

```
In [2]: import numpy as np
import pandas as pd
```

```
In [3]: df = pd.read_csv("C:/Users/SW20407278/Desktop/Final AI/Hands-On/Regression/Salary_Data.csv")
```

```
In [4]: df.head()
```

```
Out[4]:
```

	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

```
In [5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31 entries, 0 to 30
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   YearsExperience  30 non-null    float64
 1   Salary          30 non-null    float64
dtypes: float64(2)
memory usage: 624.0 bytes
```

```
In [6]: ## Dropping of null values
df.dropna(inplace=True)
```

```
In [7]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   YearsExperience  30 non-null    float64
 1   Salary          30 non-null    float64
dtypes: float64(2)
memory usage: 720.0 bytes
```

```
In [8]: ## Statistical summary
df.describe()
```

```
Out[8]:
```

	YearsExperience	Salary
count	30.000000	30.000000
mean	5.313333	76003.000000
std	2.837888	27414.429785
min	1.100000	37731.000000
25%	3.200000	56720.750000
50%	4.700000	65237.000000
75%	7.700000	100544.750000
max	10.500000	122391.000000

```
In [9]: ## Features and Label
features = df.iloc[:,[0]].values
label = df.iloc[:,[1]].values
```

```
In [18]: ## Creation of Train_Test_split
from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test = train_test_split(features,
                                                  label,
                                                  test_size=0.2,
                                                  random_state=23)
```

```
In [19]: ## Model Building
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(X_train,y_train)
```

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

```
In [20]: ## Training Accuracy
model.score(X_train,y_train)
```

```
Out[20]: 0.9603182547438908
```

```
In [21]: ## Testing Accuracy

model.score(X_test,y_test)
```

```
Out[21]: 0.9184170849214232
```

```
In [22]: ## Co-efficient
model.coef_
```

```
Out[22]: array([[9281.30847068]])
```

```
In [23]: ## Intercept
model.intercept_
```

```
Out[23]: array([27166.73682891])
```

```
In [ ]: ## Equation of Line  
## salary = 9281.30847068 + 27166.73682891 * YearsExperience
```

```
In [21]: ## Deployment with pickle  
  
import pickle  
pickle.dump(model , open('SalaryPred.model' , 'wb'))
```

```
In [ ]:
```