CHRIST

(DEEMED TO BE UNIVERSITY)

DELHI-NCR, INDIA

AI/ML Programming

MCA-475

Assignment – 03

BY

HIMANSHU HEDA (24225013)

SUBMITTED TO

Dr. Manjula Shannhog

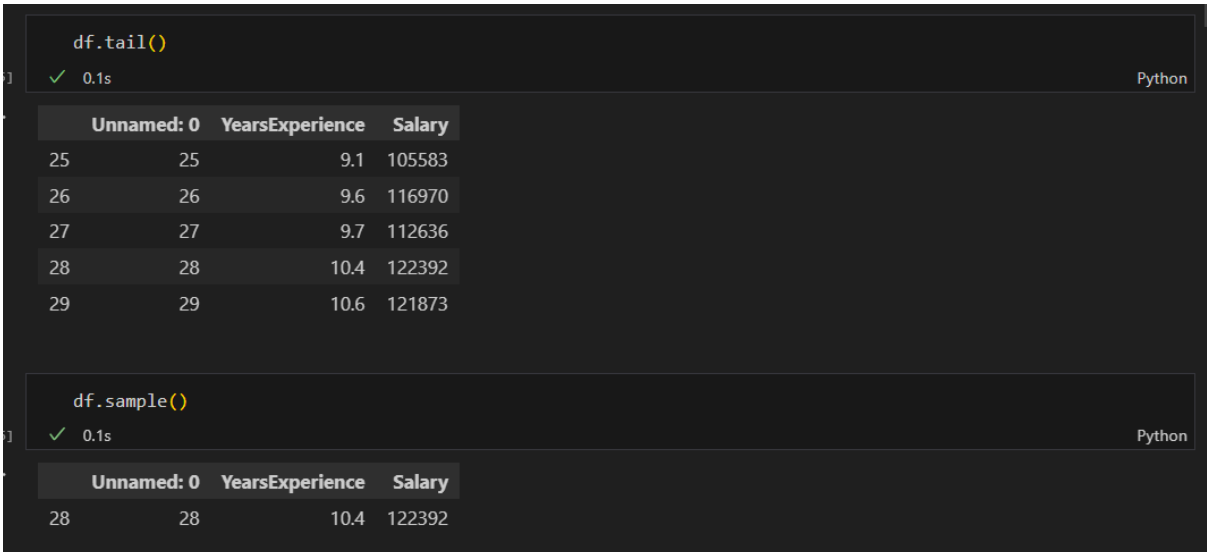
SCHOOL OF SCIENCES

2024-25

Importing Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import date
```

```
df = pd.read_csv('./Dataset/Salary_dataset.csv')
```

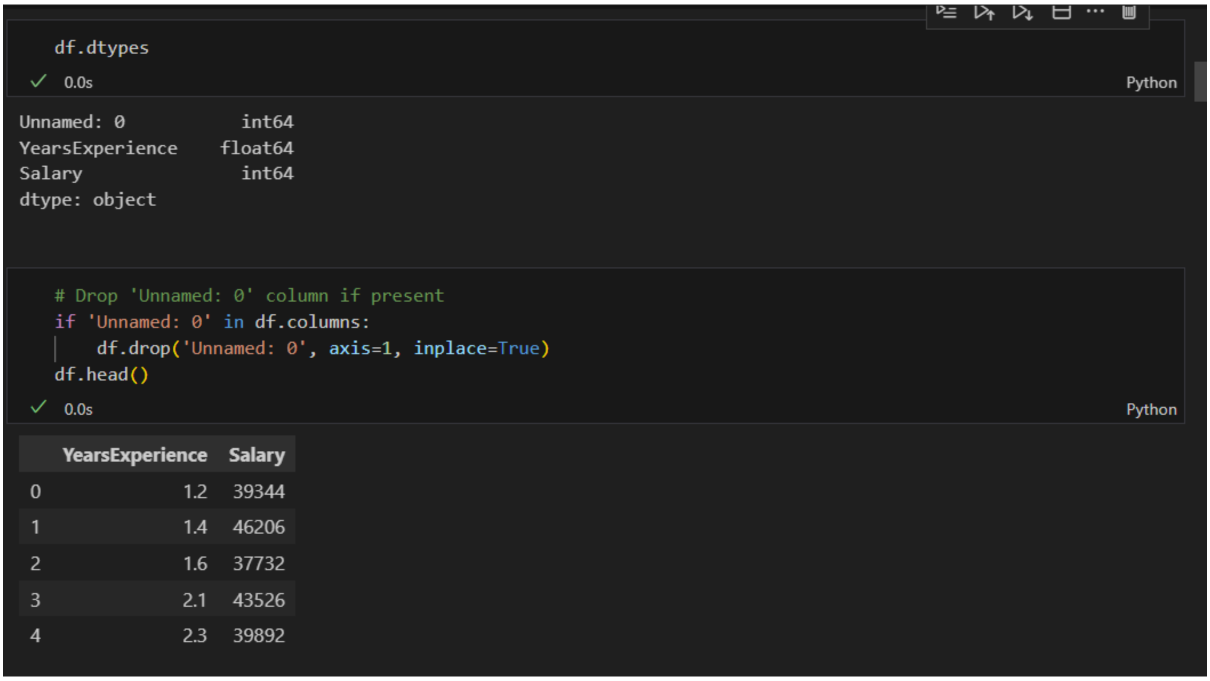


```
df.tail()
```

	Unnamed: 0	YearsExperience	Salary
25	25	9.1	105583
26	26	9.6	116970
27	27	9.7	112636
28	28	10.4	122392
29	29	10.6	121873

```
df.sample()
```

	Unnamed: 0	YearsExperience	Salary
28	28	10.4	122392

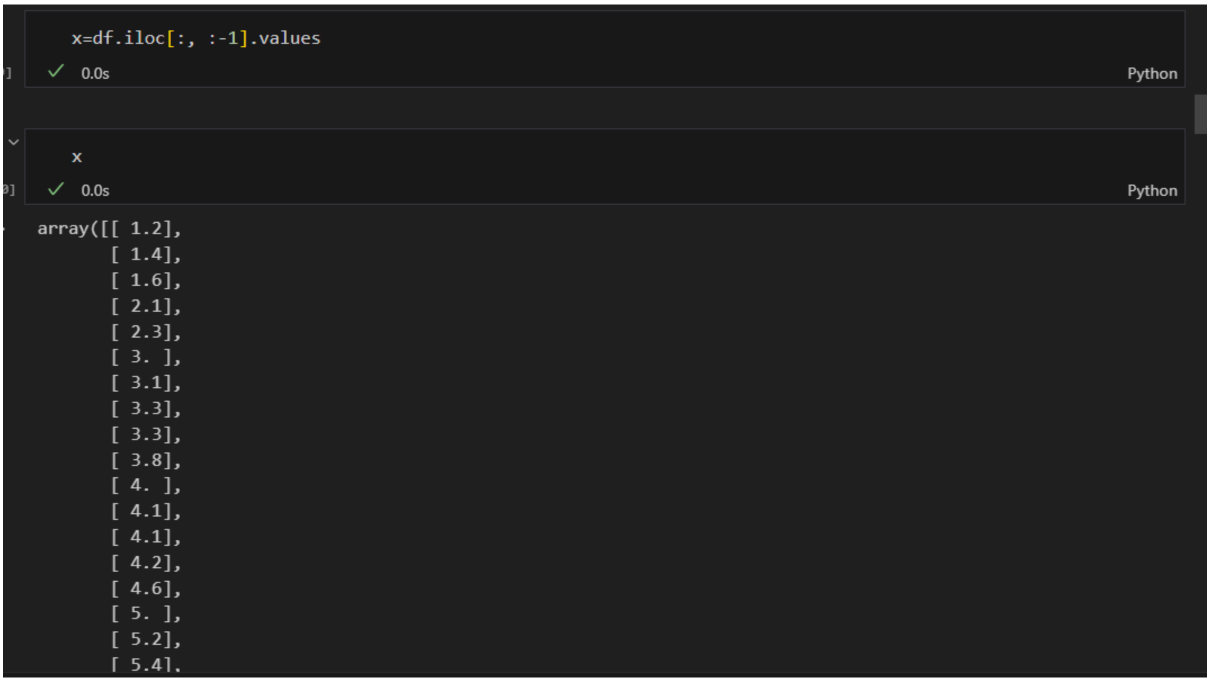


```
df.dtypes
```

```
Unnamed: 0            int64
YearsExperience     float64
Salary                int64
dtype: object
```

```
# Drop 'Unnamed: 0' column if present
if 'Unnamed: 0' in df.columns:
    df.drop('Unnamed: 0', axis=1, inplace=True)
df.head()
```

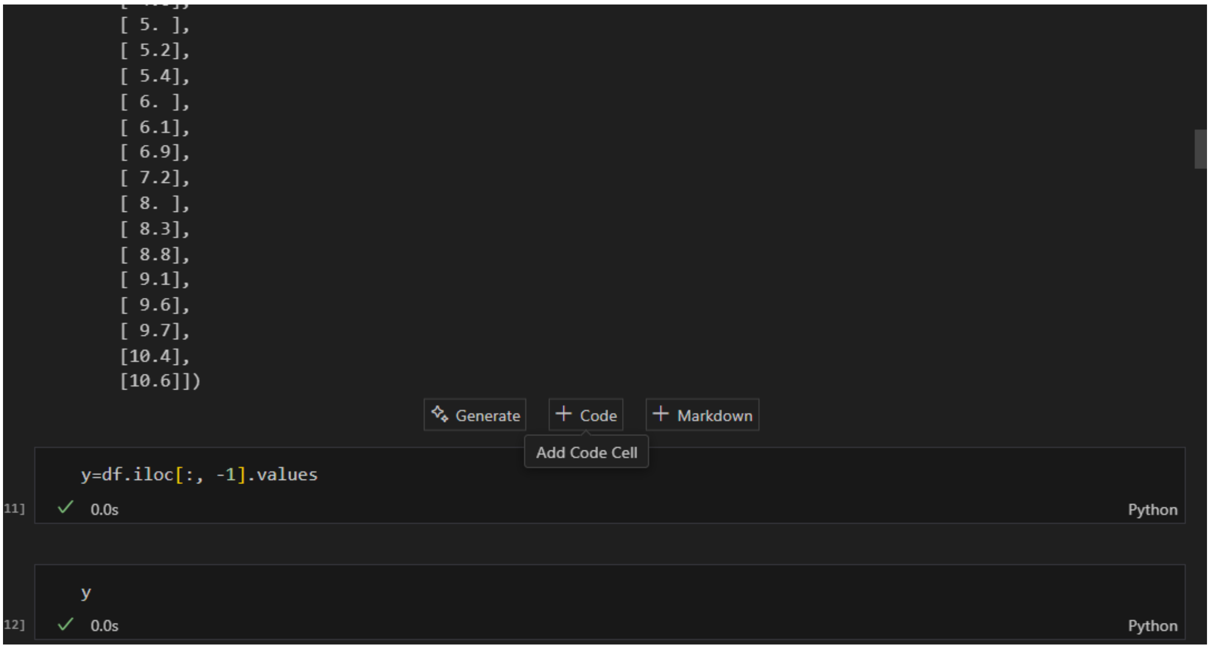
	YearsExperience	Salary
0	1.2	39344
1	1.4	46206
2	1.6	37732
3	2.1	43526
4	2.3	39892



```
x=df.iloc[:, :-1].values
```

```
x
```

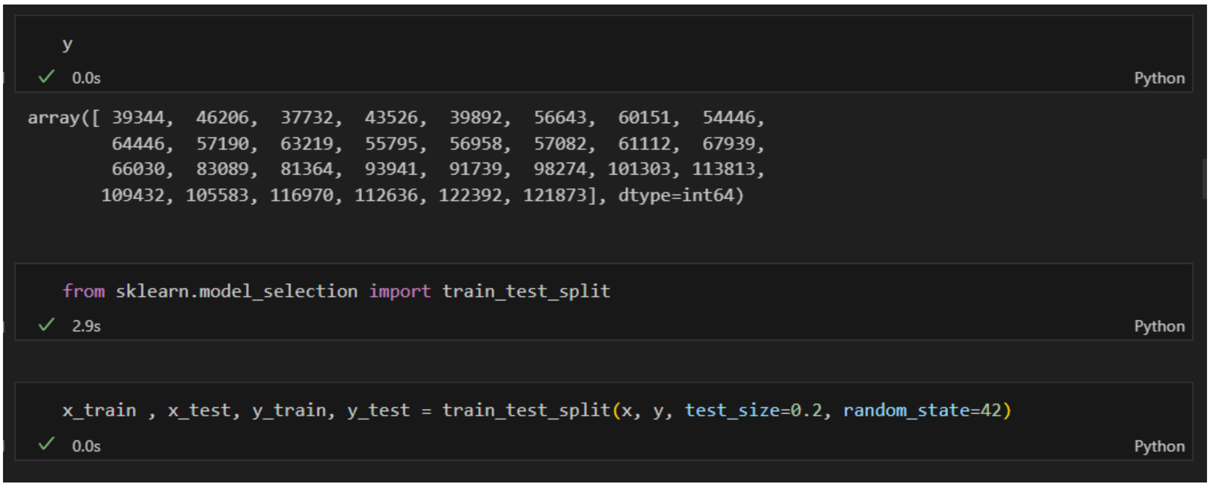
```
array([[ 1.2],
       [ 1.4],
       [ 1.6],
       [ 2.1],
       [ 2.3],
       [ 3. ],
       [ 3.1],
       [ 3.3],
       [ 3.3],
       [ 3.8],
       [ 4. ],
       [ 4.1],
       [ 4.1],
       [ 4.2],
       [ 4.6],
       [ 5. ],
       [ 5.2],
       [ 5.4],
```



```
       [ 5. ],
       [ 5.2],
       [ 5.4],
       [ 6. ],
       [ 6.1],
       [ 6.9],
       [ 7.2],
       [ 8. ],
       [ 8.3],
       [ 8.8],
       [ 9.1],
       [ 9.6],
       [ 9.7],
       [10.4],
       [10.6]])
```

```
y=df.iloc[:, -1].values
```

```
y
```

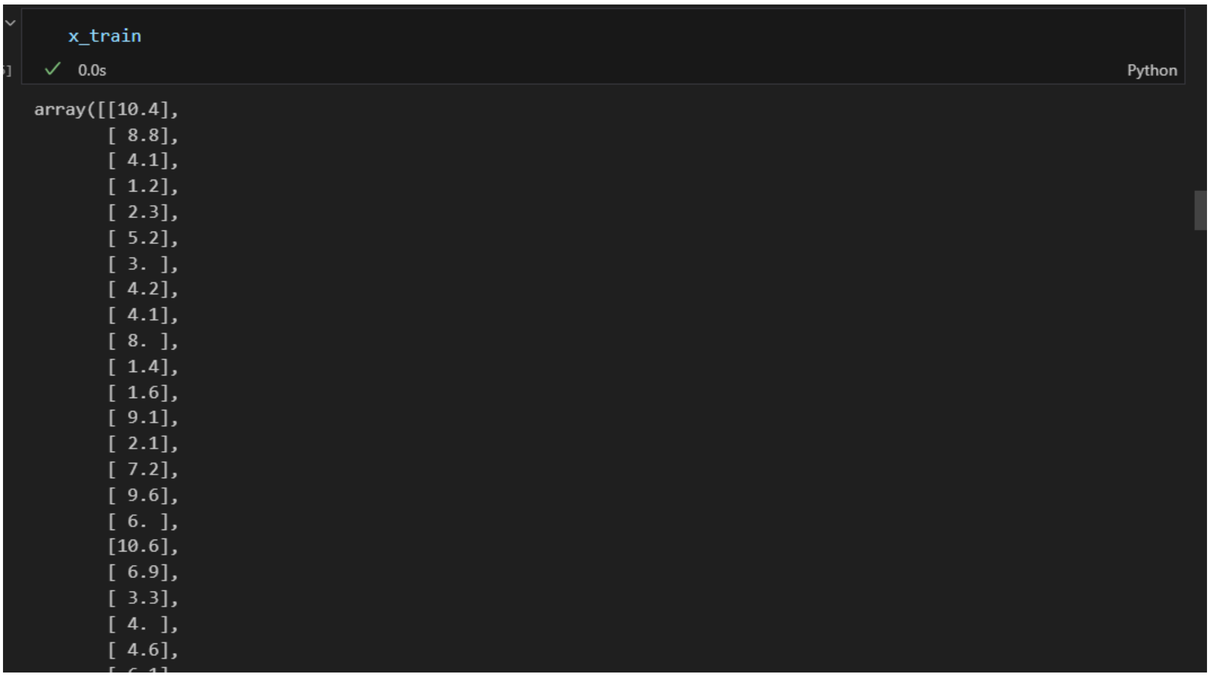


```
y
```

```
array([ 39344,  46206,  37732,  43526,  39892,  56643,  60151,  54446,
        64446,  57190,  63219,  55795,  56958,  57082,  61112,  67939,
        66030,  83089,  81364,  93941,  91739,  98274, 101303, 113813,
       109432, 105583, 116970, 112636, 122392, 121873], dtype=int64)
```

```
from sklearn.model_selection import train_test_split
```

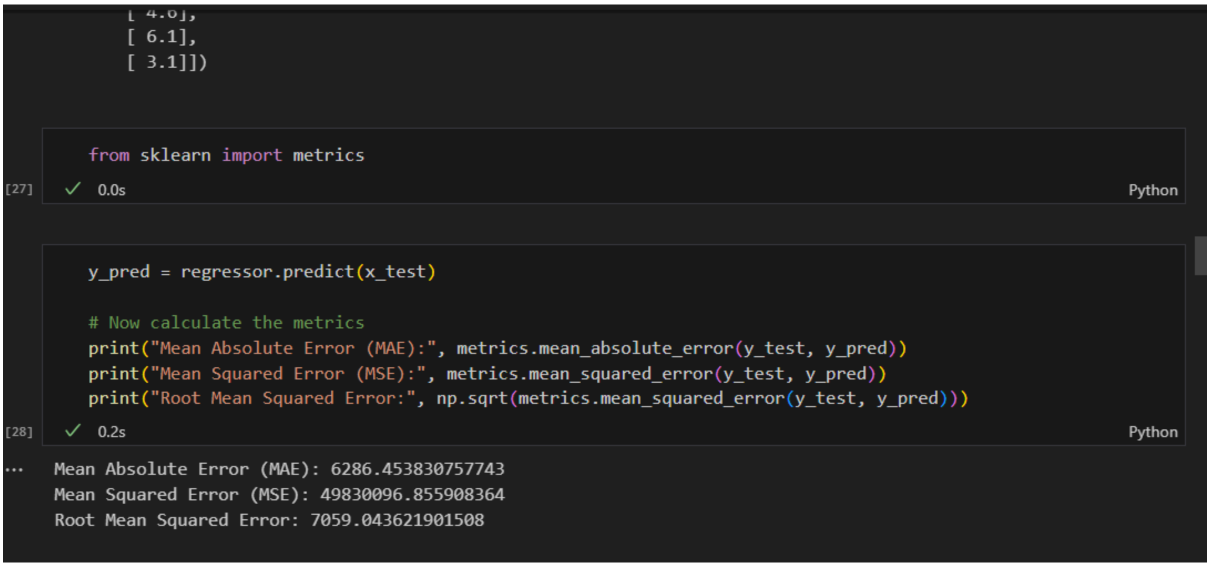
```
x_train , x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```



```
x_train
```

✓ 0.0s Python

```
array([[10.4],
       [ 8.8],
       [ 4.1],
       [ 1.2],
       [ 2.3],
       [ 5.2],
       [ 3. ],
       [ 4.2],
       [ 4.1],
       [ 8. ],
       [ 1.4],
       [ 1.6],
       [ 9.1],
       [ 2.1],
       [ 7.2],
       [ 9.6],
       [ 6. ],
       [10.6],
       [ 6.9],
       [ 3.3],
       [ 4. ],
       [ 4.6],
```



```
       [ 6.1],
       [ 3.1]])
```

```
from sklearn import metrics
```

