jupyter pract_3_4 Last Checkpoint: 2 minutes ago

File Edit View Run Kernel Settings Help Not Trusted

Code JupyterLab Python 3 (ipykernel)

[7]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[8]:
```
df=pd.read_csv("Salary_Data.csv")
```

[3]:
```
df.head(2)
```

[3]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205

[9]:
```
df.tail(2)
```

[9]:

	YearsExperience	Salary
28	10.3	122391
29	10.5	121872

[10]:
```
df.isnull().sum()
```

[10]:
```
YearsExperience    0
Salary             0
dtype: int64
```

[6]:
```
df.notnull().sum()
```

[6]:
```
YearsExperience    30
Salary             27
dtype: int64
```

[11]:
```
df.isnull().sum()
```

[11]:
```
YearsExperience    0
Salary             0
dtype: int64
```

[12]:
```
df.dropna(inplace=True)
```

[13]:
```
df.corr() # correlation
```

[13]:

	YearsExperience	Salary
YearsExperience	1.000000	0.978242

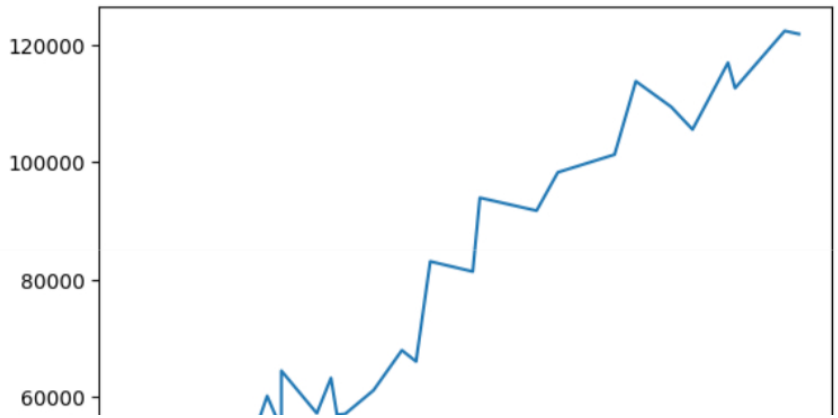
	Salary	0.978242	1.000000

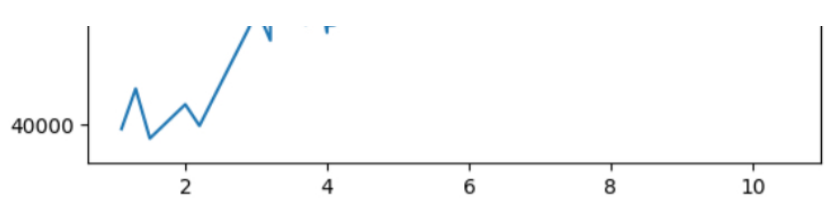
[14]:
```
X=df.iloc[:,:-1].values # independent variable
y = df.iloc[:, -1].values # dependent variable
```

[34]:
```
X
```

[34]:
```
array([[ 1.1],
       [ 1.3],
       [ 1.5],
       [ 2. ],
       [ 2.2],
       [ 2.9],
       [ 3. ],
       [ 3.2],
       [ 3.2],
       [ 3.7],
       [ 3.9],
       [ 4. ],
       [ 4. ],
       [ 4.1],
       [ 5.3],
       [ 5.9],
       [ 6. ],
       [ 6.8],
       [ 7.1],
       [ 7.9],
       [ 8.2],
       [ 8.7],
       [ 9. ],
       [ 9.5],
       [ 9.6],
       [10.3],
       [10.5]])
```

[15]:
```
# graph
plt.plot(X,y)
plt.show()
```





[16]:
```
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)
```

[18]:
```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train, y_train)
```

[18]:
```
LinearRegression()
```

[19]:
```
pred = lm.predict(X_test)
```

[16]:
```
print("y_test",y_test)
print("X_test",X_test)
```

```
y_test [ 37731. 112635.  83088.  91738.  56642.  55794.]
X_test [[1.5]
 [9.6]
 [5.3]
 [6.8]
 [2.9]
 [4. ]]
```

[20]:
```
pred
```

[20]:
```
array([ 40748.96184072, 122699.62295594,  64961.65717022,  63099.14214487,
       115249.56285456, 107799.50275317])
```

[21]:
```
print(lm.intercept())
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[21], line 1
----> 1 print(lm.intercept())

AttributeError: 'LinearRegression' object has no attribute 'intercept'
```

