```
import os
import numpy as np
import pandas as pd


from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import seaborn as sns
from io import StringIO


from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
import random
from sklearn.preprocessing import StandardScaler


data = pd.read_excel("/Task_Data_File.xlsx",skiprows=[0,1],index_col=0)


data.head()
```

S.No.	College	Role	City type	Previous CTC	Previous job changes	Graduation marks	Exp (Months)	CTC
1	Tier 1	Manager	Non-Metro	55523	3	66	19	71406.576531
2	Tier 2	Executive	Metro	57081	1	84	18	68005.870631
3	Tier 2	Executive	Metro	60347	2	52	28	76764.020277
4	Tier 3	Executive	Metro	49010	2	81	33	82092.386880

```
data.shape
```

```
(1338, 8)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1338 entries, 1 to 1338
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   College               1338 non-null   object 
 1   Role                  1338 non-null   object 
 2   City type             1338 non-null   object 
 3   Previous CTC          1338 non-null   int64  
 4   Previous job changes  1338 non-null   int64  
 5   Graduation marks      1338 non-null   int64  
 6   Exp (Months)          1338 non-null   int64  
 7   CTC                   1338 non-null   float64
dtypes: float64(1), int64(4), object(3)
memory usage: 94.1+ KB
```

```
data
```

S.No.	College	Role	City type	Previous CTC	Previous job changes	Graduation marks	Exp (Months)	CTC
1	Tier 1	Manager	Non-	55523	3	66	19	71406.576531

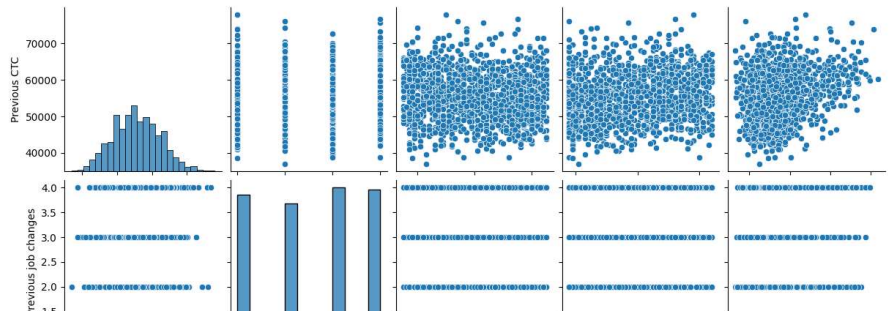
```
data.isna().sum()
```

```
College                0
Role                   0
City type              0
Previous CTC           0
Previous job changes   0
Graduation marks       0
Exp (Months)           0
CTC                    0
dtype: int64
```

```
cat_clmns = data.select_dtypes(['object']).columns
```

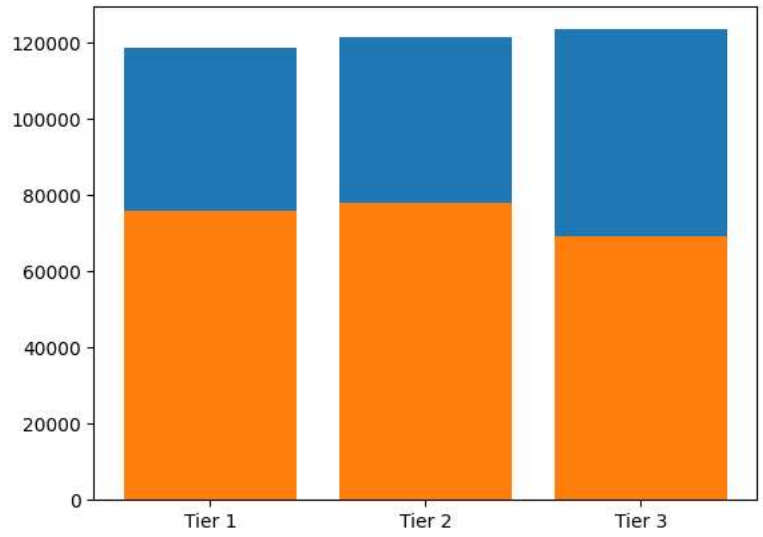
```
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7fdcf14ce050>
```



