

```
In [1]: import numpy as np
import pandas as pd
```

```
In [2]: dataset=pd.read_csv(r"D:\ML_Course\Works_on_python\Multi_Linear_Regression\50_Startups.csv")
```

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

Out[3]:

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94

```
In [4]: dataset["State"].unique()
```

Out[4]: array(['New York', 'California', 'Florida'], dtype=object)

```
In [5]: dataset.isnull().any()
```

Out[5]: R&D Spend False
Administration False
Marketing Spend False
State False
Profit False
dtype: bool

```
In [6]: from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
dataset["State"]=le.fit_transform(dataset["State"])
```

```
In [7]: dataset.head()
```

```
Out[7]:
```

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	2	192261.83
1	162597.70	151377.59	443898.53	0	191792.06
2	153441.51	101145.55	407934.54	1	191050.39
3	144372.41	118671.85	383199.62	2	182901.99
4	142107.34	91391.77	366168.42	1	166187.94

```
In [8]: x=dataset.iloc[:,0:4].values  
y=dataset.iloc[:,4:5].values
```

```
In [9]: x.shape
```

```
Out[9]: (50, 4)
```

```
In [10]: y.shape
```

```
Out[10]: (50, 1)
```

```
In [11]: from sklearn.preprocessing import OneHotEncoder  
one=OneHotEncoder()  
z=one.fit_transform(x[:,3:4]).toarray()  
x=np.delete(x,3,axis=1)  
x=np.concatenate((z,x),axis=1)
```

```
In [12]: x.shape
```

```
Out[12]: (50, 6)
```

```
In [13]: 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)
```

```
In [14]: x_test.shape
```

```
Out[14]: (10, 6)
```

```
In [15]: x_train.shape
```

```
Out[15]: (40, 6)
```

```
In [16]: y_train.shape
```

```
Out[16]: (40, 1)
```

```
In [17]: y_test.shape
```

```
Out[17]: (10, 1)
```

```
In [18]: from sklearn.linear_model import LinearRegression  
mlr=LinearRegression()  
mlr.fit(x_train,y_train)
```

```
Out[18]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
In [ ]: import pickle  
pickle.dump(mlr,open('multilinear.pkl','wb'))
```

```
In [18]: y_pred=mlr.predict(x_test)
```

In [20]: x_test

```
Out[20]: array([[0.000000e+00, 1.000000e+00, 0.000000e+00, 6.605152e+04,
 1.8264556e+05, 1.181482e+05],
 [1.000000e+00, 0.000000e+00, 0.000000e+00, 1.0067196e+05,
 9.1790610e+04, 2.4974455e+05],
 [0.000000e+00, 1.000000e+00, 0.000000e+00, 1.0191308e+05,
 1.1059411e+05, 2.2916095e+05],
 [0.000000e+00, 1.000000e+00, 0.000000e+00, 2.7892920e+04,
 8.4710770e+04, 1.6447071e+05],
 [0.000000e+00, 1.000000e+00, 0.000000e+00, 1.5344151e+05,
 1.0114555e+05, 4.0793454e+05],
 [0.000000e+00, 0.000000e+00, 1.000000e+00, 7.2107600e+04,
 1.2786455e+05, 3.5318381e+05],
 [0.000000e+00, 0.000000e+00, 1.000000e+00, 2.0229590e+04,
 6.5947930e+04, 1.8526510e+05],
 [0.000000e+00, 0.000000e+00, 1.000000e+00, 6.1136380e+04,
 1.5270192e+05, 8.8218230e+04],
 [0.000000e+00, 1.000000e+00, 0.000000e+00, 7.3994560e+04,
 1.2278275e+05, 3.0331926e+05],
 [0.000000e+00, 1.000000e+00, 0.000000e+00, 1.4210734e+05,
 9.1391770e+04, 3.6616842e+05]])
```

In [21]: y_test

```
Out[21]: array([[103282.38],
 [144259.4 ],
 [146121.95],
 [ 77798.83],
 [191050.39],
 [105008.31],
 [ 81229.06],
 [ 97483.56],
 [110352.25],
 [166187.94]])
```

```
In [19]: y_pred
```

```
Out[19]: array([[103015.20159794],
               [132582.27760817],
               [132447.73845174],
               [ 71976.09851258],
               [178537.48221058],
               [116161.24230168],
               [ 67851.69209676],
               [ 98791.73374684],
               [113969.43533015],
               [167921.06569553]])
```

```
In [22]: from sklearn.metrics import r2_score
accuracy=r2_score(y_test,y_pred)
```

```
In [23]: accuracy
```

```
Out[23]: 0.9347068473282567
```

```
In [27]: #Random prediction, state col has been spliited into 3 cols and 1st enter it's value then rest
y=mlr.predict([[0,0,1,12345,123456,123456]])#0,0,1 is for 1 is Florida and 6 cols so 6 values
```

```
In [28]: y
```

```
Out[28]: array([[61381.51739101]])
```

```
In [29]: y[0][0]
```

```
Out[29]: 61381.51739101054
```

```
In [ ]:
```