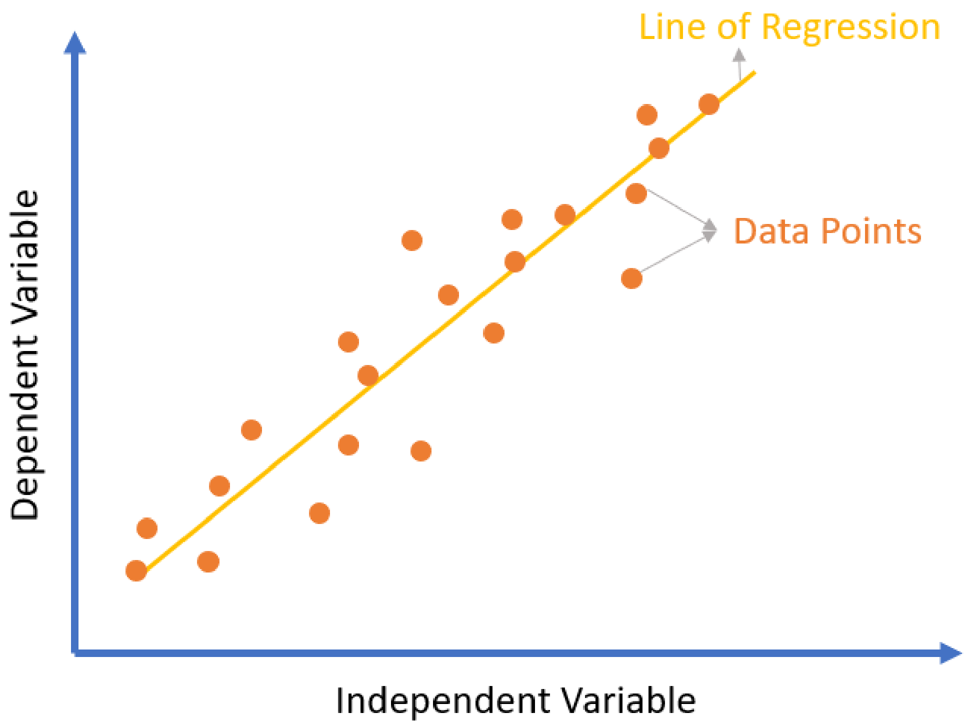
1. Linear Regression

- Supervised Learning Model
- Mainly used for Regression tasks
- Suitable for predicting continuous target variables
- Line of Best Fit

Linear Regression is a popular and widely used supervised learning algorithm used for predicting continuous target variables based on one or more input features. It assumes a linear relationship between the input variables (features) and the output variable (target).



Assumptions of Linear Regression:

- **Linearity:** It assumes a linear relationship between the input features and the target variable.
- **Independence:** The input features should be independent of each other (no multicollinearity).
- **Homoscedasticity:** The residuals (the differences between the actual and predicted values) should have a constant variance across all levels of the input variables.
- **Normality:** The residuals should follow a normal distribution.

Simple Linear Regression:

Simple Linear Regression is the basic form of Linear Regression involving a single input feature (X) and a single target variable (y). The relationship is represented by the equation:

y = b0 + b1*X

where,

- y is the target variable.
- X is the input feature.
- b0 is the y-intercept (the value of y when X is zero).
- b1 is the slope (the change in y for a one-unit change in X).

The goal is to estimate the values of b0 and b1 that best fit the given data. This is typically done by minimizing the sum of squared errors (SSE) or by maximizing the likelihood function.

Multiple Linear Regression:

Multiple Linear Regression extends the simple linear regression to include multiple input features (X1, X2, ..., Xn) and a single target variable (y). The relationship is represented by the equation:

y = b0 + b1*X1 + b2*X2 + ... + bn*Xn

where:

- y is the target variable.
- X1, X2, ..., Xn are the input features.
- b0 is the y-intercept.
- b1, b2, ..., bn are the slopes associated with each input feature.

The goal remains the same: to estimate the values of b0, b1, b2, ..., bn that best fit the given data.

Model Evaluation:

To assess the performance of a linear regression model, several evaluation metrics are commonly used:

- **Mean Squared Error (MSE):** It measures the average squared difference between the predicted and actual values. A lower MSE indicates better model performance.
- **Root Mean Squared Error (RMSE):** It is the square root of the MSE and provides the measure of the average prediction error in the same units as the target variable.
- **R-squared (R2) Score:** It represents the proportion of the variance in the target variable that can be explained by the model. It ranges from 0 to 1, with 1 indicating a perfect fit.
- **Adjusted R-squared Score:** It adjusts the R-squared score by considering the number of input features and the sample size. It penalizes the addition of irrelevant features.

Limitations of Linear Regression:

Linear Regression has certain limitations that should be considered:

- **Linearity Assumption:** Linear Regression assumes a linear relationship between the input features and the target variable. If the relationship is non-linear, Linear Regression may not provide accurate predictions.
- **Sensitive to Outliers:** Linear Regression is sensitive to outliers, as they can significantly impact the estimated coefficients and the model's performance.
- **Assumptions Violation:** If the assumptions of Linear Regression (linearity, independence, homoscedasticity, normality) are violated, the model's performance may be affected.
- **Multicollinearity:** Linear Regression assumes independence between input features. When features are highly correlated (multicollinearity), it can lead to unstable and unreliable coefficient estimates.
- **Limited to Linear Relationships:** Linear Regression is not suitable for capturing complex non-linear relationships between features and the target variable.

Applications of Linear Regression:

- **Economics and Finance:** Linear Regression is extensively used in economic analysis, financial modeling, and forecasting. It can help analyze the relationship between economic variables, predict stock prices, estimate demand and supply, evaluate the impact of policies, and assess risk.

- **Marketing and Sales:** Linear Regression is employed in market research and sales forecasting. It can assist in understanding the factors influencing consumer behavior, predicting product demand, optimizing pricing strategies, and measuring the effectiveness of marketing campaigns.
- **Social Sciences:** Linear Regression is used in social science research to analyze relationships between variables. It can help examine factors affecting education outcomes, assess social and economic disparities, study population trends, and analyze survey data.
- **Healthcare:** Linear Regression is applied in healthcare for various purposes, including analyzing the impact of medical treatments, predicting patient outcomes, modeling disease progression, and estimating healthcare costs.
- **Real Estate:** Linear Regression can be utilized in real estate for property price prediction, rental price estimation, assessing market trends, and evaluating the impact of location and property characteristics.
- **Environmental Science:** Linear Regression is used in environmental studies to analyze the relationships between environmental variables, predict pollution levels, model climate change, and assess the impact of human activities on ecosystems.
- **Engineering and Manufacturing:** Linear Regression is employed in engineering and manufacturing for quality control, process optimization, predicting equipment failure, and optimizing production efficiency.

Implementing Linear Regression using Python

Dataset Required:

https://drive.google.com/file/d/1uesxH_CQprom9HqwhspvoesUyHo4KA7h/view?usp=sharing

Importing Required Libraries:

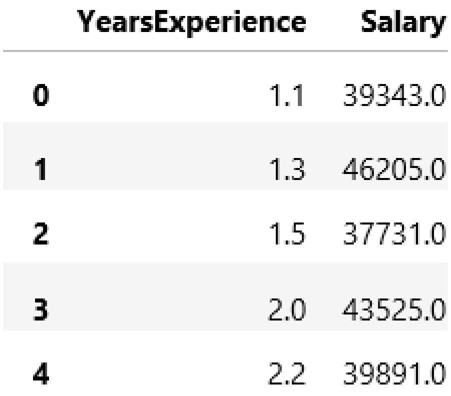
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

Importing (Reading) Datasets:

```
data=pd.read_csv('/content/Salary_Data.csv')
```

Exploring Dataset:

```
data.head()
```



	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

```
data.shape
```

```
(30, 2)
```

Checking for any null values in dataset:

```
data.isnull().sum()
#Checking for any Null values in the imported Datasets
```

```
YearsExperience    0
Salary             0
dtype: int64
```

Assigning Dependent and Independent variables:

```
x=data.iloc[:,:1].values
y=data.iloc[:, 1:2].values
```

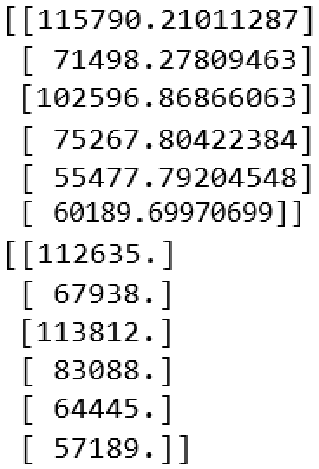
Splitting the dataset into Training and Testing Dataset:

```
x_train, x_test, y_train, y_test = train_test_split(x,y,
      test_size=0.2, random_state = 42)
```

Fitting the Model (Linear Regression):

```
model=LinearRegression()
model.fit(x_train, y_train)
y_pred=model.predict(x_test)
```

