Linear Regression Model

Step 1 - *Import dataset*

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

df = pd.read_csv("Salary.csv")
df
```

Out[1]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891
5	2.9	56642
6	3.0	60150
7	3.2	54445
8	3.2	64445
9	3.7	57189
10	3.9	63218
11	4.0	55794
12	4.0	56957
13	4.1	57081
14	4.5	61111
15	4.9	67938
16	5.1	66029
17	5.3	83088
18	5.9	81363
19	6.0	93940
20	6.8	91738
21	7.1	98273
22	7.9	101302
23	8.2	113812
24	8.7	109431

	YearsExperience	Salary
25	9.0	105582
26	9.5	116969
27	9.6	112635
28	10.3	122391
29	10.5	121872
30	11.2	127345
31	11.5	126756
32	12.3	128765
33	12.9	135675
34	13.5	139465

Step 2 - *Splitting dataset into training data and testing data*

In [9]:
```
x = df[["YearsExperience"]]     # Independent variable
y = df["Salary"]              # Dependent variable
```

In [10]:
```
x.head()
```

Out[10]:

	YearsExperience
0	1.1
1	1.3
2	1.5
3	2.0
4	2.2

In [11]:
```
y.head()
```

Out[11]:
```
0    39343
1    46205
2    37731
3    43525
4    39891
Name: Salary, dtype: int64
```

Importing Library

In [12]:
```
from sklearn.model_selection import train_test_split
```

In [18]:
```
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 [19]:
```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model = model.fit(x_train, y_train)
model
```

Out[19]:
```
LinearRegression()
```

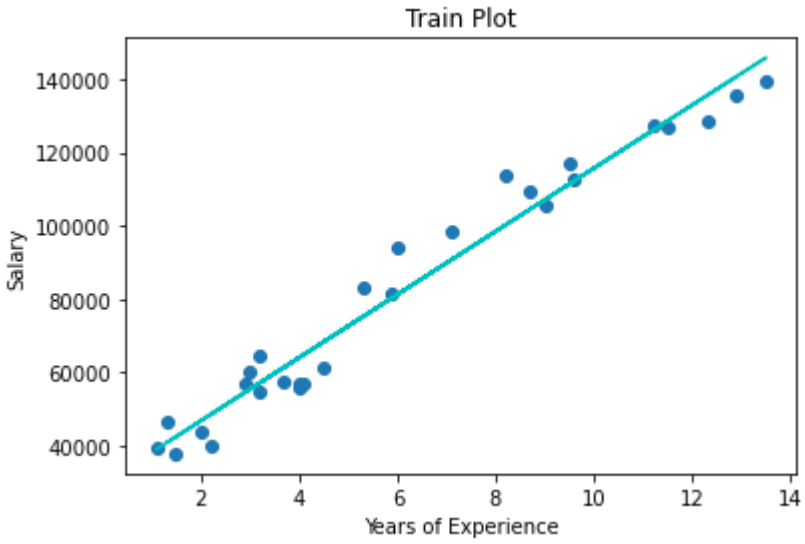
Step 4 - Plotting

In [26]:
```
import matplotlib.pyplot as plt

plt.scatter(x_train, y_train)
plt.plot(x_train, model.predict(x_train), color="c")

plt.xlabel("Years of Experience")
plt.ylabel("Salary")
plt.title("Train Plot")


plt.show()
```



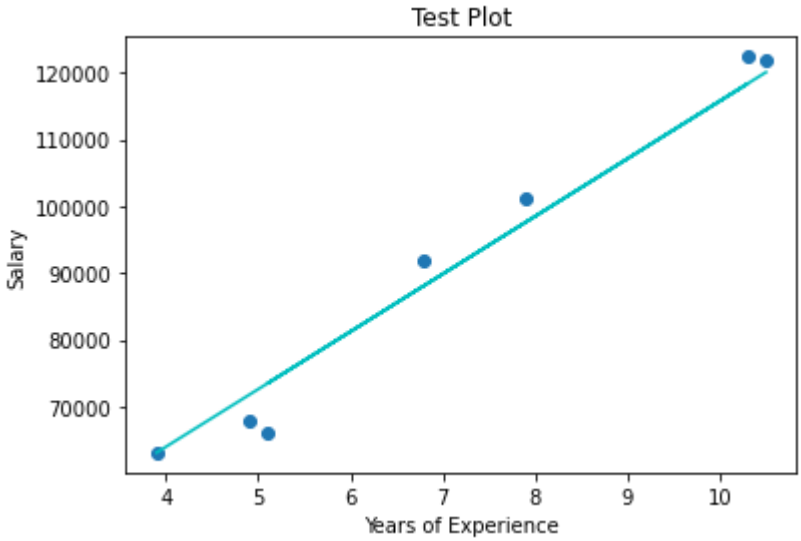
In [27]:
```
import matplotlib.pyplot as plt

plt.scatter(x_test, y_test)
plt.plot(x_test, model.predict(x_test), color="c")

plt.xlabel("Years of Experience")
plt.ylabel("Salary")
```

```
plt.title("Test Plot")


plt.show()
```



Step 5 - Testing or Evaluating Model

In [31]:
```
# Model Fitness - Training

print("Score for Training Data -", model.score(x_train, y_train))
print("Score for Testing Data -", model.score(x_test, y_test))
```

```
Score for Training Data - 0.9638371903672509
Score for Testing Data - 0.9708090245443415
```

Step 6 - Prediction of unknown values

In [35]:
```
model.predict([[10]])
```

Out[35]:
```
array([115742.98052455])
```

In [36]:
```
model.predict(x_test)
```

Out[36]:
```
array([120057.87672477,  88127.64484315,  73456.99776241, 118331.91824468,
        97620.41648363,  71731.03928232,  63101.24688189])
```

In [37]:
```
model.predict([[5],[10],[2],[2.5]])
```

Out[37]:
```
array([ 72594.01852237, 115742.98052455,  46704.64132106,  51019.53752127])
```