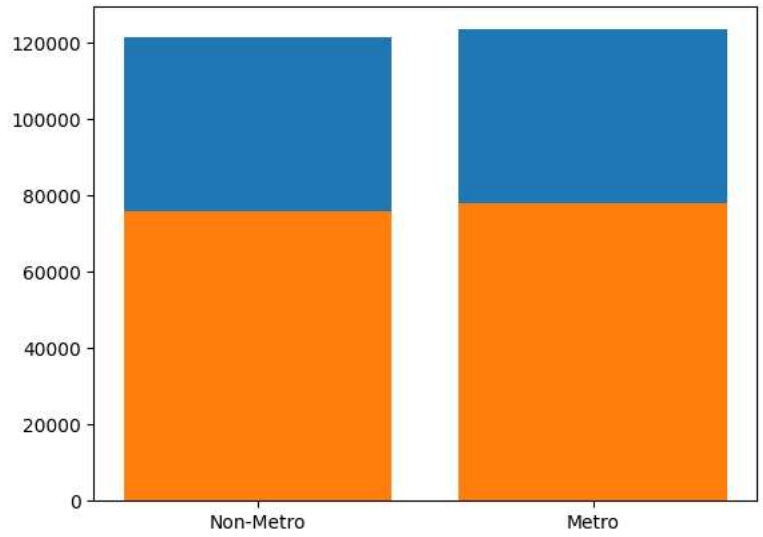

```
plt.bar('College' , 'CTC' , data = data), plt.bar('College','Previous CTC', data = data)
```

```
(<BarContainer object of 1338 artists>, <BarContainer object of 1338 artists>)
```



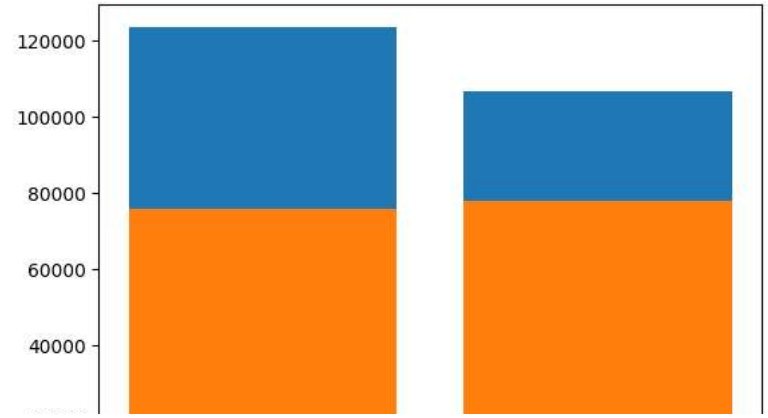

```
plt.bar('City type','CTC', data = data), plt.bar('City type','Previous CTC', data = data)
```

```
(<BarContainer object of 1338 artists>, <BarContainer object of 1338 artists>)
```



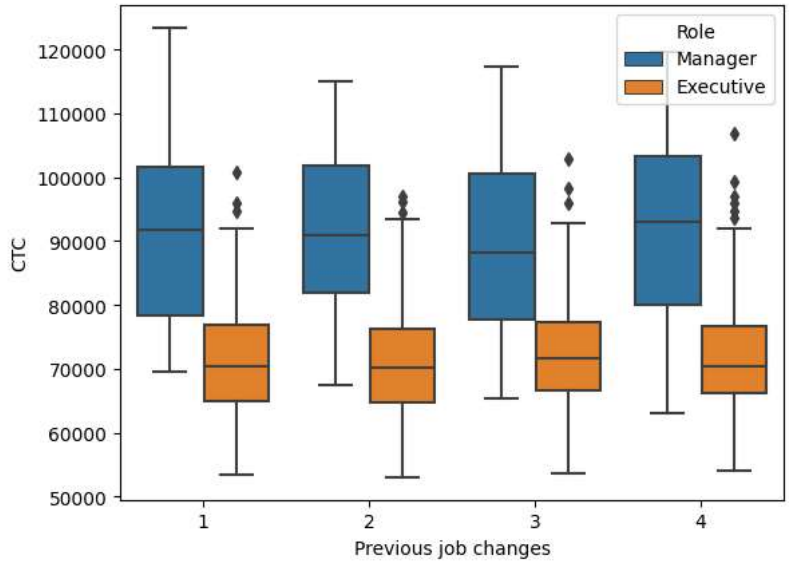

```
plt.bar('Role','CTC',data = data), plt.bar('Role','Previous CTC', data = data)
```

```
(<BarContainer object of 1338 artists>, <BarContainer object of 1338 artists>)
```




```
sns.boxplot(x='Previous job changes', y='CTC',hue='Role', data=data)
```

```
<Axes: xlabel='Previous job changes', ylabel='CTC'>
```




```
cm = data.corr()
```

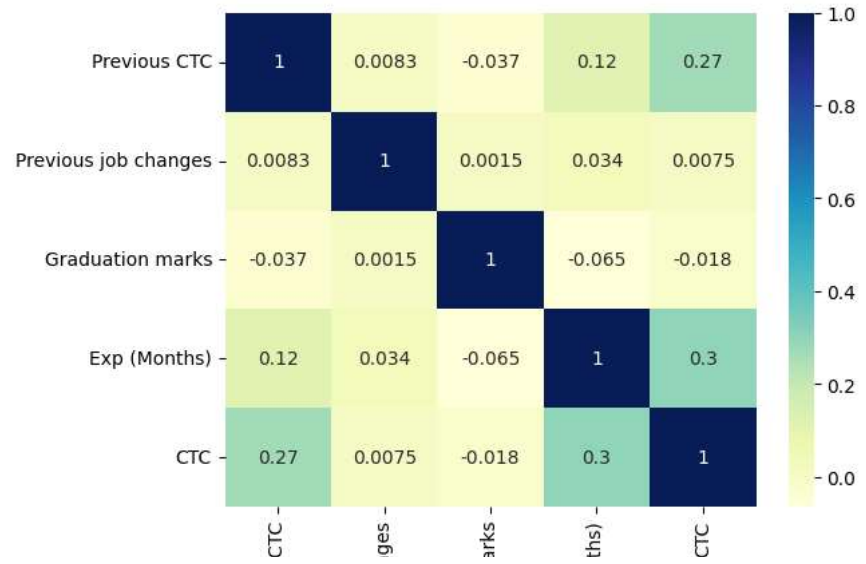
```
<ipython-input-18-251cf5733e53>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future versior
  cm = data.corr()
```

```
cm
```

	Previous CTC	Previous job changes	Graduation marks	Exp (Months)	CTC
Previous CTC	1.000000	0.008282	-0.037170	0.117035	0.270260
Previous job changes	0.008282	1.000000	0.001507	0.034137	0.007518
Graduation marks	-0.037170	0.001507	1.000000	-0.065412	-0.017557
Exp (Months)	0.117035	0.034137	-0.065412	1.000000	0.301569
CTC	0.270260	0.007518	-0.017557	0.301569	1.000000

```
sns.heatmap(cm, annot=True, cmap='YlGnBu' )
```

```
<Axes: >
```




```
x = data.drop(["CTC"],axis=1)

y = data['CTC']
```

```
x
```

S.No.	College	Role	City type	Previous CTC	Previous job changes	Graduation marks	Exp (Months)
1	Tier 1	Manager	Non-Metro	55523	3	66	19
2	Tier 2	Executive	Metro	57081	1	84	18
3	Tier 2	Executive	Metro	60347	2	52	28
4	Tier 3	Executive	Metro	49010	2	81	33
5	Tier 3	Executive	Metro	57879	4	74	32
...	...	...	...	...	...	...	...
1334	Tier 3	Executive	Metro	59661	4	68	50
1335	Tier 1	Executive	Non-Metro	53714	1	67	18
1336	Tier 2	Executive	Non-Metro	61957	1	47	18
1337	Tier 1	Executive	Non-Metro	53203	3	69	21
1338	Tier 3	Manager	Non-Metro	51820	1	47	61

1338 rows × 7 columns

```
tx = pd.get_dummies(x,drop_first=True)
tx.head()
```

S.No.	Previous CTC	Previous job changes	Graduation marks	Exp (Months)	College_Tier 2	College_Tier 3	Role_Manager	City type_Non-Metro
1	55523	3	66	19	0	0	1	1
2	57081	1	84	18	1	0	0	0
3	60347	2	52	28	1	0	0	0
4	49010	2	81	33	0	1	0	0

```
sc = StandardScaler()
tx = sc.fit_transform(tx)
```

```
pd.DataFrame(tx)
```