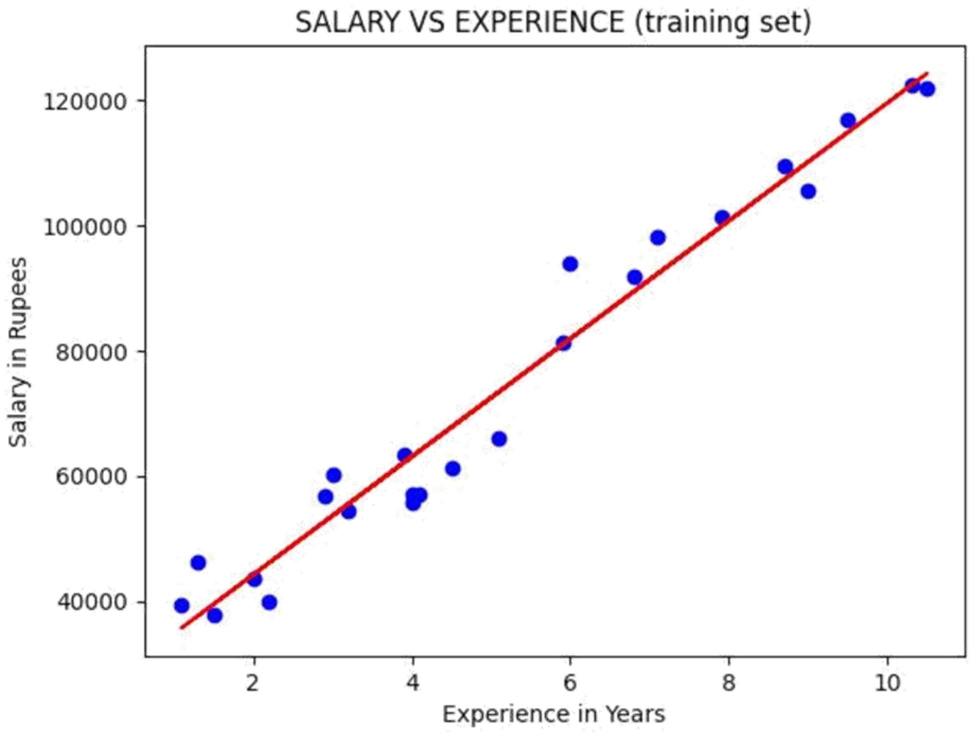
```
print(y_pred)
print(y_test)
```



```
[[115790.21011287]
 [ 71498.27809463]
 [102596.86866063]
 [ 75267.80422384]
 [ 55477.79204548]
 [ 60189.69970699]]
[[112635.]
 [ 67938.]
 [113812.]
 [ 83088.]
 [ 64445.]
 [ 57189.]]
```

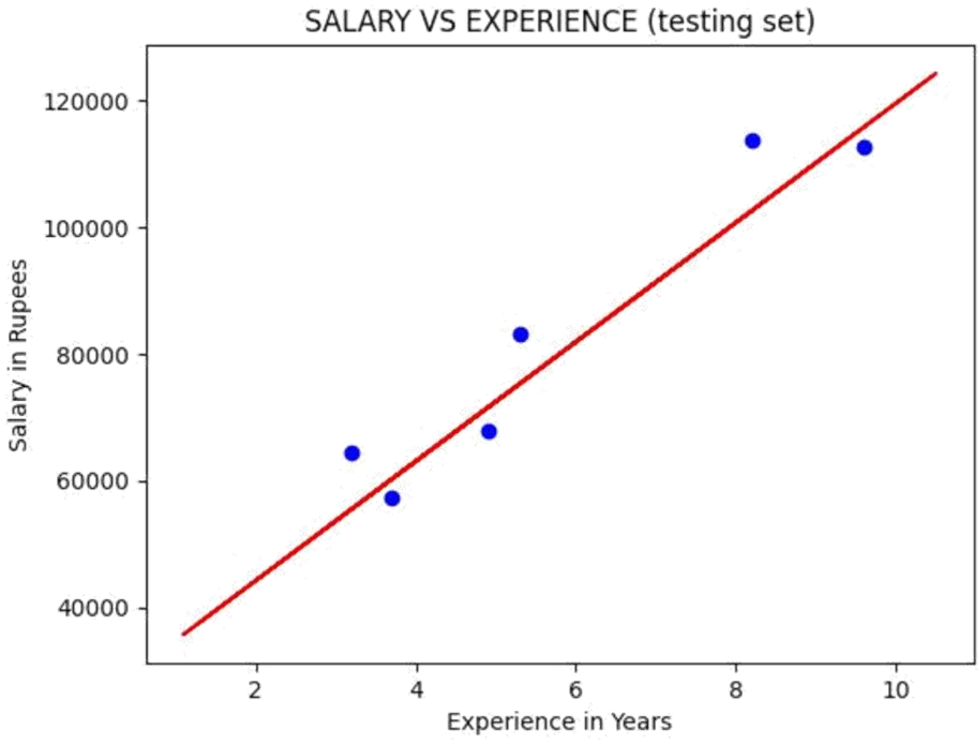
Plot for Training dataset

```
plt.scatter(x_train, y_train, color='blue')
plt.plot(x_train, model.predict(x_train), color='red')
plt.title('SALARY VS EXPERIENCE (training set)')
plt.xlabel('Experience in Years')
plt.ylabel('Salary in Rupees')
plt.show()
```



Plot for Testing dataset

```
plt.scatter(x_test, y_test, color='blue')
plt.plot(x_train, model.predict(x_train), color='red')
plt.title('SALARY VS EXPERIENCE (testing set)')
plt.xlabel('Experience in Years')
plt.ylabel('Salary in Rupees')
plt.show()
```


SALARY VS EXPERIENCE (testing set)
Salary in Rupees
Experience in Years