[22]:
```
a=lm.intercept_
c=lm.coef_
```

[23]:
```
print(a,c)
```

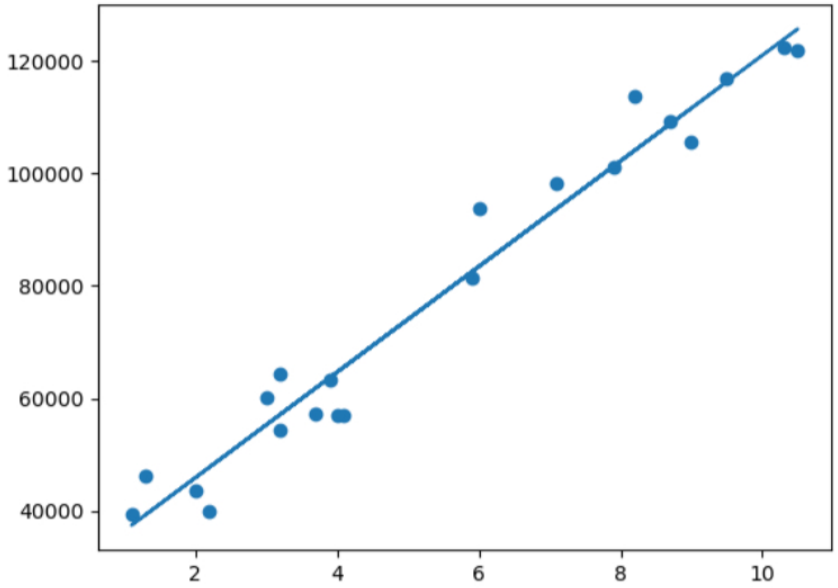
```
26780.09915062818 [9312.57512673]
```

[24]:
```
my_sal_pred=a*10+c  # prediction of salary for 10 year exp
print(my_sal_pred)
```

```
[277113.56663301]
```

[22]:
```
# graph od train and test data
plt.scatter(X_train, y_train)
plt.plot(X_train, lm.predict(X_train))
```

[22]:
```
[<matplotlib.lines.Line2D at 0x17e1718e010>]
```



[23]:
```
from sklearn import metrics
```

[24]:
```
print('Mean Absolute Error is : ',metrics.mean_absolute_error(y_test,pred))
```

```
Mean Absolute Error is :  4374.731786040949
```

[25]:
```
print('Mean Squared Error is :',metrics.mean_squared_error(y_test,pred))
```

```
Mean Squared Error is : 26104141.43339284
```

[26]:
```
print('Root Mean Squared Error is: ',np.sqrt(metrics.mean_squared_error(y_test,pred)))
```

```
Root Mean Squared Error is:  5109.2212159381825
```

[27]:
```
#root Mean Squared Error (RMSE)
```

[28]:
```
train_score_lm = lm.score(X_train, y_train)
test_score_lm = lm.score(X_test, y_test)

print("Train score: ", train_score_lm)
print("Test score : ",test_score_lm)
```

```
Train score:  0.9633907320629322
```

```
Test score :  0.9591199103412812
```

[ ]:
```
from sklearn.metrics import r2_score
print(" Root mean Squared error is:",r2_score(y_test,pred) )
r2=r2_score(y_test,pred)
```

[32]:
```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
col_transf = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [])], remainder='passthrough')
X = np.array(col_transf.fit_transform(X))
```

[33]:
```
X
```

[33]:
```
array([[ 1.1],
       [ 1.3],
       [ 1.5],
       [ 2. ],
       [ 2.2],
       [ 2.9],
       [ 3. ],
       [ 3.2],
       [ 3.2],
       [ 3.7],
       [ 3.9],
       [ 4. ],
       [ 4. ],
       [ 4.1],
       [ 5.3],
       [ 5.9],
       [ 6. ],
       [ 6.8],
       [ 7.1],
       [ 7.9],
       [ 8.2],
       [ 8.7],
       [ 9. ],
       [ 9.5],
       [ 9.6],
       [10.3],
       [10.5]])
```

[35]:
```
from sklearn.model_selection import train_test_split
Xm_train, Xm_test, ym_train, ym_test = train_test_split(X, y, test_size = 0.3, random_state = 45)
```

[36]:
```
from sklearn.linear_model import LinearRegression
lm1 = LinearRegression()
lm1.fit(Xm_train, ym_train)
```

[36]:
```
LinearRegression()
```

[37]:
```
ym = lm1.predict(Xm_test)
np.set_printoptions(precision=2)
print(np.concatenate((ym.reshape(len(ym),1), ym_test.reshape(len(ym_test),1)),1))
```

```
[[ 64179.34  63218.  ]
```

```
 [113452.36 105582.  ]
 [127944.42 121872.  ]
 [ 84468.23  93940.  ]
 [ 66111.61  57081.  ]
 [119249.19 112635.  ]
 [ 57416.37  54445.  ]
 [ 62247.06  57189.  ]
 [118283.05 116969.  ]]
```

[ ]: