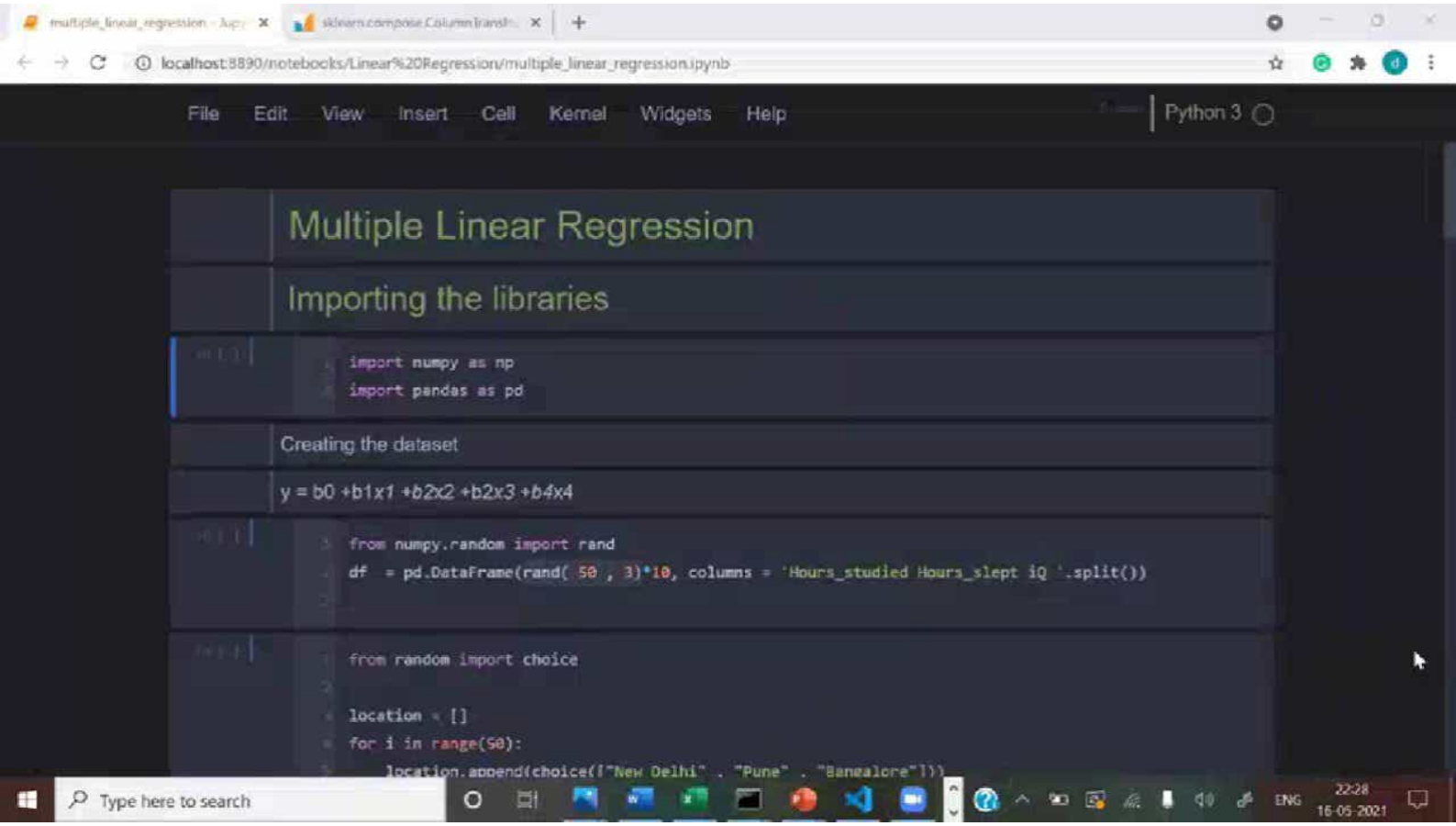
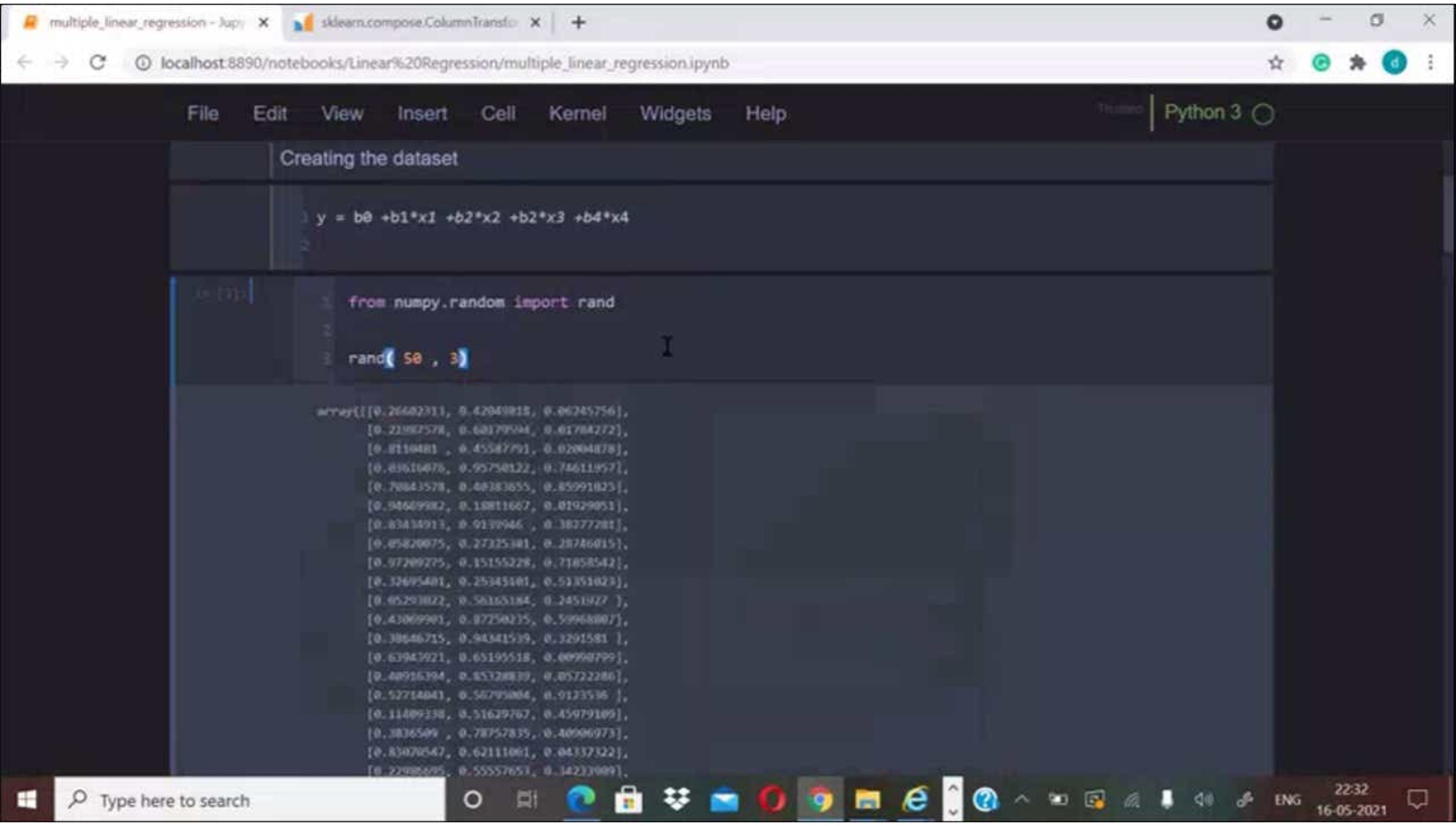
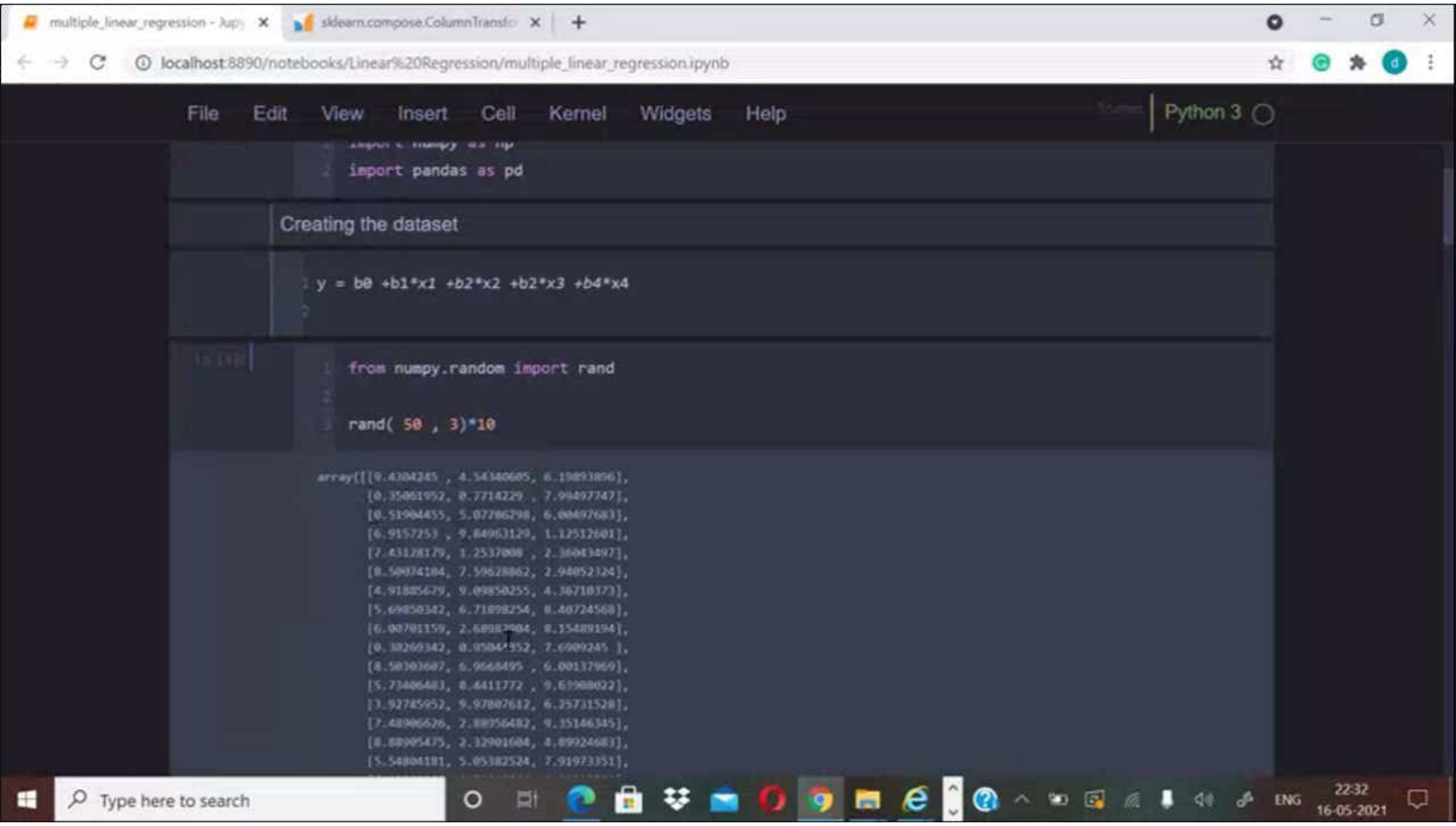
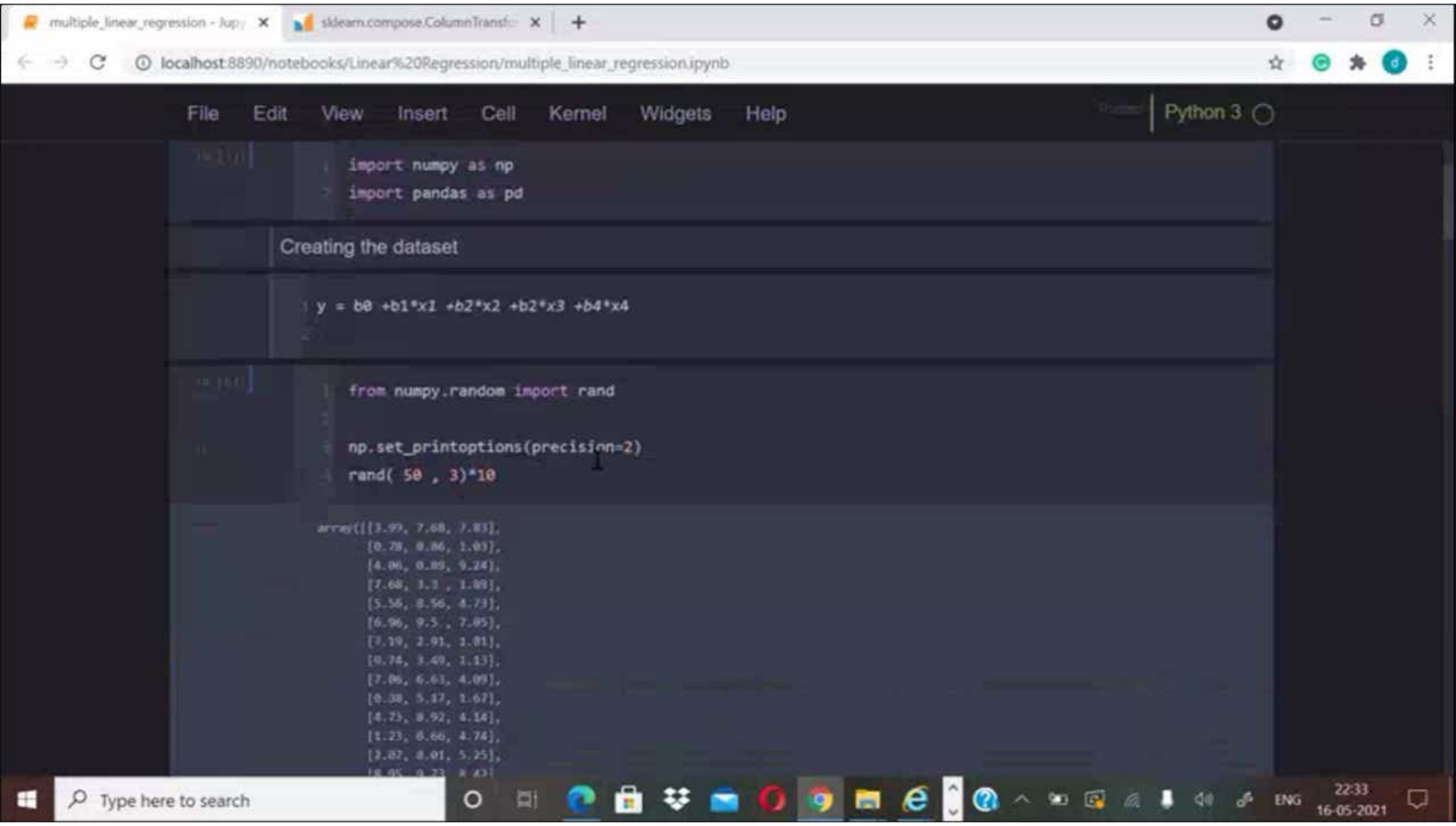
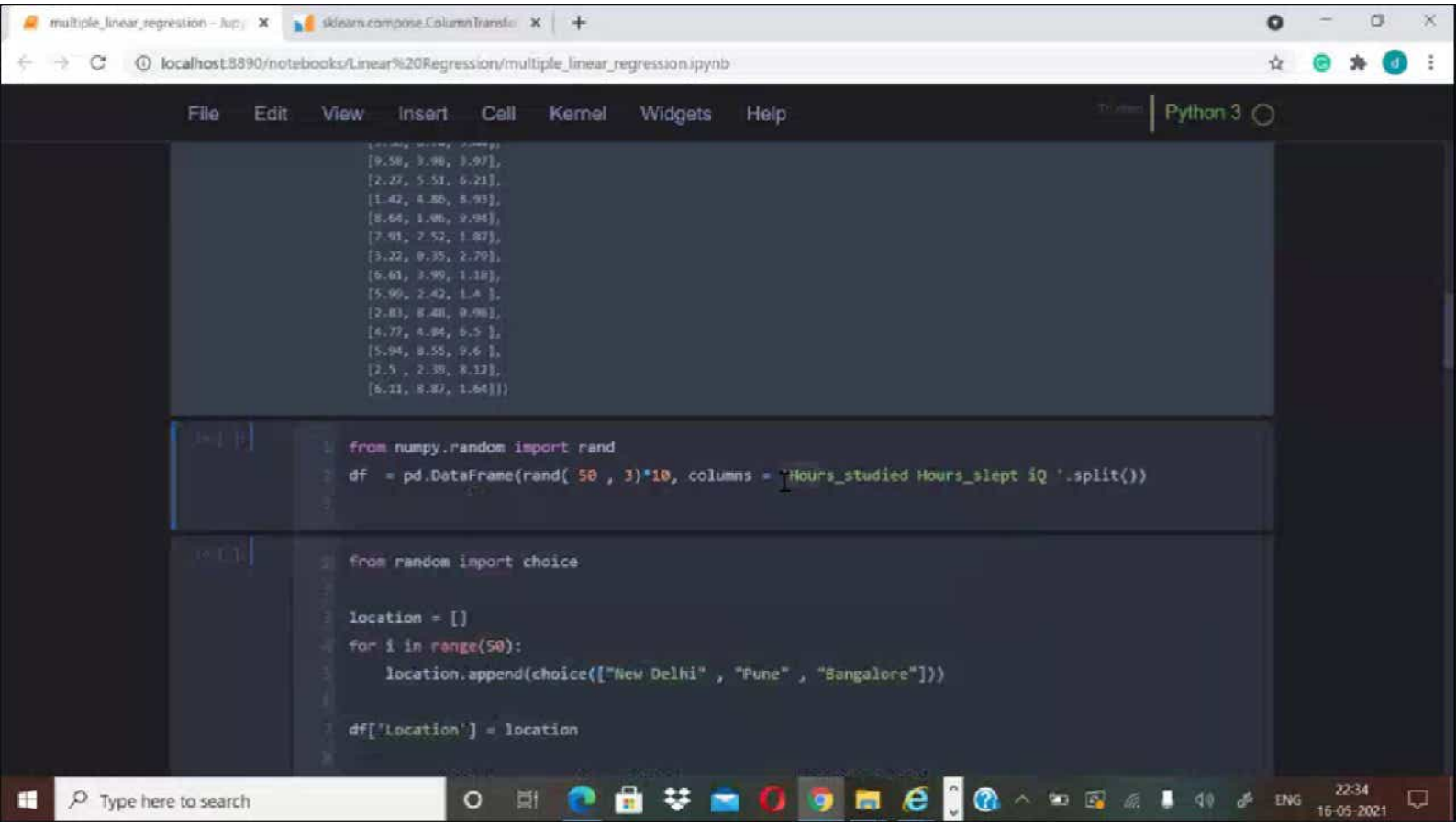

multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
Multiple Linear Regression
Importing the libraries
import numpy as np
import pandas as pd
Creating the dataset
y = b0 +b1x1 +b2x2 +b2x3 +b4x4
from numpy.random import rand
df = pd.DataFrame(rand(50 , 3)*10, columns = 'Hours_studied Hours_slept iQ '.split())
from random import choice
location = []
for i in range(50):
location.append(choice(["New Delhi" , "Pune" , "Bangalore"]))
Type here to search
ENG
22:28
16-05-2021


multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
Creating the dataset
y = b0 +b1*x1 +b2*x2 +b2*x3 +b4*x4
from numpy.random import rand
rand(50 , 3)
array([[0.26682311, 0.42049818, 0.06245756],
[0.21987578, 0.60179594, 0.61784272],
[0.8110481 , 0.45587791, 0.02004878],
[0.03616078, 0.95758122, 0.74611957],
[0.70843578, 0.40383655, 0.85991825],
[0.94669982, 0.18811667, 0.01929051],
[0.83438913, 0.9139946 , 0.38277281],
[0.05820075, 0.27325381, 0.28746015],
[0.97209275, 0.15155228, 0.71858542],
[0.32695401, 0.25345101, 0.51351823],
[0.05293822, 0.56365184, 0.2451927],
[0.43069901, 0.87250235, 0.59968807],
[0.38646715, 0.94341539, 0.3291581],
[0.63943921, 0.65195518, 0.00990799],
[0.40916394, 0.85320839, 0.05722286],
[0.52714041, 0.56795084, 0.9123536],
[0.11409338, 0.51629767, 0.45979109],
[0.3836509 , 0.78757835, 0.40006973],
[0.83070547, 0.62111061, 0.04337322],
Type here to search
ENG
22:32
16-05-2021


multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
import pandas as pd
Creating the dataset
y = b0 +b1*x1 +b2*x2 +b2*x3 +b4*x4
from numpy.random import rand
rand(50 , 3)*10
array([[9.4304245 , 4.54340685, 6.19893896],
[0.35061952, 0.7714229 , 7.99497747],
[0.51904455, 5.07286298, 6.004976831,
[6.9157253 , 9.84963129, 1.12532601],
[7.43128179, 1.2537008 , 2.36043497],
[8.50074104, 7.59628862, 2.94052324],
[4.91885679, 9.09850255, 4.36710373],
[5.69850342, 6.71898254, 8.48724568],
[6.00701159, 2.68083904, 8.15489194],
[0.38260342, 0.9504
[8.50303607, 6.9668495 , 6.00137969],
[5.73406483, 8.4411772 , 9.63988022],
[3.92745952, 9.97807612, 6.25731528],
[7.48906626, 2.88956482, 9.35146345],
[8.88905475, 2.32901604, 4.89924683],
[5.54804181, 5.05382524, 7.91973351],
Type here to search
ENG
22:32
16-05-2021


multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
import numpy as np
import pandas as pd
Creating the dataset
y = b0 +b1*x1 +b2*x2 +b2*x3 +b4*x4
from numpy.random import rand
np.set_printoptions(precision=2)
rand(50 , 3)*10
array([[3.99, 7.68, 7.83],
[0.78, 0.86, 1.03],
[4.06, 0.89, 9.24],
[7.68, 3.3 , 1.89],
[5.56, 0.56, 4.73],
[6.96, 9.5 , 7.05],
[7.19, 2.91, 1.01],
[0.74, 3.49, 1.13],
[7.06, 6.61, 4.09],
[0.38, 5.17, 1.67],
[4.75, 8.92, 4.14],
[1.23, 0.66, 4.74],
[2.02, 8.01, 5.25],
Type here to search
ENG
22:33
16-05-2021


multiple_linear_regression - Jup
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
[9.58, 3.98, 3.97],
[2.27, 5.51, 6.21],
[1.42, 4.86, 8.93],
[8.64, 1.06, 9.94],
[7.91, 7.52, 1.87],
[3.22, 0.35, 2.70],
[6.61, 3.99, 1.18],
[5.99, 2.42, 1.4],
[2.83, 8.48, 0.98],
[4.77, 4.84, 6.5],
[5.94, 8.55, 9.6],
[2.5 , 2.39, 8.12],
[6.11, 8.82, 1.64]])
from numpy.random import rand
df = pd.DataFrame(rand(50 , 3)*10, columns = 'Hours_studied Hours_slept IQ '.split())
from random import choice
location = []
for i in range(50):
location.append(choice(["New Delhi" , "Pune" , "Bangalore"]))
df['location'] = location
Type here to search
ENG
22:34
16-05-2021

multiple_linear_regression - Jupy | sklearn.compose.ColumnTransfo | +

localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb

File Edit View Insert Cell Kernel Widgets Help | Python 3

Creating the dataset

```
y = b0 +b1*x1 +b2*x2 +b2*x3 +b4*x4
```

```
'Hours_stud,ied ,Hou,rs_sl,ept ,iQ '.split(",")
```

```
['Hours_stud', 'ied ', 'Hou', 'rs_sl', 'ept ', 'iQ ']
```

```
from numpy.random import rand
df  = pd.DataFrame(rand( 50 , 3)*10, columns = 'Hours_studied Hours_slept iQ '.split())
df.head()
```

	Hours_studied	Hours_slept	iQ
0	1.881195	7.964956	9.075883
1	0.882992	6.609851	5.014331
2	6.529097	7.474048	9.479220
3	6.487647	6.294339	8.660800
4	3.793140	4.324589	1.179982

```
from random import choice
```

Type here to search

ENG 22:36 16-05-2021

multiple_linear_regression - Jupy | sklearn.compose.ColumnTransfo

localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb

File Edit View Insert Cell Kernel Widgets Help Python 3

```
df.head(2)
```

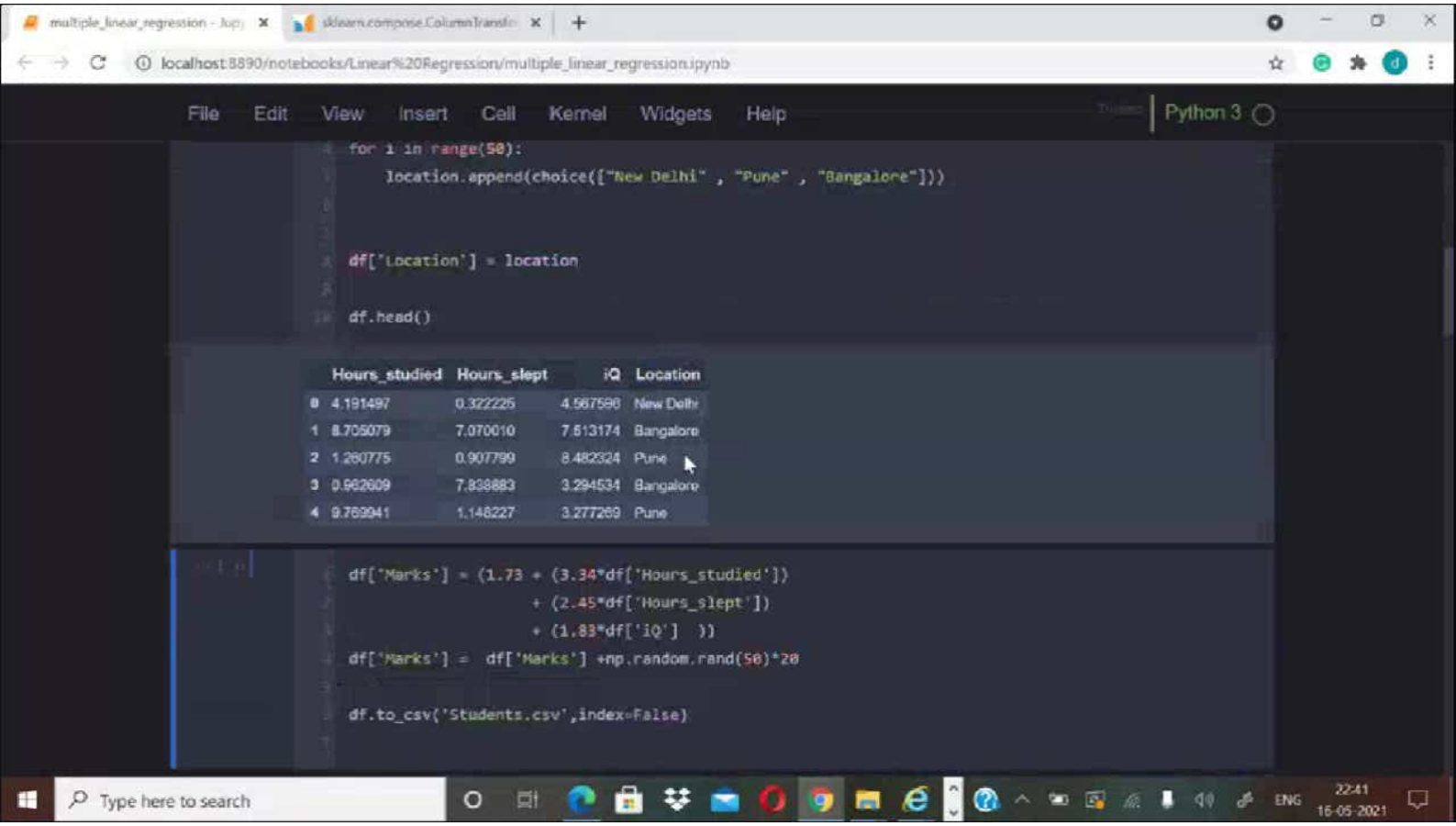
	Hours_studied	Hours_slept	iQ
0	5.391161	3.953001	3.620444
1	5.490866	3.917475	3.691032

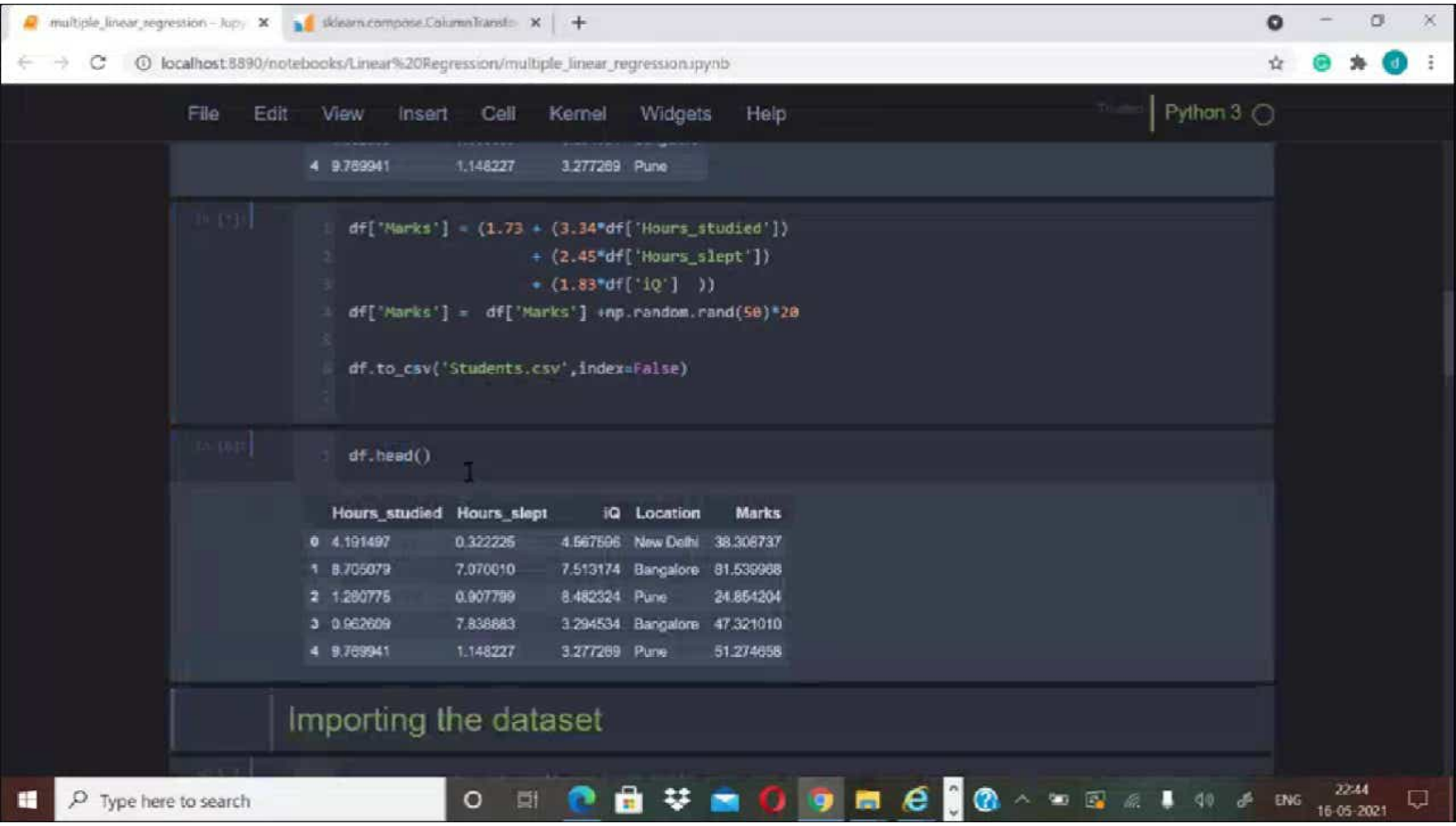
```
from random import choice

location = []
for i in range(50):
    location.append(choice(["New Delhi" , "Pune", "Bangalore"]))

df['Location'] = location

df
```

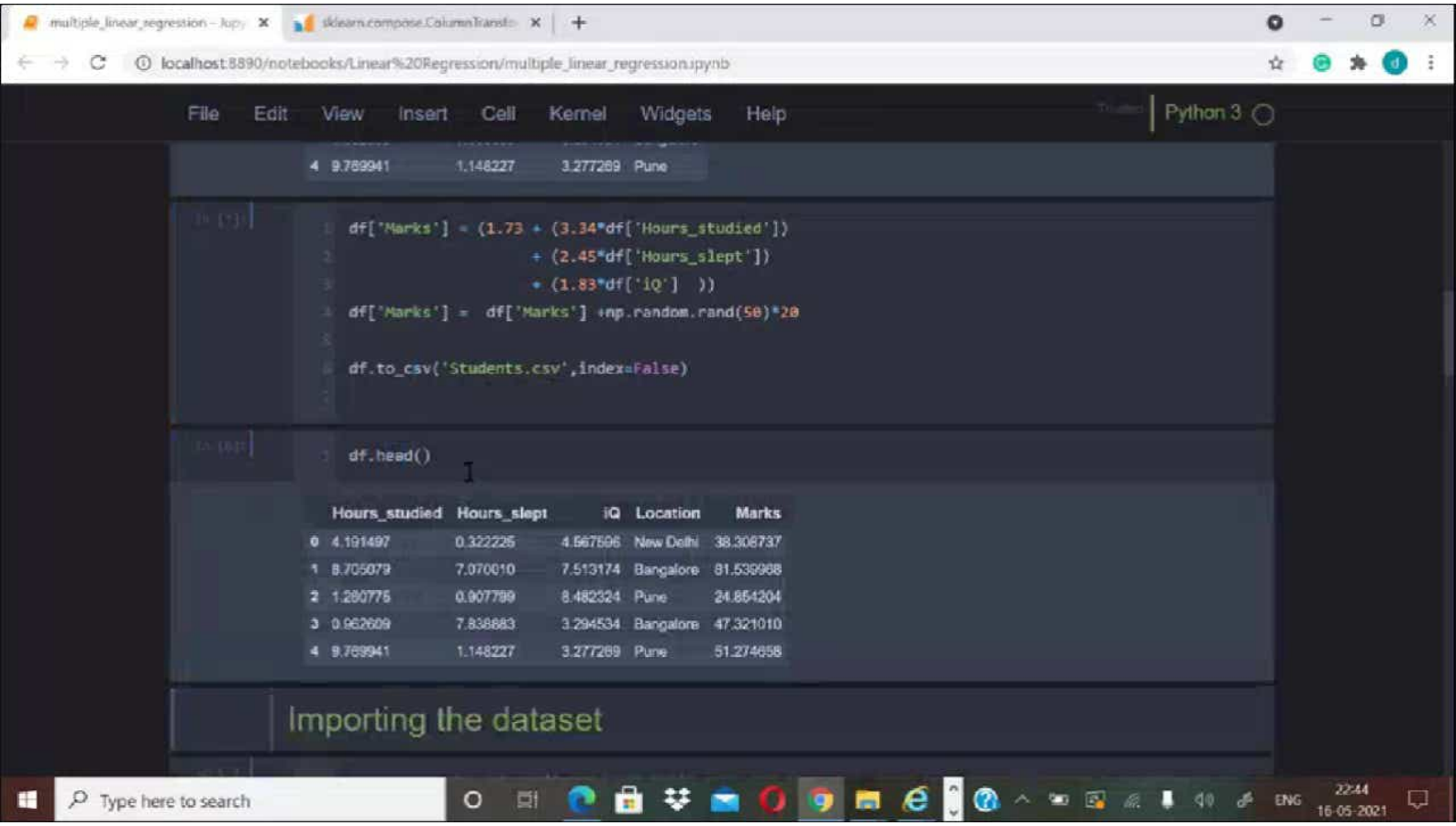
```
df['Marks'] = (1.73 + (3.34*df['Hours_studied'])
                    + (2.45*df['Hours_slept'])
                    + (1.83*df['iQ'] ))
df['Marks'] =  df['Marks'] +np.random.rand(50)*20

df.to_csv('Students.csv',index=False)
```

Type here to search

ENG 22:37 16-05-2021


multiple_linear_regression - Jup
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
for i in range(50):
location.append(choice(["New Delhi" , "Pune" , "Bangalore"]))
df['Location'] = location
df.head()
Hours_studied Hours_slept iQ Location
0 4.191497 0.322225 4.567598 New Delhi
1 8.705079 7.070010 7.613174 Bangalore
2 1.260775 0.907799 8.482324 Pune
3 0.962609 7.838883 3.294534 Bangalore
4 9.769941 1.148227 3.277269 Pune
df['Marks'] = (1.73 + (3.34*df['Hours_studied'])
+ (2.45*df['Hours_slept'])
+ (1.83*df['iQ']))
df['Marks'] = df['Marks'] +np.random.rand(50)*20
df.to_csv('Students.csv',index=False)
Type here to search
ENG
22:41
16-05-2021


multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
4 9.769941 1.148227 3.277269 Pune
df['Marks'] = (1.73 + (3.34*df['Hours_studied'])
+ (2.45*df['Hours_slept'])
+ (1.83*df['iQ']))
df['Marks'] = df['Marks'] +np.random.rand(50)*20
df.to_csv('Students.csv',index=False)
df.head()
Hours_studied Hours_slept iQ Location Marks
0 4.191497 0.322225 4.567596 New Delhi 38.308737
1 8.705079 7.070010 7.513174 Bangalore 81.539968
2 1.280775 0.907799 8.482324 Pune 24.854204
3 0.962609 7.838683 3.294534 Bangalore 47.321010
4 9.769941 1.148227 3.277269 Pune 51.274658
Importing the dataset
Type here to search
ENG
22:44
16-05-2021


multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
4 9.769941 1.148227 3.277269 Pune
df['Marks'] = (1.73 + (3.34*df['Hours_studied'])
+ (2.45*df['Hours_slept'])
+ (1.83*df['iQ']))
df['Marks'] = df['Marks'] +np.random.rand(50)*20
df.to_csv('Students.csv',index=False)
df.head()
Hours_studied Hours_slept iQ Location Marks
0 4.191497 0.322225 4.567596 New Delhi 38.308737
1 8.705079 7.070010 7.513174 Bangalore 81.539968
2 1.280775 0.907799 8.482324 Pune 24.854204
3 0.962609 7.838683 3.294534 Bangalore 47.321010
4 9.769941 1.148227 3.277269 Pune 51.274658
Importing the dataset
Type here to search
ENG
22:44
16-05-2021

multiple_linear_regression - Jup | sklearn.compose.ColumnTransfo

localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb

File Edit View Insert Cell Kernel Widgets Help Python 3

```
df.head()
```

	Hours_studied	Hours_slept	IQ	Location	Marks
0	4.191497	0.322225	4.567596	New Delhi	38.306737
1	8.705079	7.070010	7.513174	Bangalore	81.539968
2	1.260775	0.907799	8.482324	Pune	24.854204
3	0.962609	7.838883	3.294534	Bangalore	47.321010
4	9.789941	1.148227	3.277269	Pune	61.274658

Importing the dataset

```
dataset = pd.read_csv('Students.csv')

X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
print(X)
```

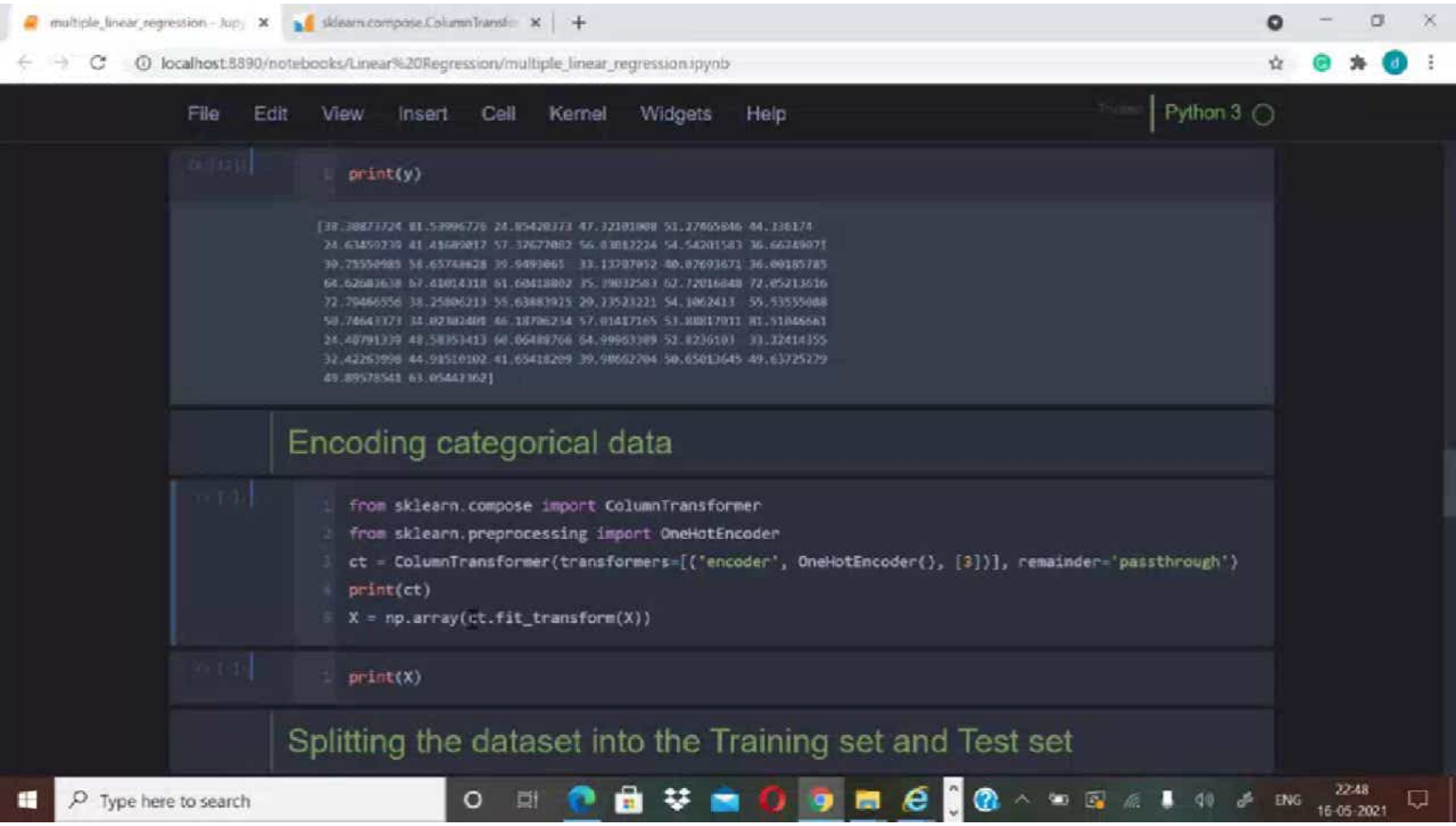
```
print(X)
```

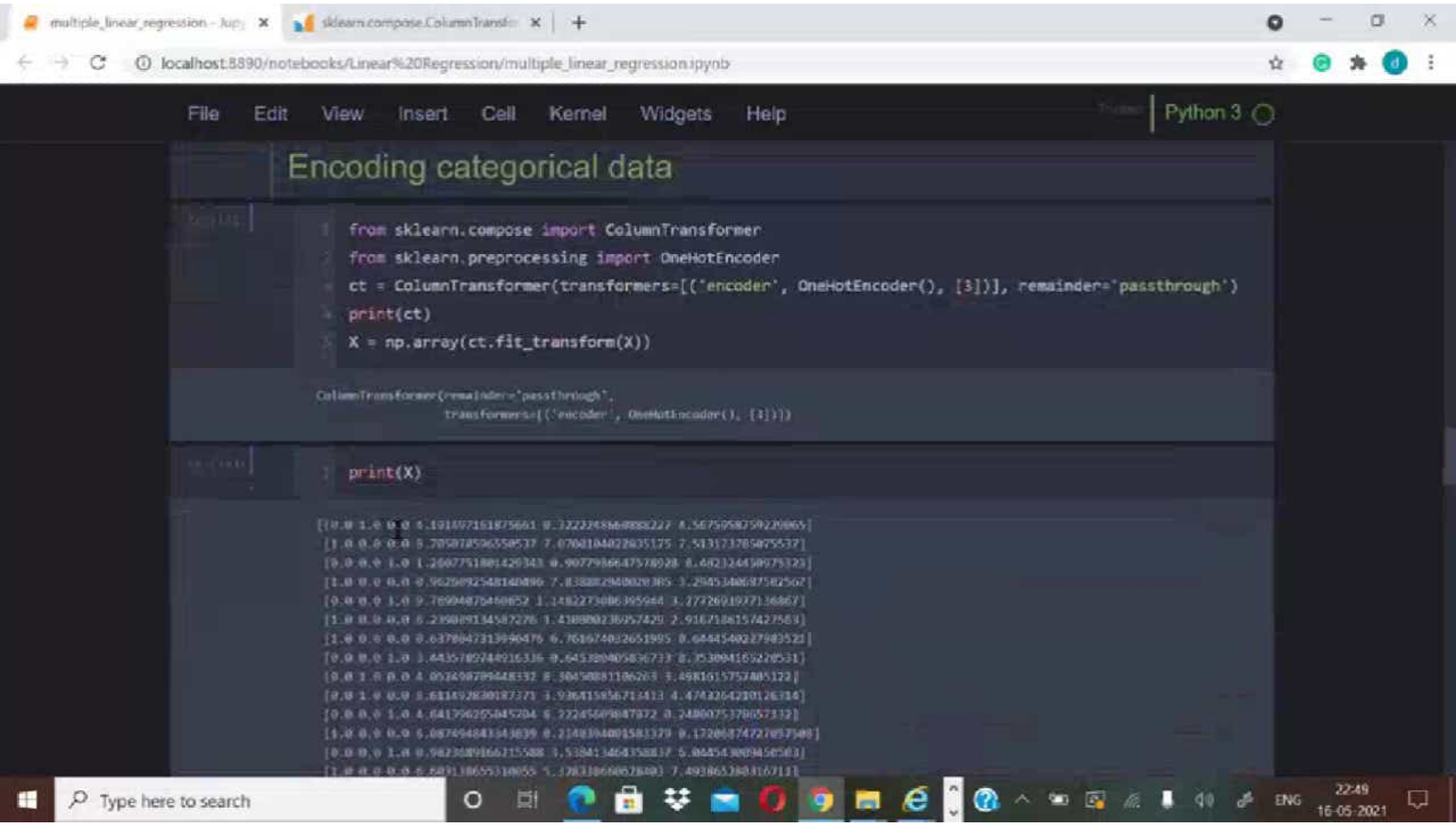
Encoding categorical data

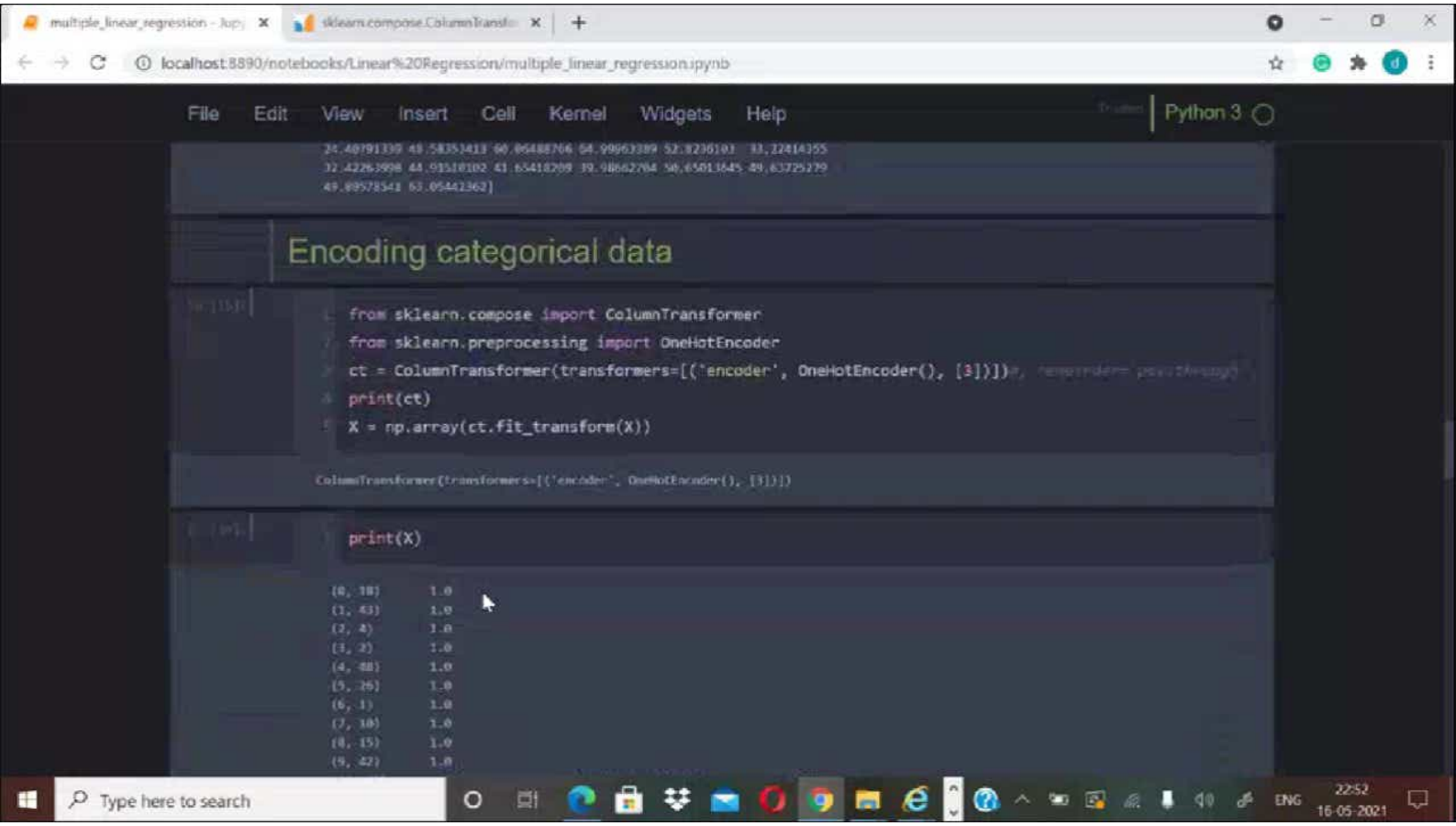
Type here to search

ENG 22:45 16-05-2021

	A	B	C	D
A	1	0	0	0
B	0	1	0	0
B	0	1	0	0
C	0	0	1	0
D	1	0	0	0
A	1	0	0	0


multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
print(y)
[38.38873724 81.53996776 24.85420373 47.32101008 51.27865846 44.336174
24.61850239 41.41689017 57.37677802 56.03812224 54.54201583 36.66249071
30.75550985 58.65748628 39.9493065 33.13787052 40.07693671 36.00185785
64.62683638 57.41014318 61.60418802 35.79832583 62.72016848 72.05213616
72.79466556 38.25806213 55.63883925 29.13523221 54.1062413 55.53555088
50.74643373 32.02382401 46.18706234 57.01417165 53.88817931 81.5184666
24.40791339 48.58353413 60.06488766 64.99663389 52.8236103 33.32414355
32.42263998 44.98510102 41.65418209 39.98662704 50.65013645 49.63725279
49.89578543 63.05443362]
Encoding categorical data
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [3])], remainder='passthrough')
print(ct)
X = np.array(ct.fit_transform(X))
print(X)
Splitting the dataset into the Training set and Test set
Type here to search
ENG
22:48
16-05-2021


multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
Encoding categorical data
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [3])], remainder='passthrough')
print(ct)
X = np.array(ct.fit_transform(X))
ColumnTransformer(remainder='passthrough',
transformers=[('encoder', OneHotEncoder(), [3])])
print(X)
Type here to search
ENG
22:49
16-05-2021


multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
24.40791339 49.58353413 60.06488766 64.99963389 52.8236103 83.22414355
32.42263998 44.91510102 41.65418209 39.98662704 50.65013645 49.63725279
49.89578543 63.05442362]
Encoding categorical data
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [3])])
print(ct)
X = np.array(ct.fit_transform(X))
ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [3])])
print(X)
(0, 10) 1.0
(1, 43) 1.0
(2, 4) 1.0
(3, 2) 1.0
(4, 88) 1.0
(5, 26) 1.0
(6, 1) 1.0
(7, 10) 1.0
(8, 15) 1.0
(9, 42) 1.0
Type here to search
ENG
22:52
16-05-2021

multiple_linear_regression - Jup | sklearn.compose.ColumnTransfo

localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb

File Edit View Insert Cell Kernel Widgets Help | Python 3

Encoding categorical data

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [3])], remainder='passthrough')
print(ct)
X = np.array(ct.fit_transform(X))
```

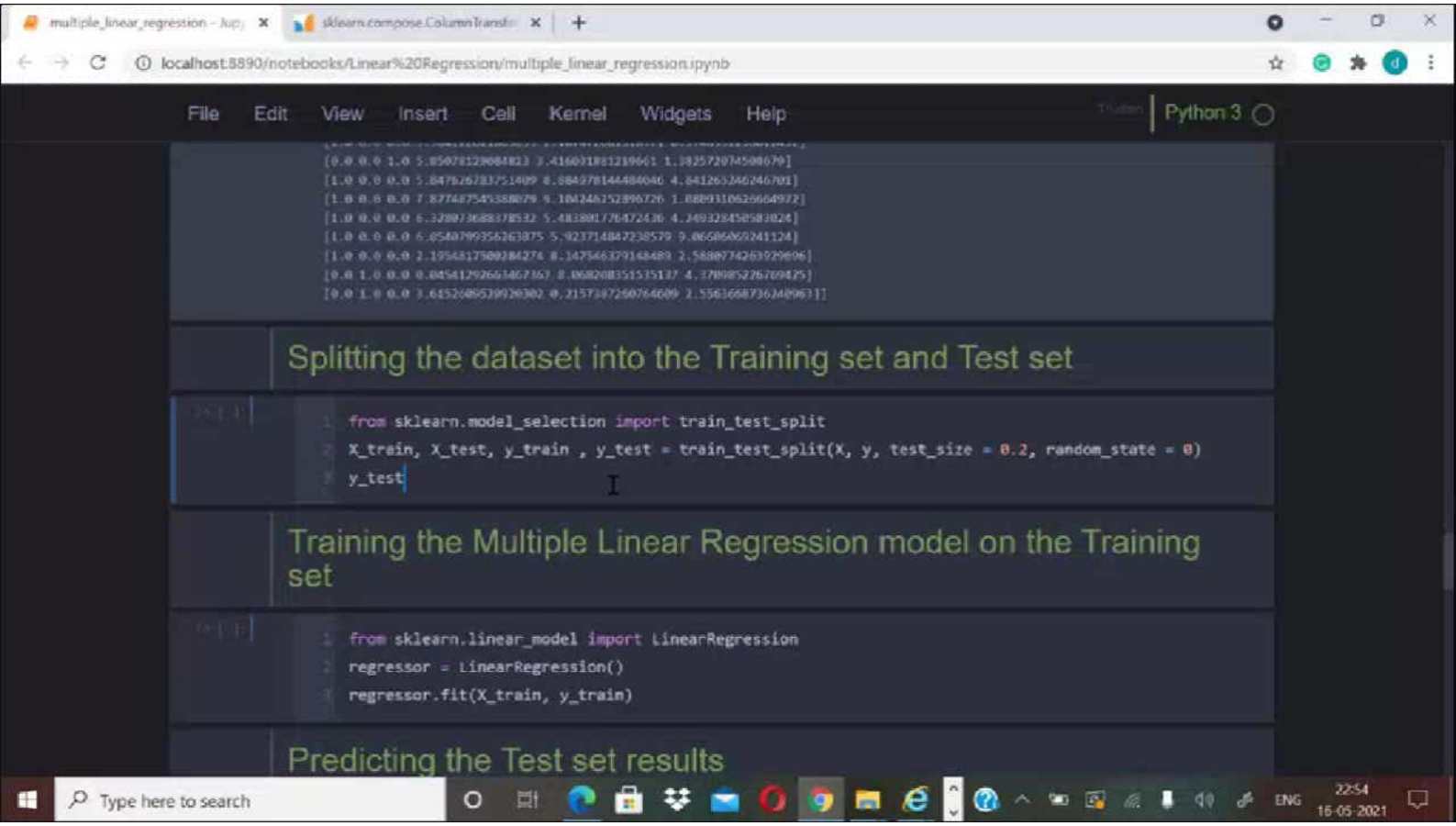
```
ColumnTransformer(remainder='passthrough',
                  transformers=[('encoder', OneHotEncoder(), [3])])
```

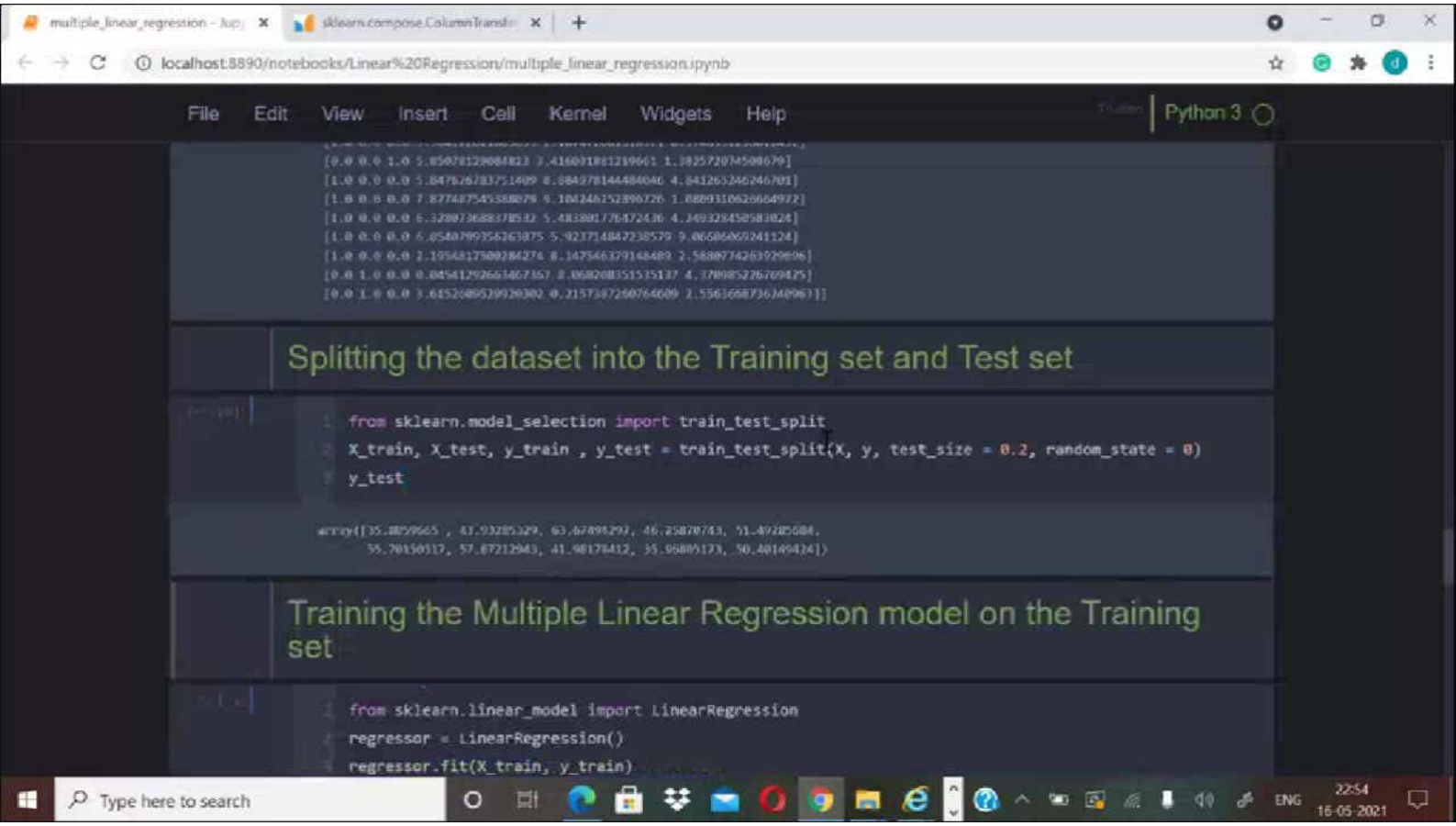
```
print(X)
```

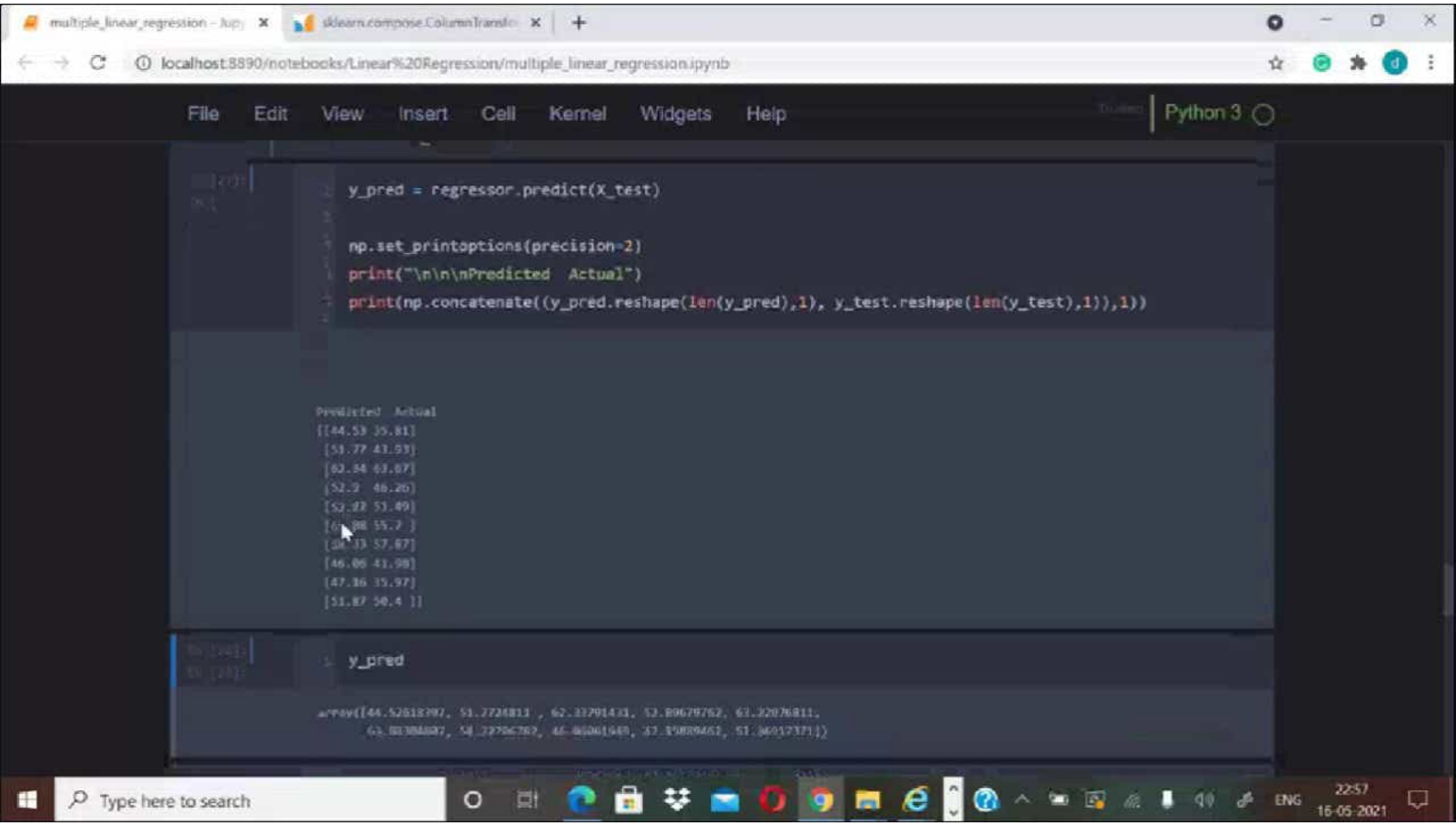
```
[[0.0 0.0 1.0 7.1130045187033477 4.4252887242181995 6.4383817028628291]
 [1.0 0.0 0.0 4.7357177479446581 1.7047172710337211 3.2262939981346122]
 [1.0 0.0 0.0 5.1013165278879 7.8096403430831673 7.1697884475489175]
 [0.0 1.0 0.0 0.9600474780432338 0.8188580513166826 4.1926237056457251]
 [0.0 0.0 1.0 6.9790945549141426 5.6569955227364101 1.0556174836773882]
 [0.0 1.0 0.0 3.4720456623547924 5.8142000519888352 6.3736773447687781]
 [0.0 1.0 0.0 9.6102601462407] 3.5886822708487368 8.3605203803710461]
 [0.0 1.0 0.0 1.5135492389912542 6.3483978148830956 0.5687304981050778]
 [0.0 0.0 1.0 3.1823534006310007 3.2945333489001603 5.5952024359893381]
 [1.0 0.0 0.0 2.0935261713078117 0.5517411766602586 1.0446551423263579]
 [0.0 1.0 0.0 7.5319206631741112 4.1822082961786721 7.0734053880653651]