	0	1	2	3	4	5	6	7
0	-0.008793	0.422577	0.410307	-1.438764	-0.611324	-0.566418	1.970587	1.010519
1	0.224333	-1.358237	1.619243	-1.509965	1.635795	-0.566418	-0.507463	-0.989591
2	0.713028	-0.467830	-0.529976	-0.797954	1.635795	-0.566418	-0.507463	-0.989591
3	-0.983340	-0.467830	1.417754	-0.441948	-0.611324	1.765481	-0.507463	-0.989591
4	0.343738	1.312985	0.947612	-0.513149	-0.611324	1.765481	-0.507463	-0.989591
...	...	...	...	...	...	...	...	...
1333	0.610381	1.312985	0.544633	0.768473	-0.611324	1.765481	-0.507463	-0.989591
1334	-0.279475	-1.358237	0.477470	-1.509965	-0.611324	-0.566418	-0.507463	1.010519
1335	0.953934	-1.358237	-0.865792	-1.509965	1.635795	-0.566418	-0.507463	1.010519
1336	-0.355937	0.422577	0.611797	-1.296362	-0.611324	-0.566418	-0.507463	1.010519
1337	-0.562877	-1.358237	-0.865792	1.551686	-0.611324	1.765481	1.970587	1.010519

1338 rows × 8 columns

```
random.seed(42)
for i in range(0,101):
    print(i)
    for j in range(1,100):
        x_train, x_test, y_train, y_test = train_test_split(tx,y, test_size=j/100, random_state=i)
        x_train.shape, x_test.shape
        logreg = LinearRegression(n_jobs=i)
        logreg.fit(x_train, y_train)
        y_pred_test = logreg.predict(x_test)
        y_pred_test
        if (r2_score(y_test,y_pred_test)>0.8 and j>1):
          print('Model accuracy score:(0:0.4f) @random_state:(1) @testsplit(%): {2}'.format(r2_score(y_test, y_pred_test),i,j))
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
```

```
43
44
45
46
47
48
49
50
Model accuracy score:(0:0.4f) @random_state:(1) @testsplit(%): 2
51
52
53
54
55
56
```

```
X_train, X_test, Y_train, Y_test = train_test_split(tx,y,test_size = .02, random_state = 50)
```

```
import random
random.seed(42)
from sklearn.linear_model import LinearRegression
lr = LinearRegression(n_jobs=2)
lr.fit(X_train, Y_train)
```

```
▾ LinearRegression
LinearRegression(n_jobs=2)
```

```
lr.intercept_
```

```
75468.6486526581
```

```
lr.coef_
```

```
array([ 2984.49295358,   -70.10966725,   -18.19289083,  3654.22577171,
       -2408.25804437, -2003.38289156,  7767.08017745, -2038.58581722])
```

```
Test_Salary = pd.read_excel("/content/Salary_test_data.xlsx")
```

```
Test_Salary.columns
```

```
Index(['College', 'Role', 'City type', 'College_T1', 'College_T2',
       'Role_Manager', 'City_Metro', 'previous CTC', 'previous job changes',
       'Graduation marks', 'Exp', 'Actual CTC', 'Predicted CTC', 'Unnamed: 13',
       'Unnamed: 14', 'Unnamed: 15', 'Unnamed: 16', 'Unnamed: 17',
       'Unnamed: 18', 'Unnamed: 19', 'Unnamed: 20', 'Unnamed: 21',
       'Unnamed: 22', 'Unnamed: 23'],
      dtype='object')
```

```
Test_Salary.drop(['Unnamed: 13','College','Role','City type', 'Unnamed: 14', 'Unnamed: 15', 'Unnamed: 16', 'Unnamed: 17',
       'Unnamed: 18', 'Unnamed: 19', 'Unnamed: 20', 'Unnamed: 21',
       'Unnamed: 22', 'Unnamed: 23'], inplace=True, axis=1)
```

```
Test_Salary
```

College_T1	College_T2	Role_Manager	City_Metro	previous CTC	previous job changes	Graduation marks	Exp	Actual CT

```
x_test = Test_Salary.drop(['Actual CTC','Predicted CTC'], axis=1)
y_test = Test_Salary['Actual CTC']
x_test
```

