CENTRAL TEST

March 25, 2024

[18]:
```
#import necessary libraries
#loading the dataset and identifying the arrays
#Create the moddel and fit it into the data
#Get results
#Make a regrassion line and predictions
#Plot the graph
```

[19]:
```
#import  necessary libraries
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
```

[20]:
```
#loading the dataset and idetify the arrays
data=pd.read_csv("C:\\Users\\DELL\\Desktop\\dataset ken.csv")
data
```

[20]:

	Work hours	Employee Output
0	9.892033	166.519278
1	4.561662	101.232060
2	5.109478	111.102744
3	7.926598	136.488276
4	7.546302	137.120214
..	…	…
195	4.207135	105.712627
196	9.243529	153.553019
197	1.680201	72.376029
198	6.630046	124.066461
199	19.766516	256.540815

[200 rows x 2 columns]

[21]:
```
#arrays
x=np.array(data["Work hours"]).reshape(-1,1)
y=np.array(data["Employee Output"])
x
```

[21]:
```
array([[ 9.89203291],
       [ 4.56166209],
```

```
[ 5.10947848],
[ 7.92659819],
[ 7.54630195],
[19.9314846 ],
[ 8.16394401],
[15.4378799 ],
[15.21073376],
[ 6.20018697],
[ 6.93082409],
[ 7.03529635],
[ 2.90933713],
[19.45329369],
[18.18356876],
[11.19914207],
[ 6.27181493],
[17.76400075],
[13.49146132],
[ 7.82174895],
[10.14368248],
[10.48207005],
[18.56001856],
[11.42746138],
[13.36675138],
[ 1.04517382],
[ 6.54114608],
[ 1.12803278],
[ 3.59655388],
[18.51866345],
[18.76030446],
[14.28185415],
[14.65375218],
[ 9.23495352],
[18.62658535],
[ 8.12840487],
[13.66411541],
[12.99831736],
[11.97530357],
[ 4.44078784],
[13.64714338],
[17.56112605],
[15.93434527],
[ 8.64004496],
[18.35756438],
[15.63667355],
[14.51500557],
[ 2.49709387],
[18.32616899],
```

```
[ 7.75421988],
[ 5.89859091],
[12.33462829],
[ 9.35692458],
[ 5.10677829],
[16.77991777],
[ 3.57238405],
[ 4.54228341],
[13.19752892],
[ 9.5823249 ],
[ 1.46894685],
[ 2.77920143],
[ 2.24614367],
[ 9.55569931],
[10.80592459],
[19.16142098],
[11.67584623],
[10.53328182],
[18.44525378],
[18.38514033],
[ 5.0401773 ],
[13.65265221],
[19.28552237],
[ 4.53923294],
[14.32034397],
[15.95520215],
[18.73502399],
[17.07444992],
[ 8.43091613],
[ 0.10879738],
[ 0.69730658],
[ 0.27810731],
[11.77819861],
[ 7.65986193],
[ 2.29637899],
[17.28908018],
[16.43314063],
[14.7498337 ],
[16.8068834 ],
[ 8.03058207],
[14.97240003],
[11.1925889 ],
[12.26475148],
[ 5.96203303],
[12.04758347],
[ 8.51353679],
[10.7708877 ],
```

```
[ 9.73459714],
[ 9.99783276],
[18.34918965],
[ 5.25754044],
[ 0.99395372],
[ 9.25384059],
[ 8.16438045],
[ 9.74086961],
[ 1.41031557],
[11.64217222],
[19.74043597],
[ 4.18176912],
[ 4.24752236],
[19.22270099],
[14.1390815 ],
[11.85936426],
[ 7.61704814],
[12.77480993],
[10.04857138],
[ 1.79586149],
[ 5.94349334],
[16.56288468],
[ 3.50911931],
[18.08978914],
[11.76629658],
[ 0.33068914],
[ 7.40044354],
[ 0.84259006],
[ 2.54561102],
[ 3.58536452],
[ 3.3291717 ],
[17.67376993],
[18.11904095],
[ 4.70801957],
[ 8.14519849],
[ 9.42320315],
[16.60468035],
[ 5.84015244],
[ 3.47657064],
[ 9.16286217],
[18.04858326],
[11.89873849],
[ 8.25226667],
[ 8.11301874],
[ 6.16968332],
[ 6.35757462],
[ 8.30172032],
```

```
[12.76334139],
[15.90539035],
[ 0.74066826],
[14.69082195],
[15.45141726],
[ 4.88849054],
[19.70153719],
[13.50011689],
[17.85529472],
[ 4.86109401],
[ 1.03391755],
[14.19486127],
[19.14528625],
[ 0.45670435],
[18.26285047],
[14.56547425],
[19.10810639],
[ 5.46270897],
[ 3.41187024],
[ 6.45202036],
[ 5.54476922],
[ 9.81204906],
[ 2.61431493],
[14.76367473],
[ 2.80647147],
[11.41301416],
[10.66675866],
[ 4.67046071],
[ 9.5378809 ],
[ 9.29484341],
[16.12062194],
[ 0.37170223],
[14.25633169],
[ 5.01986724],
[13.21053122],
[12.72891917],
[ 1.1397533 ],
[14.37819098],
[ 6.79846865],
[10.99285737],
[14.88639858],
[ 3.57025477],
[ 5.01408206],
[ 2.67759408],
[ 3.0154816 ],
[18.78060682],
[19.42905773],
```

```
       [ 2.85113177],
       [ 3.91730024],
       [ 2.00795677],
       [ 1.6362494 ],
       [15.36751344],
       [ 4.20713489],
       [ 9.24352894],
       [ 1.68020072],
       [ 6.63004567],
       [19.76651602]])
```

[22]:
```
y
```

[22]:
```
array([166.5192781 , 101.2320599 , 111.1027435 , 136.4882763 ,
       137.1202144 , 256.5767963 , 132.2524823 , 210.9509735 ,
       207.9388666 , 126.4276177 , 128.179451  , 130.3189111 ,
        89.14284342, 254.1711912 , 235.5169247 , 164.5552322 ,
       115.5973476 , 232.6098355 , 186.9318656 , 138.9946529 ,
       174.3342656 , 162.6729169 , 231.5953233 , 174.3707117 ,
       187.2117879 ,  74.17626476, 125.6521388 ,  64.33963402,
        99.59340618, 244.1487523 , 246.8842117 , 200.1901354 ,
       220.0982437 , 151.9600848 , 246.6763396 , 141.6056866 ,
       191.9890249 , 189.6602932 , 179.326341  , 113.2663167 ,
       191.6435798 , 242.4630319 , 216.5070602 , 141.0546574 ,
       246.4007529 , 216.6530597 , 206.3579298 ,  87.40180192,
       241.4445229 , 139.9872581 , 122.1621737 , 187.3534361 ,
       151.5771432 , 109.3430542 , 220.6383443 ,  89.37038092,
       107.6176229 , 192.7514691 , 153.4020407 ,  73.17367028,
        87.49855863,  89.9463262 , 159.7470827 , 169.2378501 ,
       247.2343396 , 175.5419291 , 173.2221703 , 253.1546129 ,
       249.2989474 , 116.5126266 , 198.8832288 , 245.5719629 ,
       108.2138901 , 197.6712496 , 213.8604059 , 250.1766845 ,
       234.3177912 , 150.7104259 ,  62.96439972,  71.98203345,
        59.08434841, 169.3651984 , 138.7409486 ,  80.65280014,
       222.7825473 , 218.1268035 , 214.7952149 , 229.8299009 ,
       139.513619  , 211.7685156 , 175.7317806 , 177.7305407 ,
       116.5796356 , 173.3238268 , 143.8939235 , 160.1577634 ,
       159.264922  , 165.5999611 , 242.5082153 , 108.174525  ,
        68.73303961, 154.5765271 , 139.6385419 , 154.6310517 ,
        72.72339688, 179.8303183 , 254.9100074 ,  97.60625347,
        93.44753333, 261.4688288 , 205.8560828 , 189.6040414 ,
       140.6556252 , 190.6378087 , 158.4578817 ,  81.13197838,
       117.1914112 , 229.0765857 ,  97.09758122, 239.7772106 ,
       189.2953538 ,  59.43128441, 138.3377017 ,  66.2745848 ,
        79.92711504,  93.0902541 ,  90.36551668, 239.9403871 ,
       245.4207045 , 110.4601697 , 137.4141555 , 151.8069429 ,
       216.2857314 , 123.0015592 ,  98.80552883, 149.6690991 ,
```

```
       240.7701697 , 169.2570257 , 135.7616461 , 143.3359188 ,
       116.2795834 , 120.4711087 , 144.49536   , 187.0082382 ,
       221.4651224 ,  71.63436288, 212.6477029 , 216.0098732 ,
       114.3752588 , 257.280909  , 190.8487472 , 241.4232102 ,
       112.7116286 ,  76.73637714, 191.9111369 , 247.8271546 ,
        58.71833251, 246.7027141 , 199.8556287 , 243.0758258 ,
       115.3732345 , 103.5431104 , 129.67723   , 106.5080329 ,
       158.6974192 ,  89.09715485, 206.1566015 ,  89.35163502,
       183.2481929 , 167.8511193 , 102.1421367 , 158.3382612 ,
       154.2417222 , 220.4891204 ,  57.65658447, 198.4891549 ,
       117.3774853 , 189.7726621 , 188.4992289 ,  66.15301229,
       201.8544232 , 130.860514  , 169.4341371 , 205.2648882 ,
        92.18160392, 101.2480008 ,  84.99446117,  89.03047989,
       244.0834194 , 248.3156636 ,  96.52648397, 101.1336475 ,
        76.72949328,  75.85879091, 212.867657  , 105.7126269 ,
       153.5530185 ,  72.37602864, 124.0664607 , 256.5408145 ])
```

