X):
    373     """
    374     Predict using the linear model.
    375
    (...)
    384     Returns predicted values.
    385     """
--> 386     return self._decision_function(X)

File ~\anaconda3\Lib\site-packages\sklearn\linear_model\_base.py:369, in LinearModel._decision_function(self, X)
    369 def _decision_function(self, X):
    370     """
    371     Compute the decision function of the samples.
    372     """
    373     X = self._validate_data(X, accept_sparse=['csr', 'csc', 'coo'],
    374                             dtype=[float, double], order='C',
    375                             ensure_min_samples=1, ensure_min_features=1,
    376                             force_all_finite=True)
    377     # Check for non-regression case
    378     if self._is_fitted:
    379         # Compute the dot product of the features and the coefficients
    380         # (X @ self.coef_)
    381         # Note: self.coef_ is a 1D array, so we need to reshape it
    382         # to a 2D array to perform the dot product
    383         return X @ self.coef_
    384     else:
    385         raise ValueError("This model has not been fitted. Call fit() before using this method.")
```

Type here to search 30°C 15:19 19-11-2023

reg.intercept_

The screenshot shows a Jupyter Notebook titled "Test and Train dataset" with a last checkpoint of "an hour ago (unsaved changes)". The notebook is running on a Python 3 (ipykernel) environment. The code in the first cell defines a dataset of 15 data points as a list of lists, where each inner list contains a feature value and a target value. The second cell shows the output of the first two steps of a linear regression model fit: the coefficients and the intercept.

```
'38.14753871', '50.96155532', '77.9969477', '72.9138853',  
'-2.344738542', '91.69240746', '23.52647153', '7.468501839',  
'59.12912974', '55.7383467', '97.61793701', '5.405220518',  
'84.06676818', '70.34329706', '39.31485292', '13.12109842',  
'96.58808601', '95.3958003', '52.45946688', '27.42497237',  
'13.41310757', '46.11021062', '67.22008001', '68.19721905',  
'33.37834391'], dtype=object)
```

```
In [91]: reg.coef_  
Out[91]: array([9.77147252e-04, 1.01337027e+00])
```

```
In [92]: reg.intercept_  
Out[92]: -0.5979316663393348
```

```
In [ ]:  
In [ ]:
```

let work with linear regression method

from sklearn.linear_model import LinearRegression

reg=LinearRegression()

reg.fit(X_train,y_train)

The screenshot displays a Jupyter Notebook environment within a web browser. The browser's address bar shows the local host address: `localhost:8888/notebooks/Test and Train dataset.ipynb`. The Jupyter interface includes a top menu bar with options like File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with various icons for file operations and code execution. The main area of the notebook shows a code cell labeled `In [64]:` with the following Python code:

```
# Let work with Linear regression method
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(X_train,y_train)
```

Below the code cell is an output cell labeled `Out[64]:` which displays the result of the code execution: `LinearRegression()`. The interface also shows a 'Logout' button in the top right corner and a 'Python 3 (ipykernel)' label in the bottom right corner of the notebook area.