ML_01

January 20, 2022

1 Machine Learning (Lecture-2)

1.1 Linear regression practice

1.1.1 Two variables problem

- One dependent and one independent variable
- Equation (y=a+bx)
  - x= independent variable
  - y= dependent variable
    - a= constant/intercept(a is the constant or the y intercept. It is the value of the dependent variable when x = 0)
    - b= function/slop of x (b is the coefficient of the independent variable. It is also known as the slope and gives the rate of change of the dependent variable)

1.1.2 Install libraries

- Use pip if you are uisng windows

[ ]:
```
#pip install numpy
#pip install pandas
#pip install scikit-learn
```

1.2 Step-1

1.2.1 Import Libraries

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

1.3 Step-2 loaddataset

1.3.1 Load Dataset

- It is better to keep the dataset in the same folder in which you have your notebook, otherwise you have to enter the complete path

[ ]:
```
# load dataset
df = pd.read_csv("ml_data_salary.csv")
df.head()
```

[ ]:

	age	distance	YearsExperience	Salary
0	31.1	77.75	1.1	39343
1	31.3	78.25	1.3	46205
2	31.5	78.75	1.5	37731
3	32.0	80.00	2.0	43525
4	32.2	80.50	2.2	39891

[ ]:
```
X= df[['YearsExperience']]
y= df['Salary']
```

1.4 Step-3 Split dataset

- training and testing

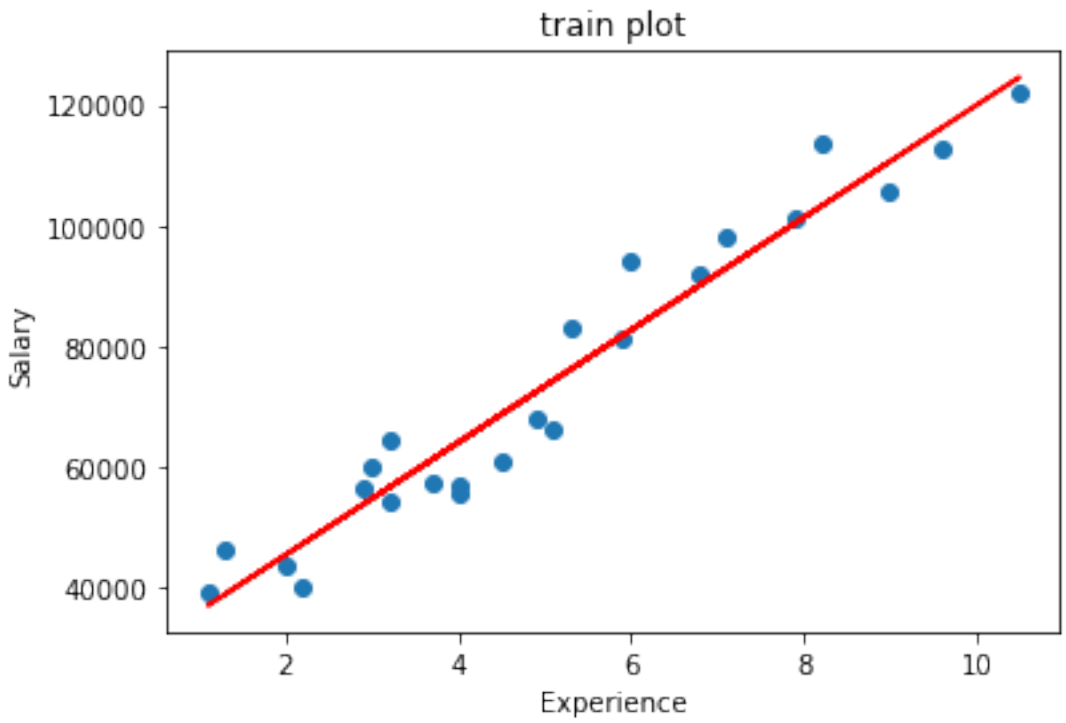
[ ]:
```
#split dataset
from sklearn.model_selection import train_test_split
X_train, X_test,y_train,y_test= train_test_split(X,y,test_size=0.
 ↪2,random_state=0)
```