[27]:
```
#spliting the dataset
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.8)
```

[29]:
```
#Create the moddel and fit it into train
model=LinearRegression()
model.fit(x_train,y_train)
```

[29]:
```
LinearRegression()
```

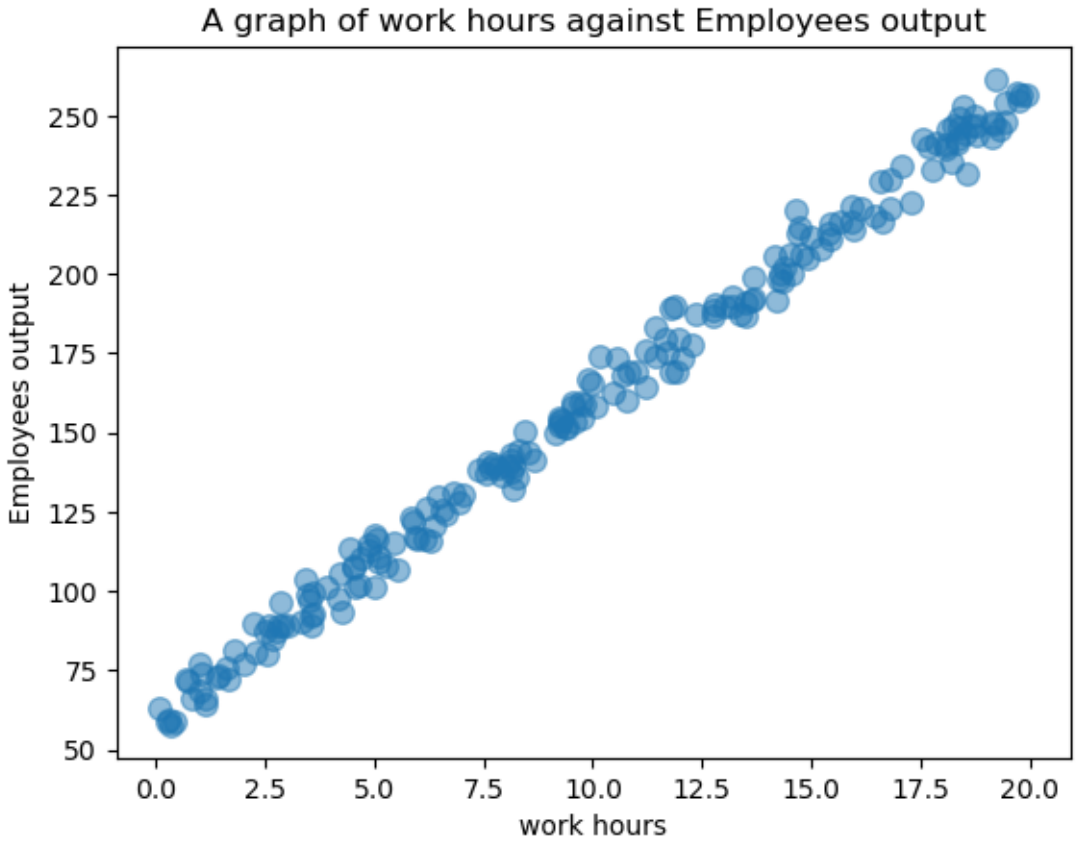
[30]:
```
#Make predictions
y_pred=model.predict(x_test)
y_pred
```

