```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

saldf=pd.read_csv('/content/drive/MyDrive/Life Expectancy Data.csv')
print("Columns available in saldf:", saldf.columns.tolist())
```

```
Columns available in saldf: ['Country', 'Year', 'Status', 'Life
expectancy ', 'Adult Mortality', 'infant deaths', 'Alcohol',
'percentage expenditure', 'Hepatitis B', 'Measles ', ' BMI ', 'under-
five deaths ', 'Polio', 'Total expenditure', 'Diphtheria ', '
HIV/AIDS', 'GDP', 'Population', ' thinness  1-19 years', ' thinness 5-
9 years', 'Income composition of resources', 'Schooling']
```

```
saldf_clean = saldf.dropna()

status_encoder = LabelEncoder()

saldf_clean["Status_Enc"] =
status_encoder.fit_transform(saldf_clean["Status"])
```

```
/tmp/ipython-input-1498511424.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  saldf_clean["Status_Enc"] =
status_encoder.fit_transform(saldf_clean["Status"])
```

```
X = saldf_clean[["Schooling"]]
y = saldf_clean["Life expectancy "]

LR = LinearRegression()
LR.fit(X, y)
```

```
LinearRegression()
```

```
pred = LR.predict([[5]])
print("Predicted Salary for 5 years experience:", pred[0])
```

```
Predicted Salary for 5 years experience: 52.99927434764797

/usr/local/lib/python3.12/dist-packages/sklearn/utils/
validation.py:2739: UserWarning: X does not have valid feature names,
but LinearRegression was fitted with feature names
  warnings.warn(
sample_status = status_encoder.inverse_transform([0])[0]
print("Decoded Status (0):", sample_status)
```

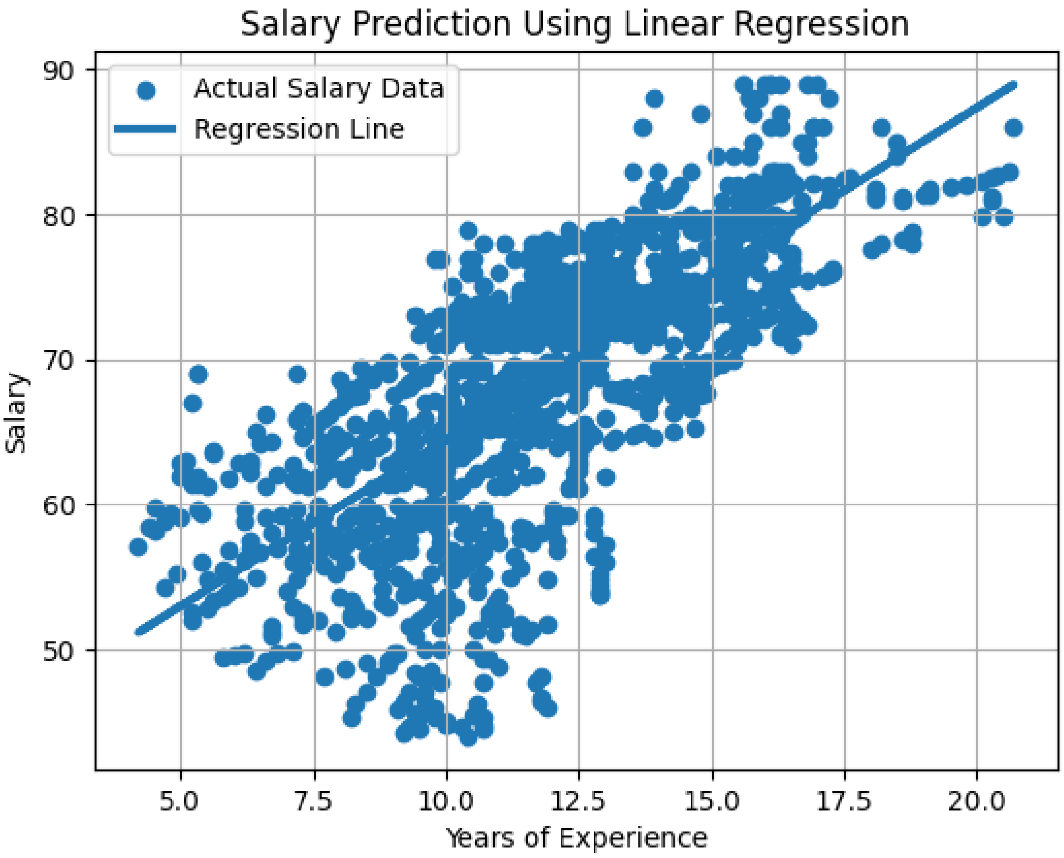
```
Decoded Status (0): Developed
```

```
plt.figure(figsize=(10,6))
```

```
<Figure size 1000x600 with 0 Axes>
```

```
<Figure size 1000x600 with 0 Axes>
```

```
plt.scatter(X, y, label="Actual Salary Data")
plt.plot(X, LR.predict(X), label="Regression Line", linewidth=3)
plt.xlabel("Years of Experience")
plt.ylabel("Salary")
plt.title("Salary Prediction Using Linear Regression")
plt.legend()
plt.grid(True)
plt.show()
```


Salary Prediction Using Linear Regression
● Actual Salary Data
Regression Line
Salary
Years of Experience