Basics of Linear Regression

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [2]:
```
df = pd.read_csv("ml_data_salary.csv")
df.head()
```

Out[2]:

	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 [3]:
```
df.drop(["age", "distance"], axis=1, inplace=True)
df.head()
```

Out[3]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

In [4]:
```
df.rename(columns={"YearsExperience":"Years of Exprience"}, inplace=True)
df.head()
```

Out[4]:

	Years of Exprience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

In [5]:
```
x = df[["Years of Exprience"]]
y = df["Salary"]
```

In [6]:
```
X_train, X_test, y_train, y_test = train_test_split(x,y, train_size=0.8, rand
```

In [7]:
```
m = LinearRegression().fit(X_train, y_train)
m
```

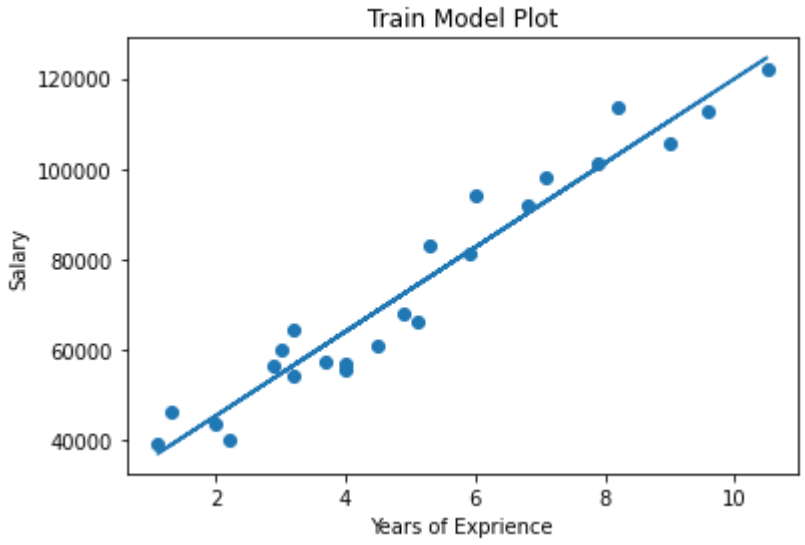
Out[7]:
```
LinearRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [8]:
```
df.head()
```

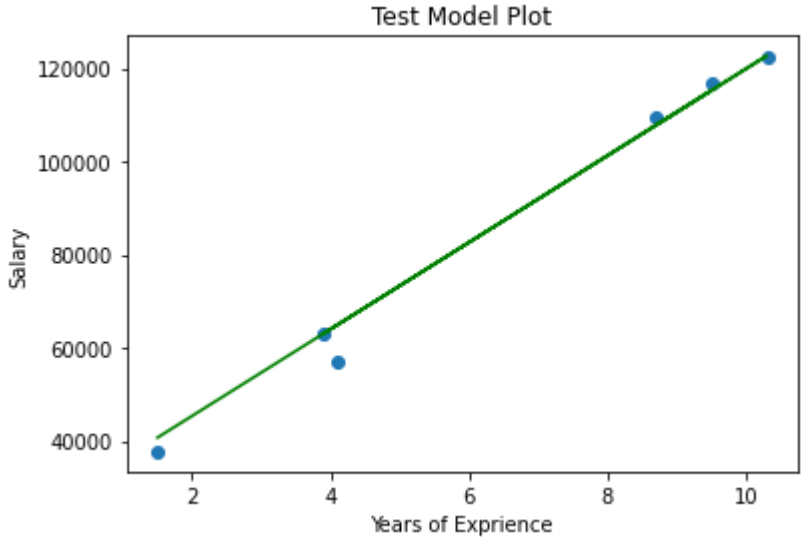
Out[8]:

	Years of Exprience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

In [9]:
```
plt.scatter(X_train, y_train)
plt.plot(X_train, m.predict(X_train))
plt.title("Train Model Plot")
plt.xlabel("Years of Exprience")
plt.ylabel("Salary")
plt.show()
```



In [10]:
```
plt.scatter(X_test, y_test)
plt.plot(X_test, m.predict(X_test), color="Green")
plt.title("Test Model Plot")
plt.xlabel("Years of Exprience")
plt.ylabel("Salary")
plt.show()
```



Regression Score

In [11]:
```
print("Train Model Rgression Score is = ",m.score(X_train, y_train))
print("Test Model Regression Score is = ",m.score(X_test, y_test))
```

```
Train Model Rgression Score is =  0.9411949620562126
Test Model Regression Score is =  0.988169515729126
```

Prediciton of unkown values

In [12]:
```
Predictions = m.predict(X_test)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: