

## EXPERIMENT 7

### Aim:

To build and train a simple linear regression model to predict salary based on years of experience using Python's Scikit-learn library.

### Algorithm:

1. Import the required libraries — NumPy, Pandas, Scikit-learn, and Pickle.
2. Load the dataset and inspect its contents using `read_csv()`.
3. Check data information using `info()` and ensure there are no missing values.
4. Describe statistical properties of the dataset using `describe()`.
5. Split the dataset into training and testing sets using `train_test_split()`.
6. Train a linear regression model using `LinearRegression().fit()`.
7. Evaluate the model performance using `score()` for both training and testing data.
8. Display model parameters — coefficient and intercept.
9. Save and load the model using Pickle.
10. Predict the salary for a given number of years of experience and display the output.

### Code:

```
import numpy as np
import pandas as pd
df = pd.read_csv('Salary_data - Salary_data.csv')
df
```

### Output:

	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

	YearsExperience	Salary
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	113381
24	8.7	109431
25	9.0	105582
26	9.5	116969
27	9.6	112635
28	10.3	122391
29	10.5	121872

---

```
df.info()
```

**Output:**

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 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  int64
```

```
dtypes: float64(1), int64(1)
```

```
memory usage: 612.0 bytes
```

---

```
df.dropna(inplace=True)
```

```
df.info()
```

### Output:

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 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  int64
```

```
dtypes: float64(1), int64(1)
```

```
memory usage: 612.0 bytes
```

---

```
df.describe()
```

### Output:

	YearsExperience	Salary
count	30.000000	30.000000
mean	5.313333	76003.000000
std	2.837888	27414.429785
min	1.100000	37731.000000

### YearsExperience Salary

25%	3.200000	56720.750000
50%	4.700000	65237.000000
75%	7.700000	100544.750000
max	10.500000	122391.000000

---

```
features = df.iloc[:, [0]].values
```

```
label = df.iloc[:, [1]].values
```

```
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=42)
```

```
from sklearn.linear_model import LinearRegression
```

```
model = LinearRegression()
```

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

(No output for this cell)

---

```
model.score(x_train, y_train)
```

**Output:**

0.9645401573418146

---

```
model.score(x_test, y_test)
```

**Output:**

0.9024461774180497

---

```
model.coef_
```

**Output:**

array([[9423.81532303]])

---

```
model.intercept_
```

**Output:**

```
array([25321.58301178])
```

---

```
import pickle

pickle.dump(model, open('SalaryPred.model', 'wb'))

model = pickle.load(open('SalaryPred.model', 'rb'))

yr_of_exp = 44

yr_of_exp_NP = np.array([[yr_of_exp]])

Salary = model.predict(yr_of_exp_NP)

print("Estimated Salary for {} years of experience is {}".format(yr_of_exp, Salary))
```

**Output:**

```
Estimated Salary for 44 years of experience is [[439969.45722514]]:
```

**Result:**

Thus, the Python program to predict salary using a simple linear regression model was successfully executed. The model was trained, evaluated, and tested, and it accurately predicted salary based on years of experience.