

Machine learning

1- Simple Linear Regression

Step 1: import dataset

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

```
In [3]: df = pd.read_csv("ml_data_salary.csv")
df.head()
```

```
Out[3]:
```

| | 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 |

```
In [5]: pip install scikit-learn
```

Requirement already satisfied: scikit-learn in h:\download\anaconda\lib\site-packages (0.24.2)
Requirement already satisfied: numpy>=1.13.3 in h:\download\anaconda\lib\site-packages (from scikit-learn) (1.20.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in h:\download\anaconda\lib\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: joblib>=0.11 in h:\download\anaconda\lib\site-packages (from scikit-learn) (1.1.0)
Requirement already satisfied: scipy>=0.19.1 in h:\download\anaconda\lib\site-packages (from scikit-learn) (1.7.1)
Note: you may need to restart the kernel to use updated packages.

Step2: Splitting dataset into training data and testing data

```
In [16]: x = df[["YearsExperience"]]
y = df[["Salary"]]
```

```
In [17]: x.head()
```

```
Out[17]:
```

| | YearsExperience |
|---|-----------------|
| 0 | 1.1 |
| 1 | 1.3 |
| 2 | 1.5 |
| 3 | 2.0 |

| YearsExperience | |
|-----------------|-----|
| 4 | 2.2 |

In [18]: `y.head()`

Out[18]:

| | Salary |
|---|--------|
| 0 | 39343 |
| 1 | 46205 |
| 2 | 37731 |
| 3 | 43525 |
| 4 | 39891 |

In [28]:

```
#import library & split data

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)
```

Step-3:Fit Linear Regression Model

In [30]:

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

Out[30]: LinearRegression()

step-4: importing visualization library

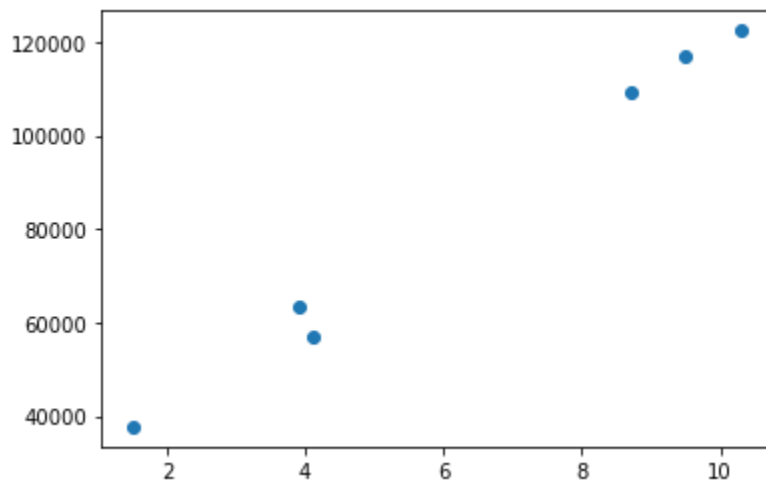
In [31]:

```
import matplotlib.pyplot as plt
```

In [33]:

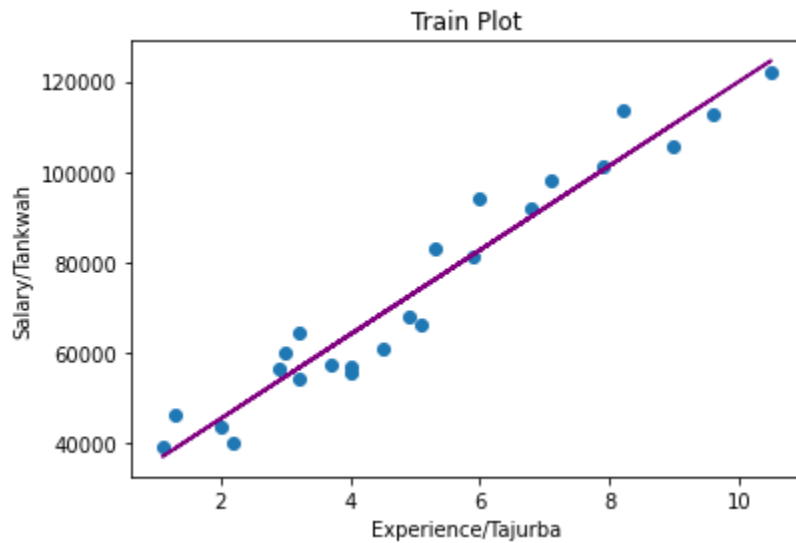
```
plt.scatter(X_test,y_test)
```

Out[33]: <matplotlib.collections.PathCollection at 0x264ac534f40>



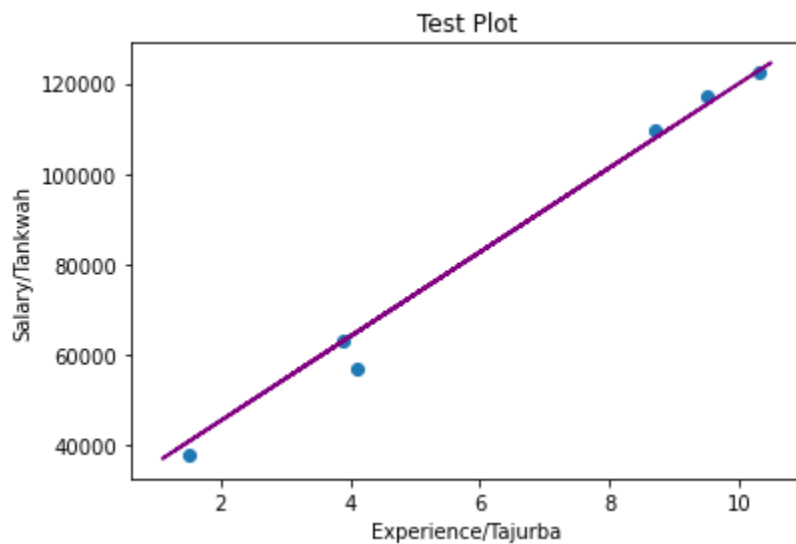
In [39]:

```
plt.scatter(X_train,y_train)
plt.plot(X_train , model.predict(X_train), color = "purple")
plt.xlabel("Experience/Tajurba")
plt.ylabel("Salary/Tankwah")
plt.title("Train Plot")
plt.show()
```



In [40]:

```
plt.scatter(X_test,y_test)
plt.plot(X_train , model.predict(X_train), color = "purple")
plt.xlabel("Experience/Tajurba")
plt.ylabel("Salary/Tankwah")
plt.title("Test Plot")
plt.show()
```



Step 5: Testing and Evaluating Your model

```
In [41]: #Model Fitness  
  
model.score(X_test,y_test)
```

```
Out[41]: 0.988169515729126
```

```
In [42]: model.score(X_train,y_train)
```

```
Out[42]: 0.9411949620562126
```

```
In [45]: print("Score of training data = ",model.score(X_train,y_train))  
print("Score of testing data = ",model.score(X_test,y_test))
```

```
Score of training data = 0.9411949620562126  
Score of testing data = 0.988169515729126
```

Step 6: Prediction of unknown values

```
In [47]: model.predict([[5]])
```

```
Out[47]: array([[73342.97478427]])
```

```
In [48]: model.predict([[1]])
```

```
Out[48]: array([[36092.67427736]])
```

```
In [49]: model.predict([[20]])
```

```
Out[49]: array([[213031.60168521]])
```

```
In [50]: model.predict(X_test)
```

```
Out[50]: array([[ 40748.96184072],
               [122699.62295594],
               [ 64961.65717022],
               [ 63099.14214487],
               [115249.56285456],
               [107799.50275317]])
```

```
In [51]: ## how to check multiple values at once
```

```
model.predict([[3],[2],[1],[7]])
```

```
Out[51]: array([[54717.82453082],
               [45405.24940409],
               [36092.67427736],
               [91968.12503773]])
```

```
In [52]: l = [10],[20],[30],[6],[9]
model.predict(l)
```

```
Out[52]: array([[119905.85041792],
               [213031.60168521],
               [306157.3529525 ],
               [ 82655.549911  ],
               [110593.27529119]])
```

```
In [ ]:
```