

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

load data

```
sald = pd.read_csv("Salary Data.csv")
```

```
sald.head()
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

Next steps:

[Generate code with sald](#)
[View recommended plots](#)
[New interactive sheet](#)

```
sald.isnull().sum()
```

	0
Age	2
Gender	2
Education Level	2
Job Title	2
Years of Experience	2
Salary	2

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

```
sald.isnull().sum()
```

	0
Age	0
Gender	0
Education Level	0
Job Title	0
Years of Experience	0
Salary	0

```
le_gender = LabelEncoder()
```

```
le_edu = LabelEncoder()
```

```
sald['Gender'] = le_gender.fit_transform(sald['Gender'])
sald['Education Level'] = le_edu.fit_transform(sald['Education Level'])
sald.head()
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	1	0	Software Engineer	5.0	90000.0
1	28.0	0	1	Data Analyst	3.0	65000.0
2	45.0	1	2	Senior Manager	15.0	150000.0
3	36.0	0	0	Sales Associate	7.0	60000.0
4	52.0	1	1	Director	20.0	200000.0

Next steps: [Generate code with sald](#) [View recommended plots](#) [New interactive sheet](#)

```
ind = sald[['Age', 'Gender', 'Education Level', 'Years of Experience']]
dep = sald['Salary']
```

```
LR = LinearRegression()
LR.fit(ind, dep)
LinearRegression()
```

```
LinearRegression()
LinearRegression()
```

```
Q=LR.predict([[21,0,2,5]])
print("SALARY IS",Q)
```

```
SALARY IS [55308.20108323]
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but Line
warnings.warn(
```

```
age = float(input("Enter Age: "))
ge=int(input("enter 0 for female,1 for male:"))
el=int(input("enter 0 for bachelors,1 for masters,2 for phd:"))
yoe=float(input("enter years of experience:"))
```

```
Enter Age: 55
enter 0 for female,1 for male:1
enter 0 for bachelors,1 for masters,2 for phd:2
enter years of experience:15
```

```
W=LR.predict([[age,ge,el,yoe]])
print("SALARY IS",W)
```

```
SALARY IS [187749.43333719]
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but Line
warnings.warn(
```

```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(modelpred)
```

```
sald.index
```

```
Index([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
...
365, 366, 367, 368, 369, 370, 371, 372, 373, 374],
dtype='int64', length=373)
```

```
age = float(input("Enter Age: "))
ge_input = input("Enter Gender (Male/Female): ").strip().capitalize()



el_input = input("Enter Education Level (Bachelors/Masters/PhD): ").strip().lower()
if el_input == "phd":
    el_input = "PhD"
else:
    el_input = el_input.capitalize()
```

```
yoe = float(input("Enter Years of Experience: "))
```

```
Enter Age: 55
Enter Gender (Male/Female): male
Enter Education Level (Bachelors/Masters/PhD): phd
Enter Years of Experience: 15
```

```
ge = le_gender.transform([ge_input])[0]
e1 = le_edu.transform([e1_input])[0]
```

```
W = LR.predict([[age, ge, e1, yoe]])
print("Predicted SALARY IS ₹", round(W[0], 2))
```

 Predicted SALARY IS ₹ 187749.43
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but Line
warnings.warn(


```
from sklearn.metrics import mean_squared_error
modelpred = LR.predict(ind)
mse = mean_squared_error(dep, modelpred)
print("Mean Squared Error:", round(mse, 2))
```

 Mean Squared Error: 231199075.25