 [0.0 0.0 1.0 5.4088325643283141 7.6147324769006751 1.0942666050578785]
 [0.0 1.0 0.0 2.2784926758159534 3.9276658590664984 3.2845369224418265]
```

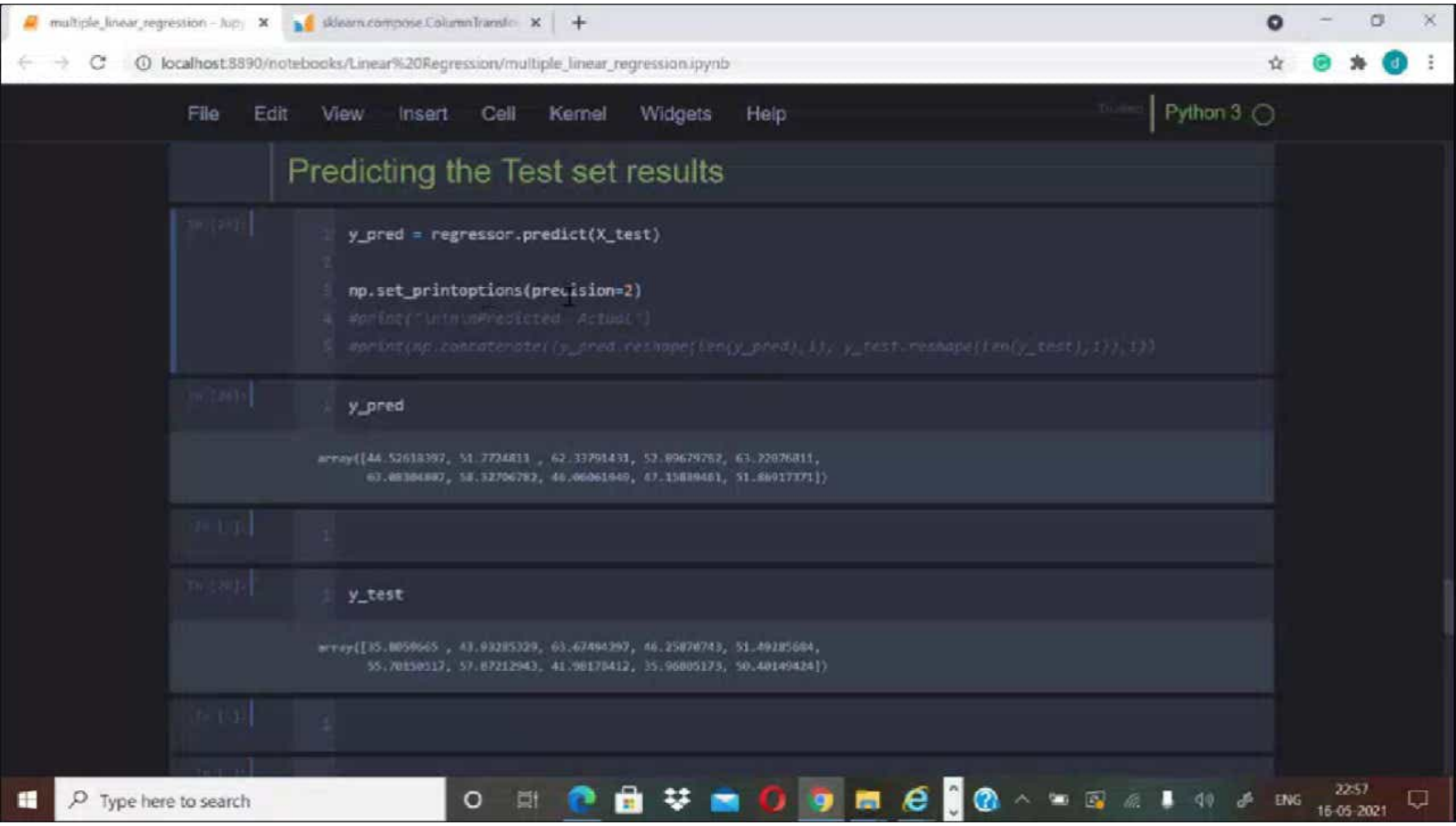
Type here to search

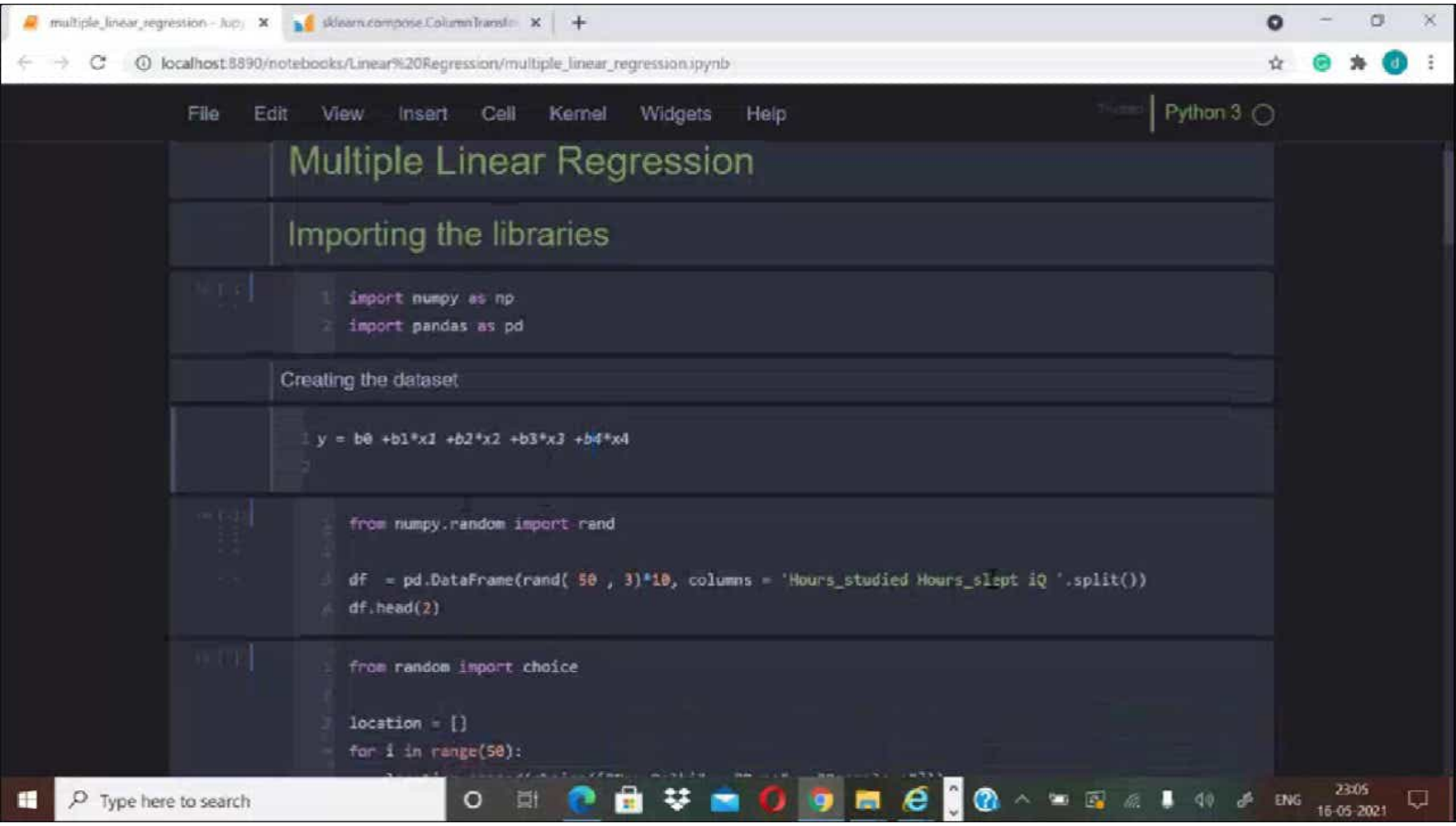
ENG 22:54 16-05-2021

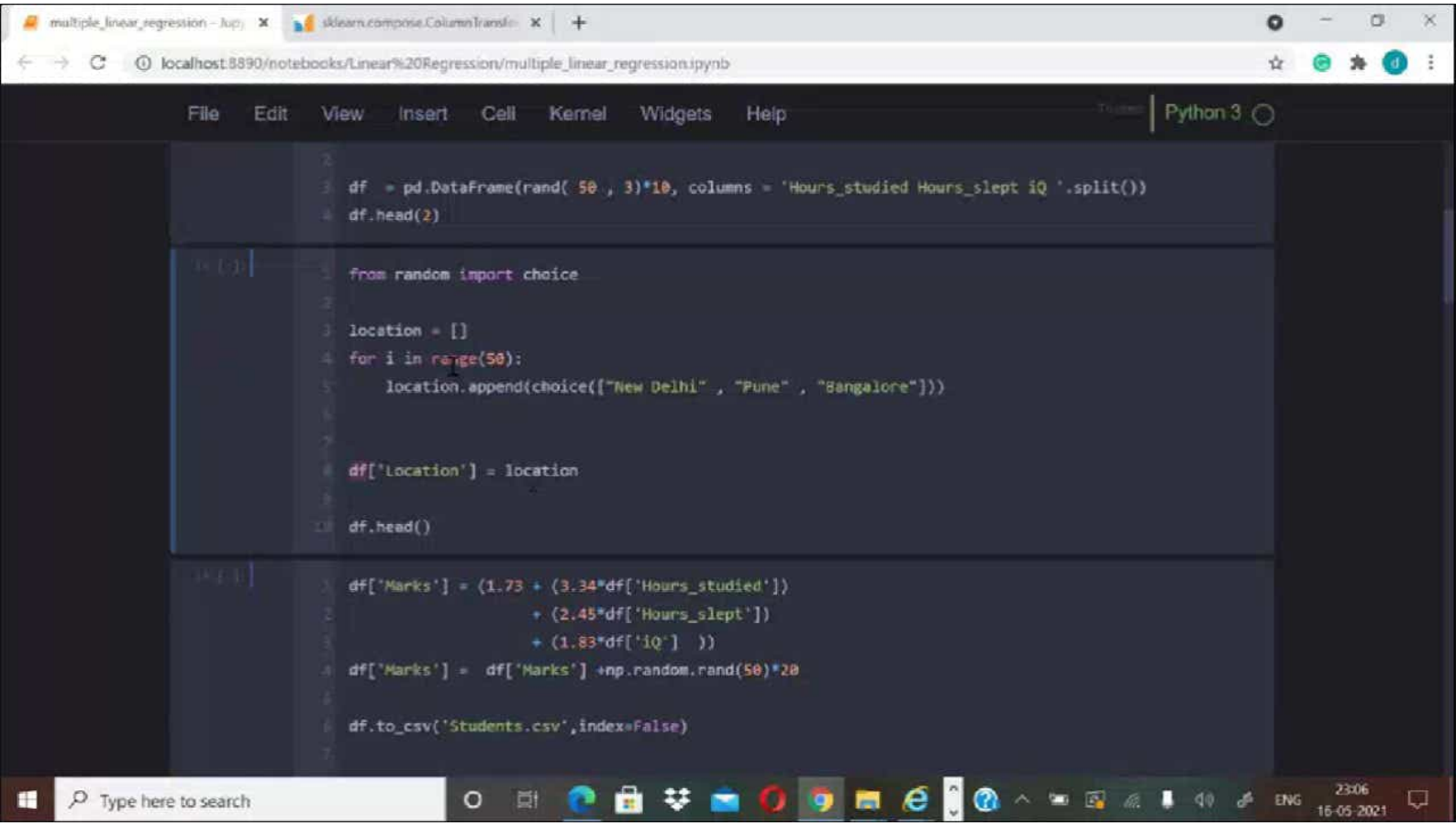

multiple_linear_regression - Jup
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
[0.0 0.0 1.0 5.850781290848823 3.4160318812189661 1.3825720745080679]
[1.0 0.0 0.0 5.8476267837514099 8.0843784448484046 4.8412652462467011]
[1.0 0.0 0.0 7.8774875453888078 9.1042462528896726 1.8889310626604972]
[1.0 0.0 0.0 6.3289736883778532 5.4838017764724236 4.2493284505838241]
[1.0 0.0 0.0 6.8540799356263875 5.9237148847238579 9.0658060924112411]
[1.0 0.0 0.0 2.1956817500284274 8.1475663791484893 2.5880774263929696]
[0.0 1.0 0.0 0.8454129266546736? 8.0682083515351374 4.3709852267694251]
[0.0 1.0 0.0 3.6452680539926302 0.2157107260764609 2.5563668736240963]]
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 = 0.2, random_state = 0)
y_test
Training the Multiple Linear Regression model on the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
Predicting the Test set results
Type here to search
ENG
22:54
16-05-2021

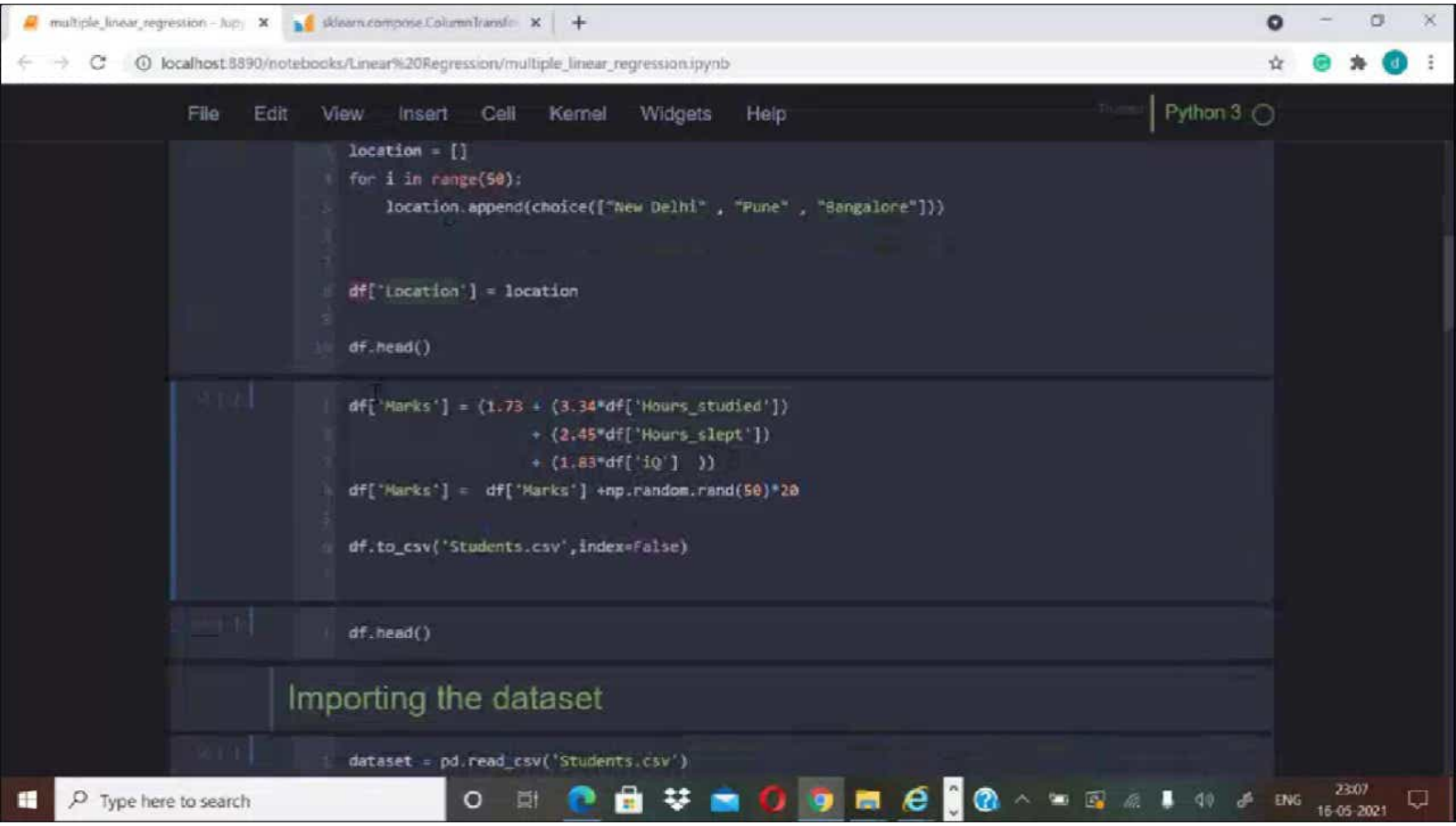

multiple_linear_regression - Jup
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
[0.0 0.0 1.0 5.850781290884823 3.4160318812189661 1.3825720745080679]
[1.0 0.0 0.0 5.8476267837514099 8.0843784448484046 4.8412652462467011]
[1.0 0.0 0.0 7.8774875453888078 9.1042462528896726 1.8889310626604972]
[1.0 0.0 0.0 6.3289736883778532 5.4838017764724436 4.2493284505838241]
[1.0 0.0 0.0 6.8540799356263875 5.9237148847238579 9.0658060924112411]
[1.0 0.0 0.0 2.1956817500284274 8.1475663791484893 2.5880774263929696]
[0.0 1.0 0.0 0.8454129266546736 8.0682083515351374 4.3709852267694251]
[0.0 1.0 0.0 3.6152680539926302 0.2157107260764609 2.5563668736240963]]
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 = 0.2, random_state = 0)
y_test
array([35.8059665 , 43.93285329, 63.67499297, 46.25870743, 51.49285504,
55.70150517, 57.87212943, 41.98178412, 35.96805173, 50.40149424])
Training the Multiple Linear Regression model on the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
Type here to search
ENG
22:54
16-05-2021

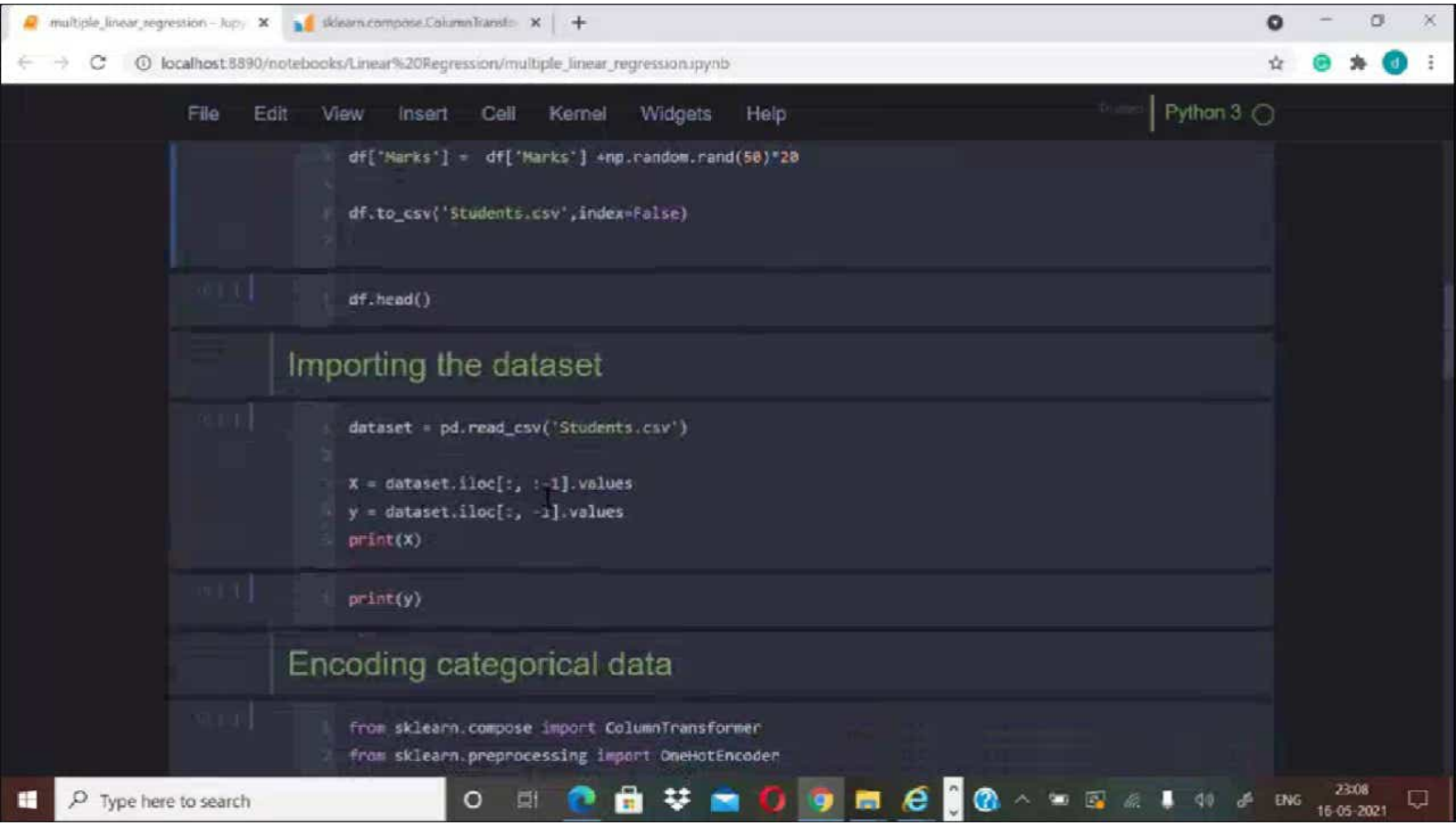