	College_T1	College_T2	Role_Manager	City_Metro	previous CTC	previous job changes	Graduation marks	Exp
0	1	0	1	0	55523	3	66	19
1	0	1	0	1	57081	1	84	18
2	0	1	0	1	60347	2	52	28
3	0	0	0	1	49010	2	81	33
4	0	0	0	1	57879	4	74	32
...	...	...	...	...	...	...	...	...
1333	0	0	0	1	59661	4	68	50
1334	1	0	0	0	53714	1	67	18
1335	0	1	0	0	61957	1	47	18
1336	1	0	0	0	53203	3	69	21
1337	0	0	1	0	51820	1	47	61

1338 rows × 8 columns

```
len(Test_Salary)
```

```
1338
```

```
x_test = sc.fit_transform(x_test)
pd.DataFrame(x_test)
```

	0	1	2	3	4	5	6	7
0	1.030356	-0.611324	1.970587	-1.010519	-0.008793	0.422577	0.410307	-1.438764
1	-0.970538	1.635795	-0.507463	0.989591	0.224333	-1.358237	1.619243	-1.509965
2	-0.970538	1.635795	-0.507463	0.989591	0.713028	-0.467830	-0.529976	-0.797954
3	-0.970538	-0.611324	-0.507463	0.989591	-0.983340	-0.467830	1.417754	-0.441948
4	-0.970538	-0.611324	-0.507463	0.989591	0.343738	1.312985	0.947612	-0.513149
...	...	...	...	...	...	...	...	...
1333	-0.970538	-0.611324	-0.507463	0.989591	0.610381	1.312985	0.544633	0.768473
1334	1.030356	-0.611324	-0.507463	-1.010519	-0.279475	-1.358237	0.477470	-1.509965
1335	-0.970538	1.635795	-0.507463	-1.010519	0.953934	-1.358237	-0.865792	-1.509965
1336	1.030356	-0.611324	-0.507463	-1.010519	-0.355937	0.422577	0.611797	-1.296362
1337	-0.970538	-0.611324	1.970587	-1.010519	-0.562877	-1.358237	-0.865792	1.551686

1338 rows × 8 columns

```
y_prediction = lr.predict(x_test)
y_prediction
```

```
array([80152.60890171, 93918.62166493, 72813.23619748, ...,
       65551.23857267, 82308.38862547, 63075.1332238 ])
```

```
print(f'r2_score of this model:{r2_score(y_test,y_prediction)}')
print(f'MAE of this model:{mean_absolute_error(y_test,y_prediction)}')
print(f'MSE of this model:{mean_squared_error(y_test,y_prediction)}')
```

```
r2_score of this model:-0.6852084456416327
MAE of this model:12846.282249614145
MSE of this model:265280284.88209185
```

```
Test_Salary['Predicted CTC'] = y_prediction
Test_Salary
```

	College_T1	College_T2	Role_Manager	City_Metro	previous CTC	previous job changes	Graduation marks	Exp	Actual CT
0	1	0	1	0	55523	3	66	19	71406.57653
1	0	1	0	1	57081	1	84	18	68005.87063
2	0	1	0	1	60347	2	52	28	76764.02027
3	0	0	0	1	49010	2	81	33	82092.38688
4	0	0	0	1	57879	4	74	32	73878.09772
...	...	...	...	...	...	...	...	...	
1333	0	0	0	1	59661	4	68	50	69712.40365
1334	1	0	0	0	53714	1	67	18	69298.75009
1335	0	1	0	0	61957	1	47	18	66397.77068
1336	1	0	0	0	53203	3	69	21	64044.38294
1337	0	0	1	0	51820	1	47	61	83346.06096

```
Test_Salary.to_excel('/content/Predicted_Salary.xlsx')
```

My Conclusions from this task:

- MSE of the considered model is 265280284.8820916
- MAE of the considered nodel is 12846.28224961414
- Avg Salary of a candidate is 73112
- Managers or people with one job change get higher salary
- Metro city candidates get higher salary than non metro city candidates
- Linear regression on this model gives an r2_score of 40.00%

Thank You All

✓ 0s completed at 4:17 PM