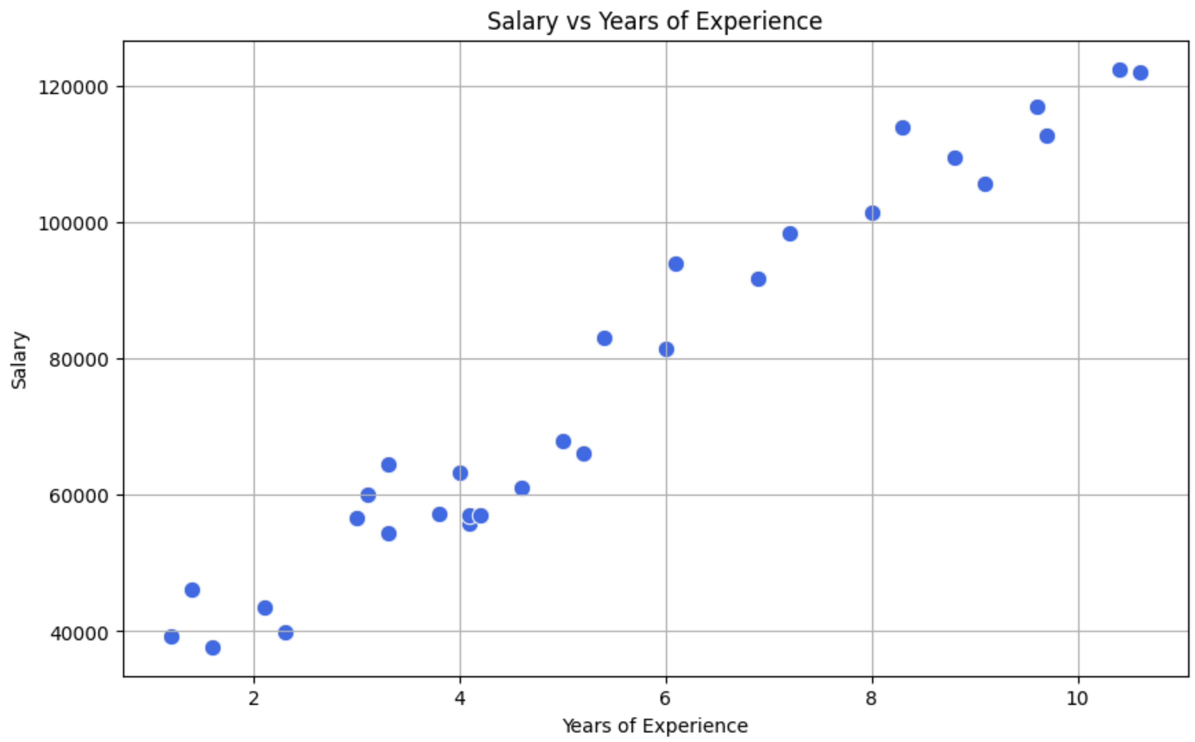
[27] ✓ 0.0s Python

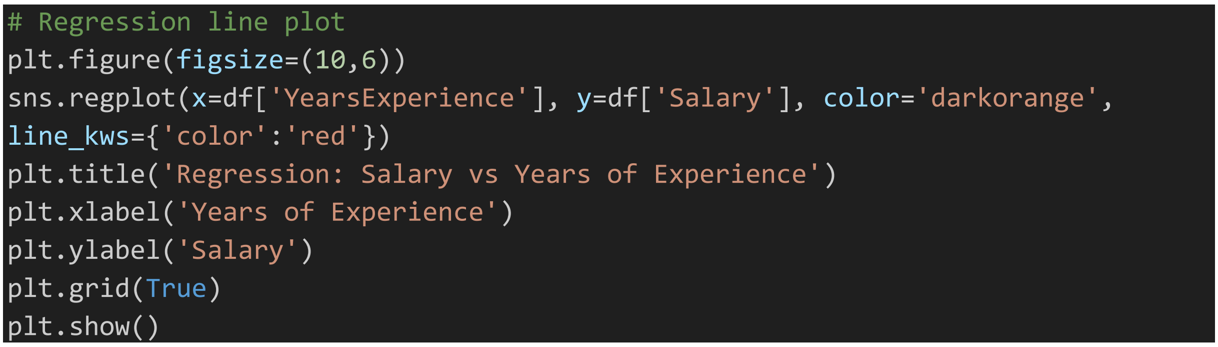
```
y_pred = regressor.predict(x_test)

# Now calculate the metrics
print("Mean Absolute Error (MAE):", metrics.mean_absolute_error(y_test, y_pred))
print("Mean Squared Error (MSE):", metrics.mean_squared_error(y_test, y_pred))
print("Root Mean Squared Error:", np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

[28] ✓ 0.2s Python

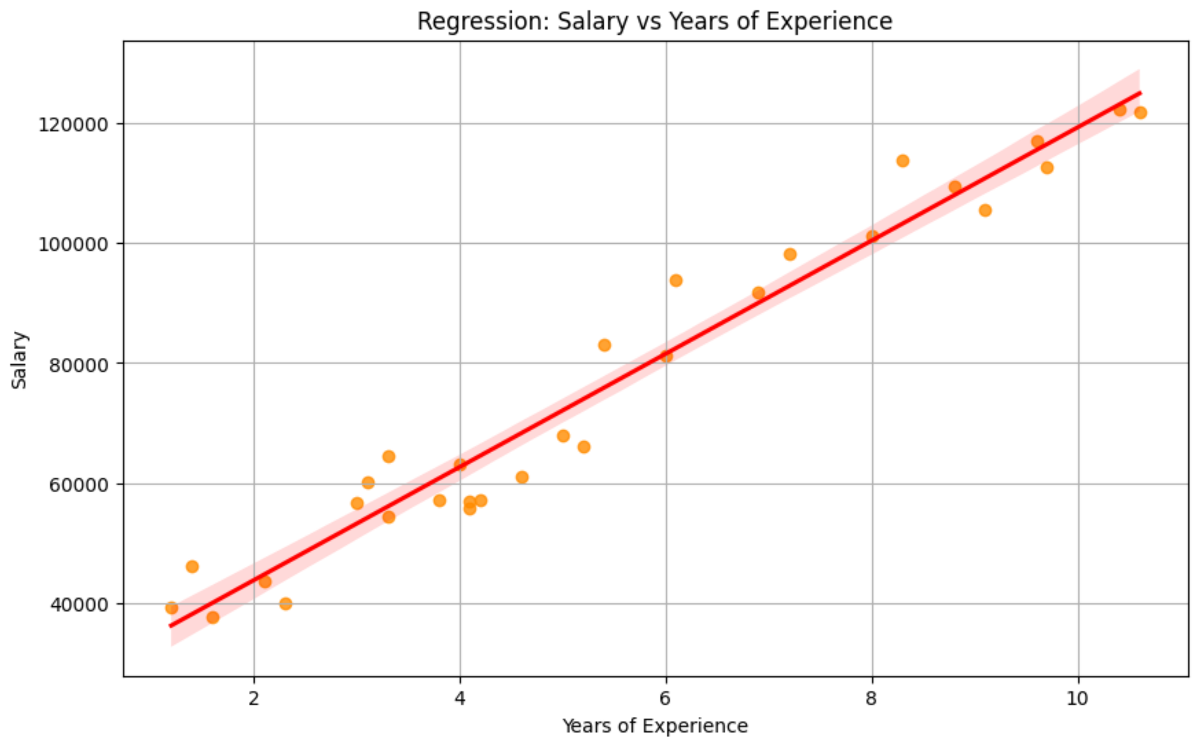
```
Mean Absolute Error (MAE): 6286.453830757743
Mean Squared Error (MSE): 49830096.855908364
Root Mean Squared Error: 7059.043621901508
```

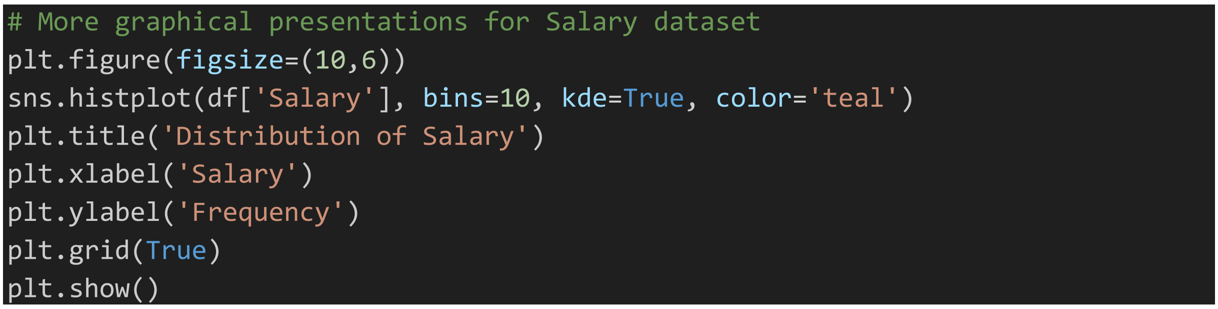
```
# Visualize Salary vs Years of Experience
plt.figure(figsize=(10,6))
sns.scatterplot(x=df['YearsExperience'], y=df['Salary'], color='royalblue',
s=80)
plt.title('Salary vs Years of Experience')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.grid(True)
plt.show()
```



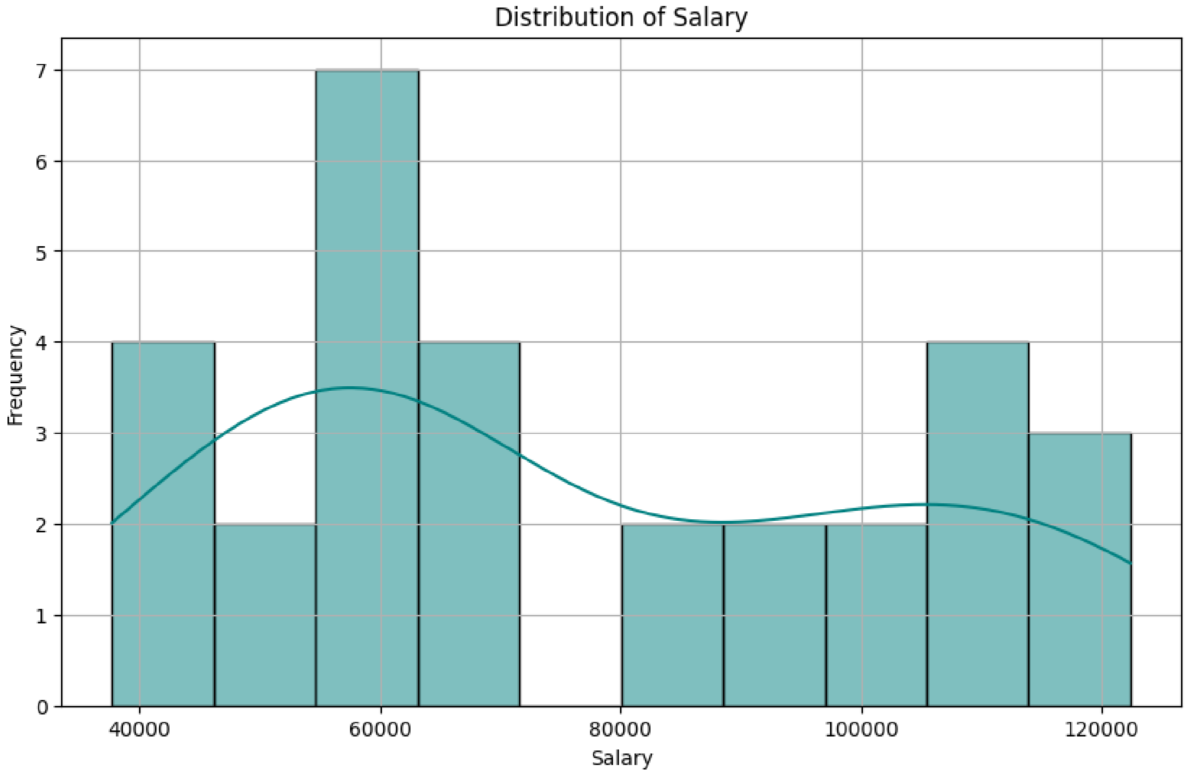


```
# Regression line plot
plt.figure(figsize=(10,6))
sns.regplot(x=df['YearsExperience'], y=df['Salary'], color='darkorange',
line_kws={'color':'red'})
plt.title('Regression: Salary vs Years of Experience')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.grid(True)
plt.show()
```

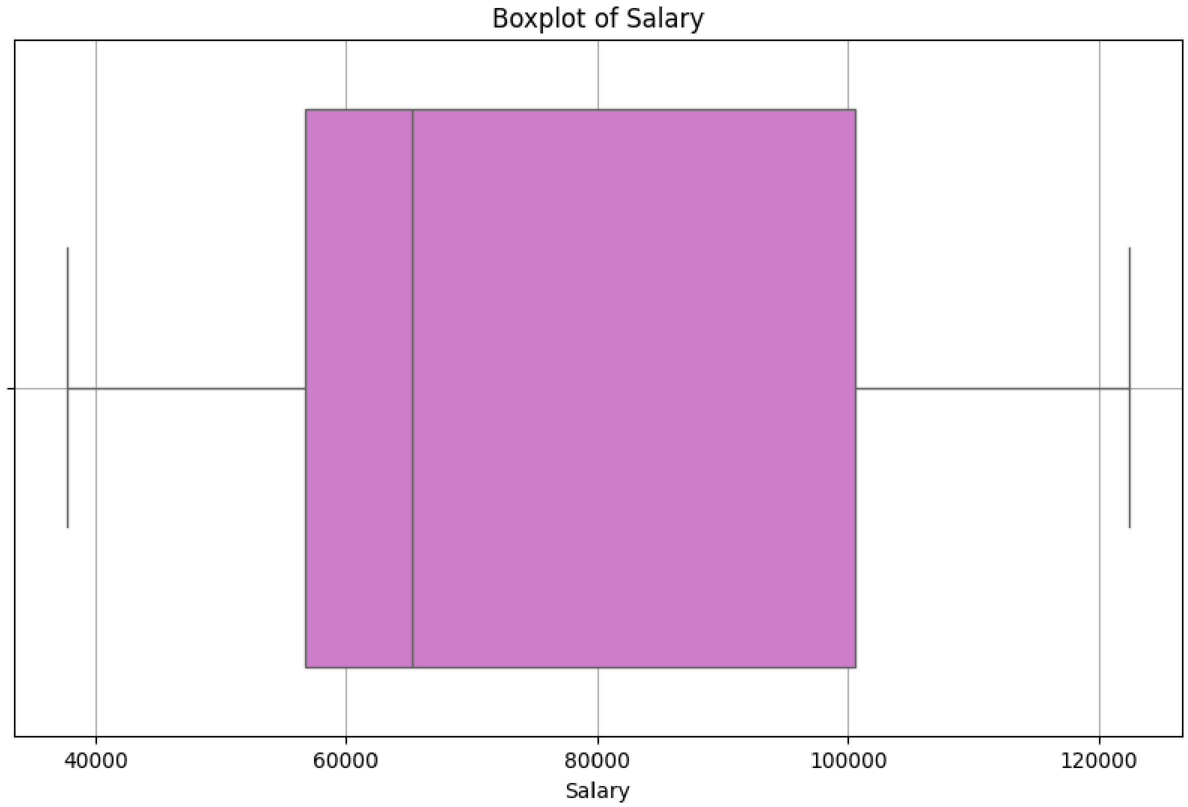




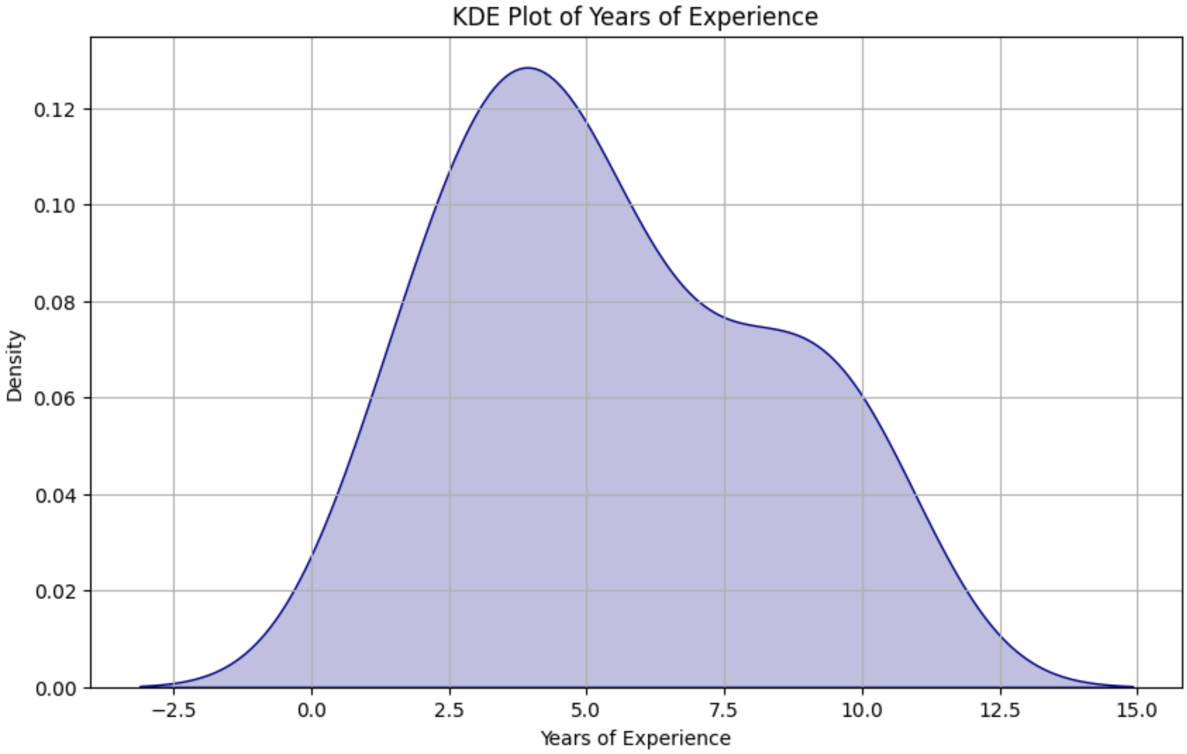
```
# More graphical presentations for Salary dataset
plt.figure(figsize=(10,6))
sns.histplot(df['Salary'], bins=10, kde=True, color='teal')
plt.title('Distribution of Salary')
plt.xlabel('Salary')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```

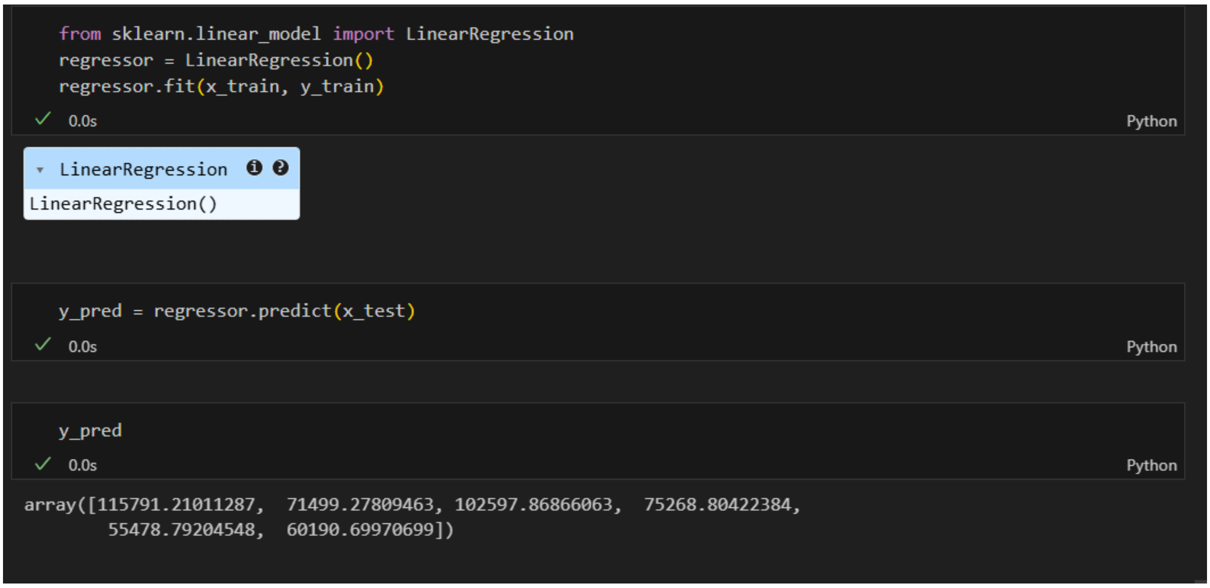


```
plt.figure(figsize=(10,6))
sns.boxplot(x=df['Salary'], color='orchid')
plt.title('Boxplot of Salary')
plt.xlabel('Salary')
plt.grid(True)
plt.show()
```



```
plt.figure(figsize=(10,6))
sns.kdeplot(df['YearsExperience'], shade=True, color='navy')
plt.title('KDE Plot of Years of Experience')
plt.xlabel('Years of Experience')
plt.ylabel('Density')
plt.grid(True)
plt.show()
```





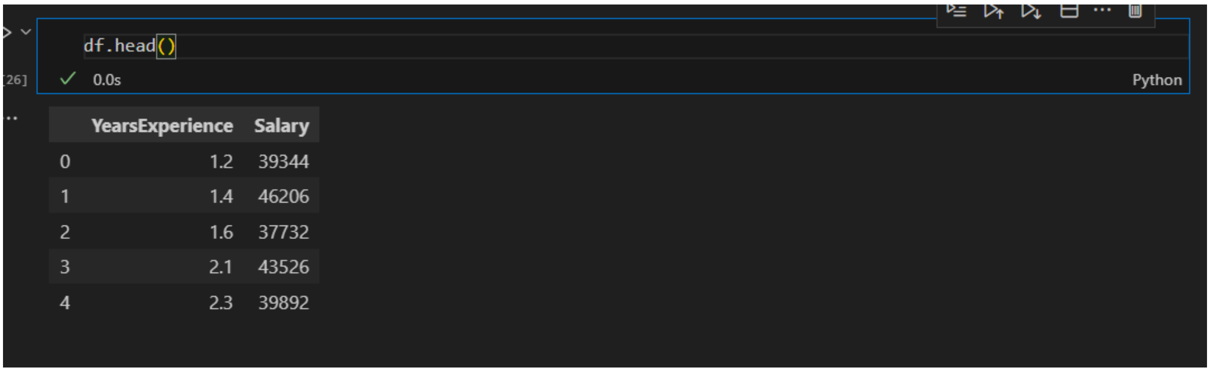
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train, y_train)
```

```
LinearRegression
LinearRegression()
```

```
y_pred = regressor.predict(x_test)
```

```
y_pred
```

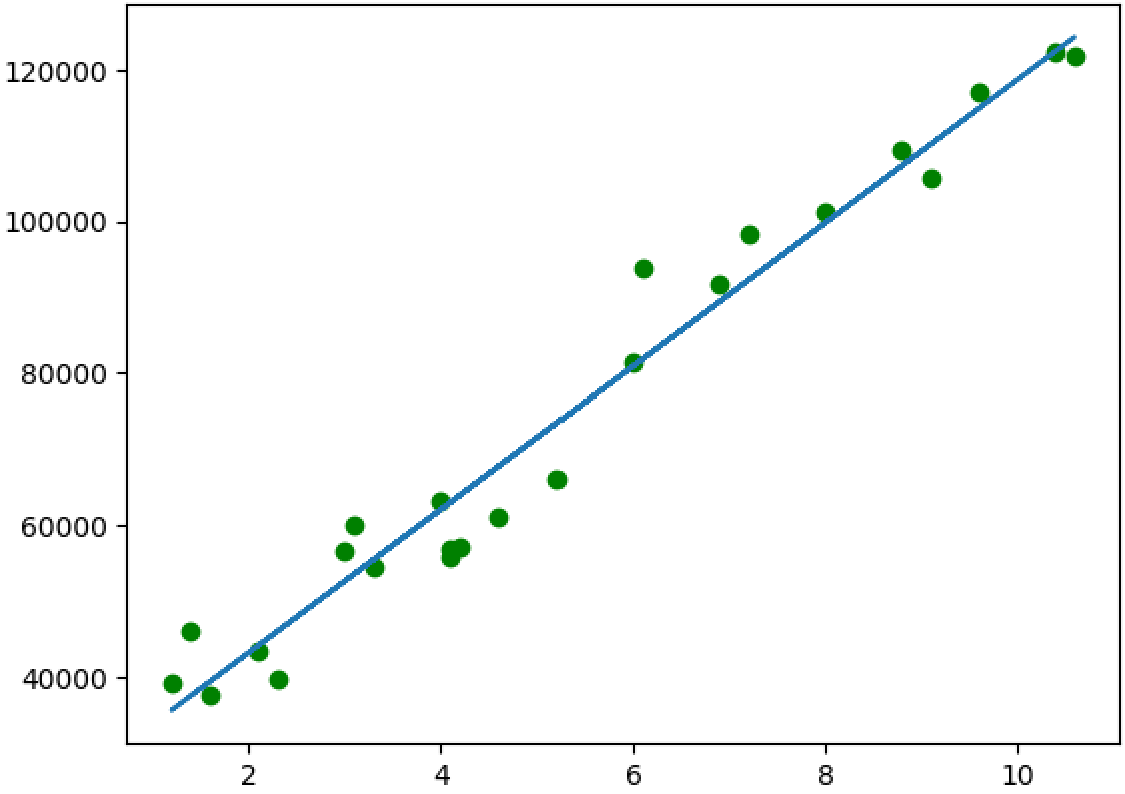
```
array([115791.21011287,  71499.27809463, 102597.86866063,  75268.80422384,
        55478.79204548,  60190.69970699])
```



```
df.head()
```

	YearsExperience	Salary
0	1.2	39344
1	1.4	46206
2	1.6	37732
3	2.1	43526
4	2.3	39892

```
plt.scatter(x_train, y_train, color='green')
plt.plot(x_train,regressor.predict(x_train))
plt.show()
```


120000
100000
80000
60000
40000
2
4
6
8
10