multiple_linear_regression - Jup
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
y_pred = regressor.predict(X_test)
np.set_printoptions(precision=2)
print("\n\n\nPredicted Actual")
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
Predicted Actual
[[44.53 35.81]
[51.77 41.53]
[62.34 63.87]
[52.2 46.26]
[53.88 53.49]
[88 55.7]
[3 57.87]
[46.06 41.98]
[47.86 35.97]
[51.87 50.4]]
y_pred
array([44.52818797, 51.2724811 , 62.33791431, 53.89679762, 63.22076811,
Type here to search
ENG
22:57
16-05-2021

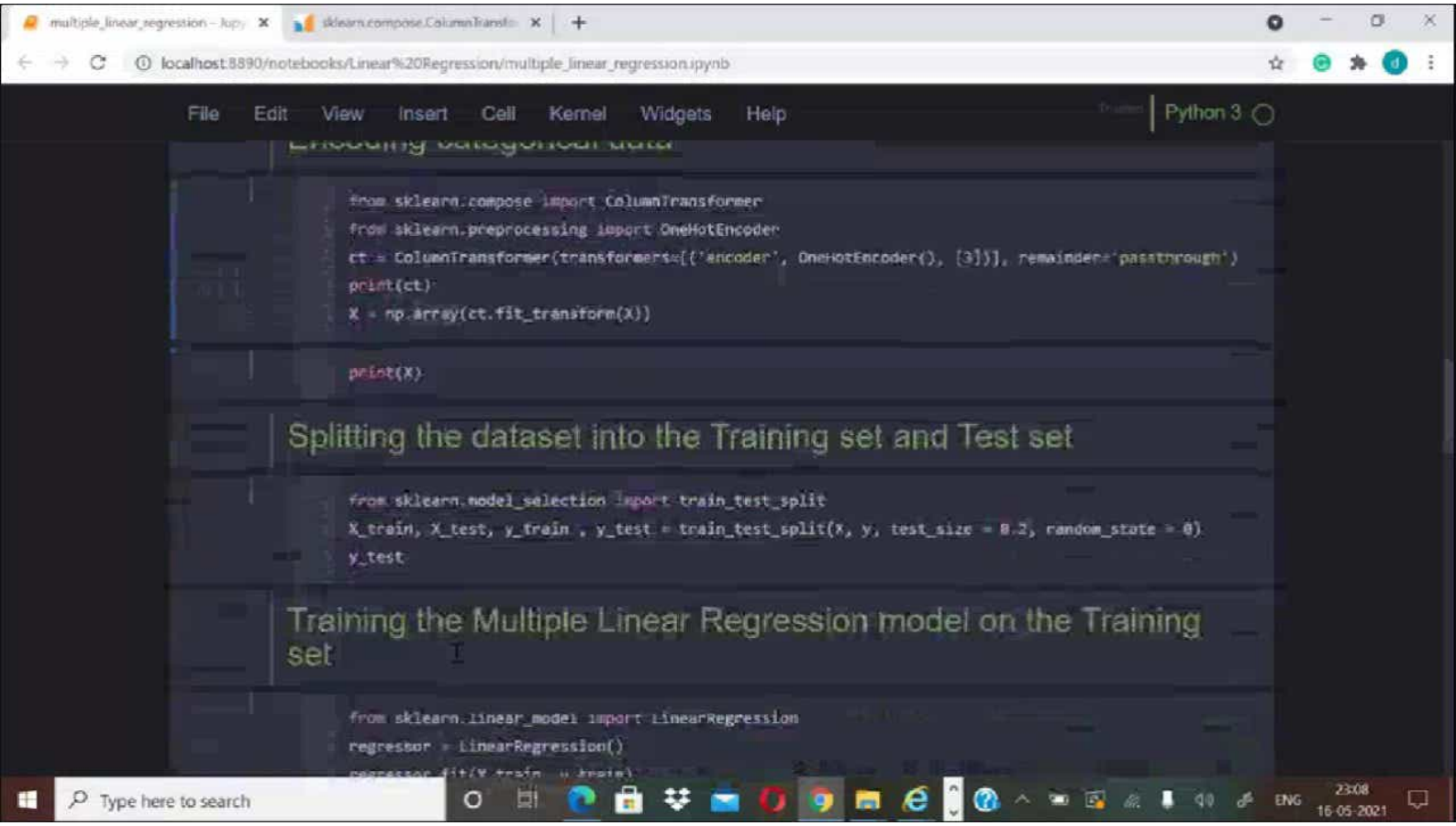

multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
Predicting the Test set results
y_pred = regressor.predict(X_test)
np.set_printoptions(precision=2)
#print("\n\nPredicted Actual")
#print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
y_pred
array([44.52618397, 51.7724811 , 62.33791431, 53.89679282, 63.22076811,
63.08304882, 58.32706782, 46.06061949, 47.15889481, 51.86917371])
y_test
array([35.8059065 , 43.93285329, 63.67496397, 46.25870743, 51.49285604,
55.70150517, 57.87212943, 41.98178412, 35.96805173, 50.40149424])
Type here to search
ENG
22:57
16-05-2021

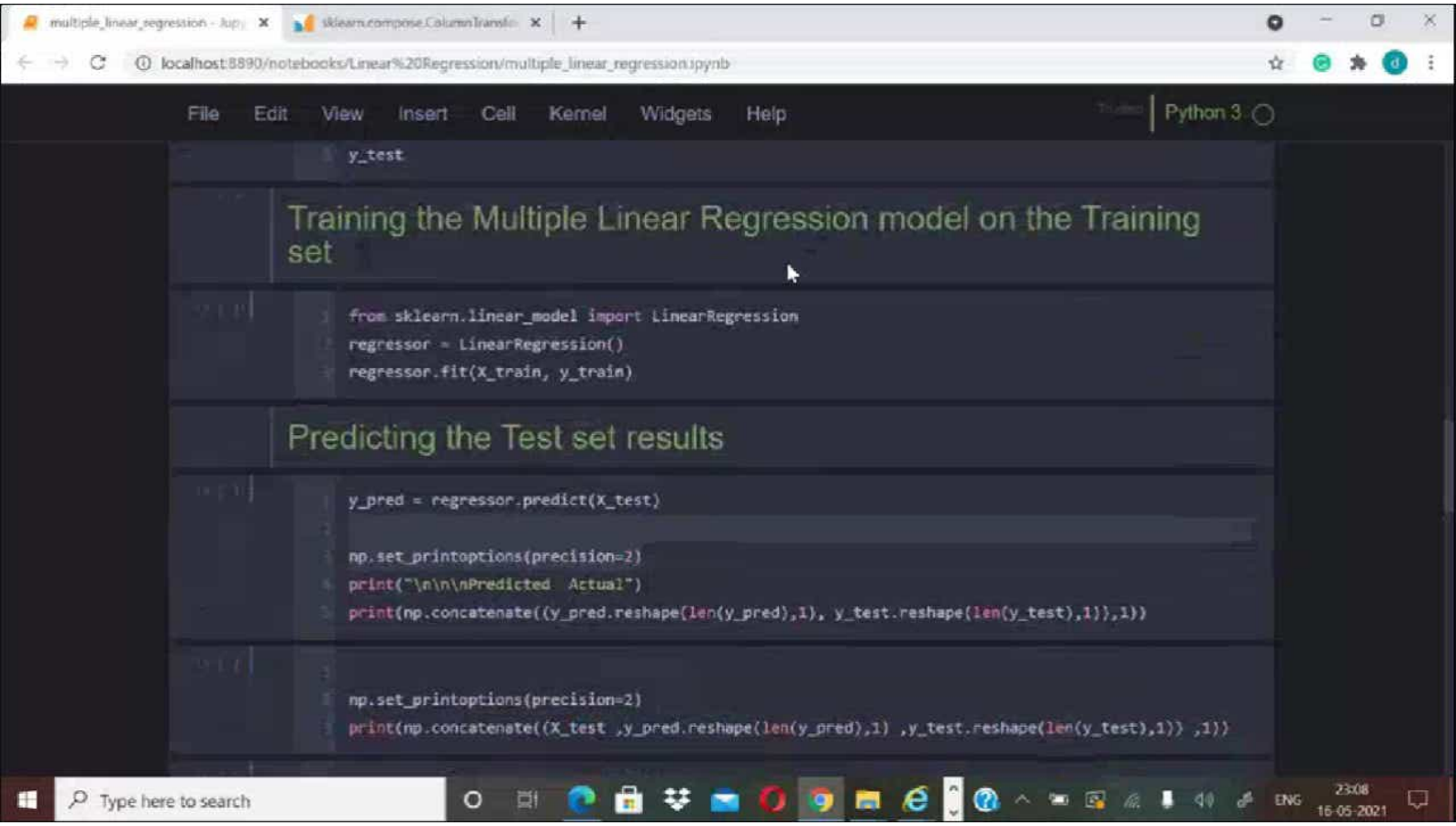

multiple_linear_regression - Jup
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
Multiple Linear Regression
Importing the libraries
import numpy as np
import pandas as pd
Creating the dataset
y = b0 +b1*x1 +b2*x2 +b3*x3 +b4*x4
from numpy.random import rand
df = pd.DataFrame(rand(50 , 3)*10, columns = 'Hours_studied Hours_slept iQ '.split())
df.head(2)
from random import choice
location = []
for i in range(50):
Type here to search
ENG
23:05
16-05-2021

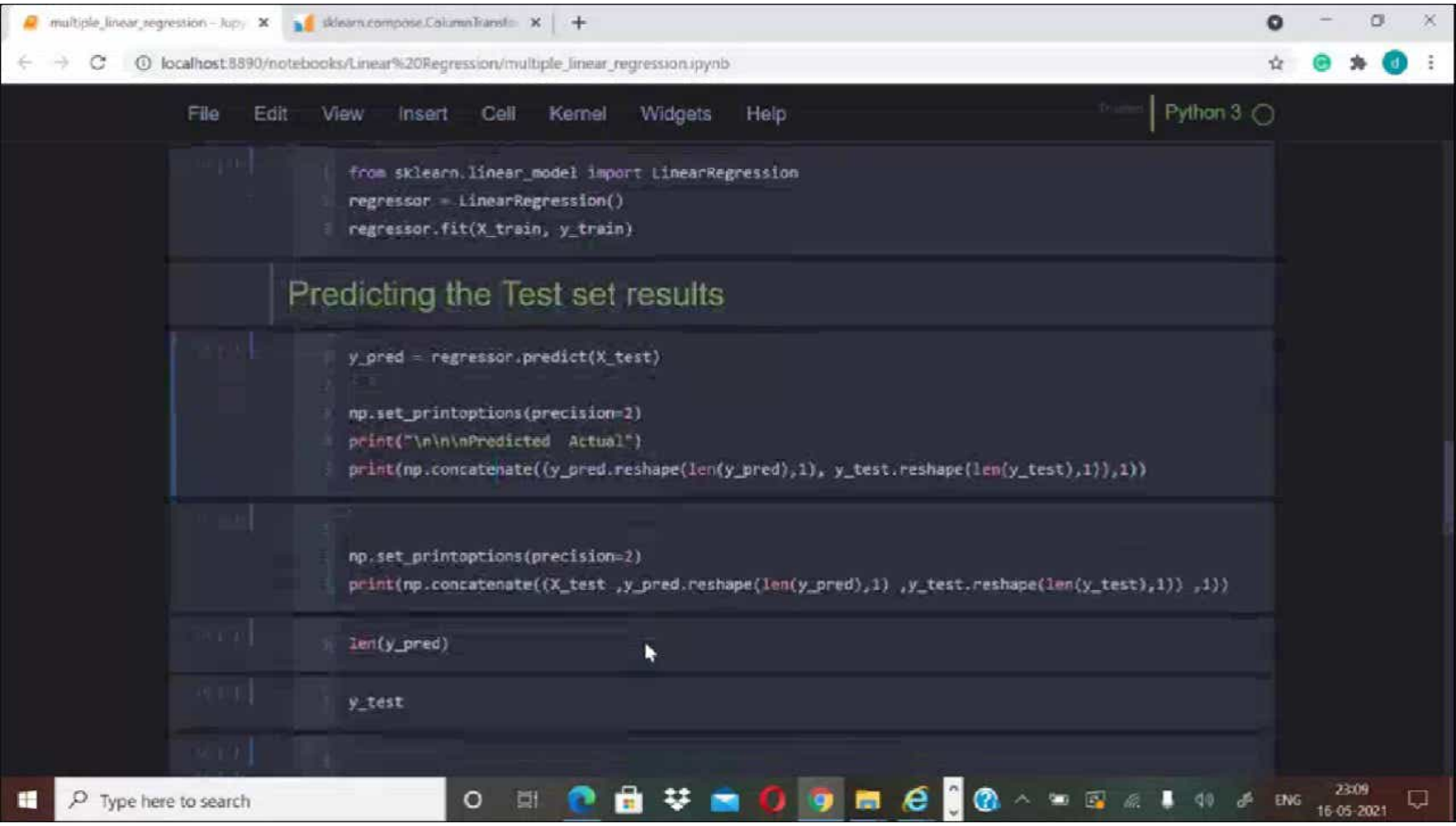

multiple_linear_regression - Jup
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
df = pd.DataFrame(rand(50 , 3)*10, columns = 'Hours_studied Hours_slept iQ '.split())
df.head(2)
from random import choice
location = []
for i in range(50):
location.append(choice(["New Delhi" , "Pune" , "Bangalore"]))
df['Location'] = location
df.head()
df['Marks'] = (1.73 + (3.34*df['Hours_studied'])
+ (2.45*df['Hours_slept'])
+ (1.83*df['iQ']))
df['Marks'] = df['Marks'] +np.random.rand(50)*20
df.to_csv('Students.csv',index=False)
Type here to search
ENG
23:06
16-05-2021


multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
location = []
for i in range(50):
location.append(choice(["New Delhi" , "Pune" , "Bangalore"]))
df['location'] = location
df.head()
df['Marks'] = (1.73 + (3.34*df['Hours_studied'])
+ (2.45*df['Hours_slept'])
+ (1.83*df['iQ']))
df['Marks'] = df['Marks'] +np.random.rand(50)*20
df.to_csv('Students.csv',index=False)
df.head()
Importing the dataset
dataset = pd.read_csv('Students.csv')
Type here to search
ENG
23:07
16-05-2021


multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
df['Marks'] = df['Marks'] +np.random.rand(50)*20
df.to_csv('Students.csv',index=False)
df.head()
Importing the dataset
dataset = pd.read_csv('Students.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
print(X)
print(y)
Encoding categorical data
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
Type here to search
ENG
23:08
16-05-2021


multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [3])], remainder='passthrough')
print(ct)
X = np.array(ct.fit_transform(X))
print(X)
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 = 0.2, random_state = 0)
y_test
Training the Multiple Linear Regression model on the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
Type here to search
ENG
23:08
16-05-2021


multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
y_test
Training the Multiple Linear Regression model on the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
Predicting the Test set results
y_pred = regressor.predict(X_test)
np.set_printoptions(precision=2)
print("\n\n\nPredicted Actual")
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
np.set_printoptions(precision=2)
print(np.concatenate((X_test ,y_pred.reshape(len(y_pred),1) ,y_test.reshape(len(y_test),1)) ,1))
Type here to search
ENG
23:08
16-05-2021


multiple_linear_regression - Jupy
sklearn.compose.ColumnTransfo
localhost:8890/notebooks/Linear%20Regression/multiple_linear_regression.ipynb
File Edit View Insert Cell Kernel Widgets Help
Python 3
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
Predicting the Test set results
y_pred = regressor.predict(X_test)
np.set_printoptions(precision=2)
print("\n\n\nPredicted Actual")
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
np.set_printoptions(precision=2)
print(np.concatenate((X_test ,y_pred.reshape(len(y_pred),1) ,y_test.reshape(len(y_test),1)) ,1))
len(y_pred)
y_test
Type here to search
ENG
23:09
16-05-2021