[30]:
```
array([244.92707431, 187.18156994,  76.3353868 ,  94.66835151,
       104.27371805, 135.91507871, 160.13749643,  67.41376103,
       110.24473304, 157.07222417, 225.03196892, 112.41006738,
        95.62282829, 247.12453674,  94.0238173 , 118.79610234,
       176.01267484, 166.29577215, 142.24301677, 201.44212561,
        88.43783679,  86.07870794, 251.52860309,  95.75213753,
       191.63639837, 157.0097389 , 236.09841736,  74.66874745,
        95.60161681, 256.29868723, 115.27139126,  90.07506154,
       121.80055062, 158.57808703, 178.56854405, 109.98477682,
       169.5443116 , 126.08272931, 202.05448339, 240.24272943])
```

[32]:
```
#Get results
accuracy=model.score(x_test,y_test)
accuracy
```

[32]:
```
0.9899154292682244
```

[26]:
```
#visualize
import matplotlib.pyplot as plt
plt.scatter(x,y,label="datapoints",alpha=0.5, s=70)
plt.title("A graph of work hours against Employees output")
plt.xlabel("work hours")
plt.ylabel("Employees output")
plt.show()
```



[34]:
```
#evaluuating the model
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
mea=mean_absolute_error(y_test,y_pred)
mse=mean_squared_error(y_test,y_pred)
r2=r2_score(y_test,y_pred)

print("r2:",r2)
print("mean_absolute_error:",mea)
print("mean_ssquareed_error",mse)
```

```
r2: 0.9899154292682244
mean_absolute_error: 4.5613727289084345
```

```
mean_ssquareed_error 31.49517107067053
```

Model Optimisation

[39]:
```
from sklearn.model_selection import train_test_split,GridSearchCV
x_train,x_test,y_train,y_test=train_test_split(x,y, test_size=0.2,␣
 ↪random_state=42)
x_train.shape
```

[39]:
```
(160, 1)
```

[40]:
```
#Create the model and fit in the trained dataset
model=LinearRegression()
model
```

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

[41]:
```
#Defining parameter grid search
param_grid={
     "fit_intercept":[True,False],
    "copy_X":[True,False],
    "n_jobs":[None,True,False],
    "positive":[False,True],
}
param_grid
```

[41]:
```
{'fit_intercept': [True, False],
 'copy_X': [True, False],
 'n_jobs': [None, True, False],
 'positive': [False, True]}
```

[46]:
```
#performing the grid search
grid_search=GridSearchCV(model,param_grid,cv=5)
grid_search.fit(x_train,y_train)
grid_search
```

[46]:
```
GridSearchCV(cv=5, estimator=LinearRegression(),
             param_grid={'copy_X': [True, False],
                         'fit_intercept': [True, False],
                         'n_jobs': [None, True, False],
                         'positive': [False, True]})
```

[49]:
```
#defining the best_parameters
best_params =grid_search.best_params_
best_params
```

[49]:
```
{'copy_X': True, 'fit_intercept': True, 'n_jobs': None, 'positive': False}
```

[51]:
```
#train the  linear regrassion model using best params
best_model = LinearRegression(**best_params)
best_model.fit(x_train, y_train)
```

[51]:
```
LinearRegression()
```

[53]:
```
#Predictions
y_pred=best_model.predict(x_test)
y_pred
```

[53]:
```
array([167.51628428, 171.76512795, 246.78175171, 205.1635476 ,
       240.41957867,  78.4722816 , 110.66013913, 106.99206692,
        64.34275833, 215.79124619, 165.15892307, 169.71851383,
        86.5924105 ,  61.73439824,  87.22022246, 191.72073997,
       105.7203804 , 108.88339864, 136.6508382 , 243.05963268,
        85.91077782, 122.87959592, 109.15520806, 180.18277672,
       176.49465692,  88.22830029, 232.18528193, 243.65603676,
        96.22649295, 225.39514017, 122.16895288, 194.50793413,
        96.09770969, 246.53093637, 194.5938087 ,  74.64716157,
       151.56256711, 178.70603977, 158.00335046, 230.05586894])
```

[55]:
```
#determining the model performance
mea=mean_absolute_error(y_test,y_pred)
mse=mean_squared_error(y_test,y_pred)
r2=r2_score(y_test,y_pred)

print("r2_score           :",r2)
print("mean_absolute_error:",mea)
print("mean_squared_error :",mse)
```

```
r2_score           : 0.9920588065936665
mean_absolute_error: 4.4637644722033905
mean_squared_error : 27.80375611405956
```

[ ]:
```

```