1.5 Step-4 Create model and fit

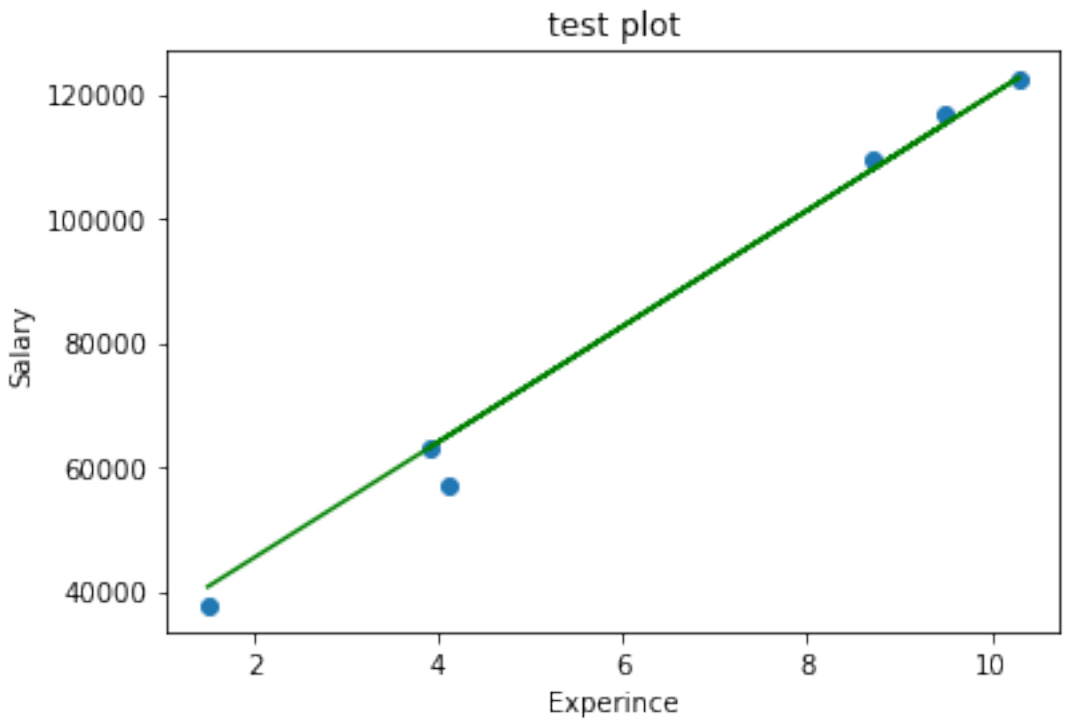
[ ]:
```
from sklearn.linear_model import LinearRegression
model= LinearRegression()
model=model.fit(X_train, y_train)
```

1.6 Step-5 plotting relation

[ ]:
```
#plotting
import matplotlib.pyplot as plt
plt.scatter(X_train,y_train)
plt.plot(X_train, model.predict(X_train),color='red')
plt.xlabel("Experience")
plt.ylabel('Salary')
plt.title("train plot")
plt.show()
```



[ ]:
```
#plotting
import matplotlib.pyplot as plt
plt.scatter(X_test,y_test)
plt.plot(X_test, model.predict(X_test),color='Green')
plt.xlabel("Experince")
plt.ylabel('Salary')
plt.title("test plot")
plt.show()
```



1.7 Step-6 Evalution of model performance

[ ]:
```
#evaluation (model fitness)
print ('score for train model=',model.score(X_test,y_test))
print ('score for test model=',model.score(X_train,y_train))
```

```
score for train model= 0.988169515729126
score for test model= 0.9411949620562126
```

1.8 Step-7 Prediction

[ ]:
```
#prediction of unknown values
model.predict([[5]])
```

```
C:\Anaconda\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have
valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

[ ]:
```
array([73342.97478427])
```

[ ]:
```
#prediction of unknown values for test dataset
model.predict(X_test)
```

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

[ ]:
```
#prediction of unknown values for array
model.predict([[5],[10],[15],[20]])
```

```
C:\Anaconda\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have
valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

[ ]:
```
array([ 73342.97478427, 119905.85041792, 166468.72605157, 213031.60168521])
```

[ ]:
```
x= ([5],[10],[15],[20])
model.predict(x)
```

```
C:\Anaconda\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have
valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

[ ]:
```
array([ 73342.97478427, 119905.85041792, 166468.72605157, 213031.60168521])
```