# Poly_1_2_Salary_1_Advertising_1

February 6, 2025

## 1 Salary

[91]:

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

[92]:

```
url = "https://raw.githubusercontent.com/content-anu/
  ↪dataset-polynomial-regression/master/Position_Salaries.csv"
df = pd.read_csv(url)
```

[93]:

```
df.head()
```

[93]:

```
             Position  Level  Salary
0    Business Analyst      1   45000
1  Junior Consultant      2   50000
2  Senior Consultant      3   60000
3             Manager      4   80000
4     Country Manager      5  110000
```

[94]:

```
# Splitting the data to X and Y
X = df.Level.values
y = df.Salary.values
print(f"The shape of X is {X.shape} and the shape of y is {y.shape}")
```

```
The shape of X is (10,) and the shape of y is (10,)
```

[95]:

```
# Reshaping the data
X = X.reshape(-1, 1)
y = y.reshape(-1, 1)
print(f"The shape of X is {X.shape} and the shape of y is {y.shape}")
```

```
The shape of X is (10, 1) and the shape of y is (10, 1)
```

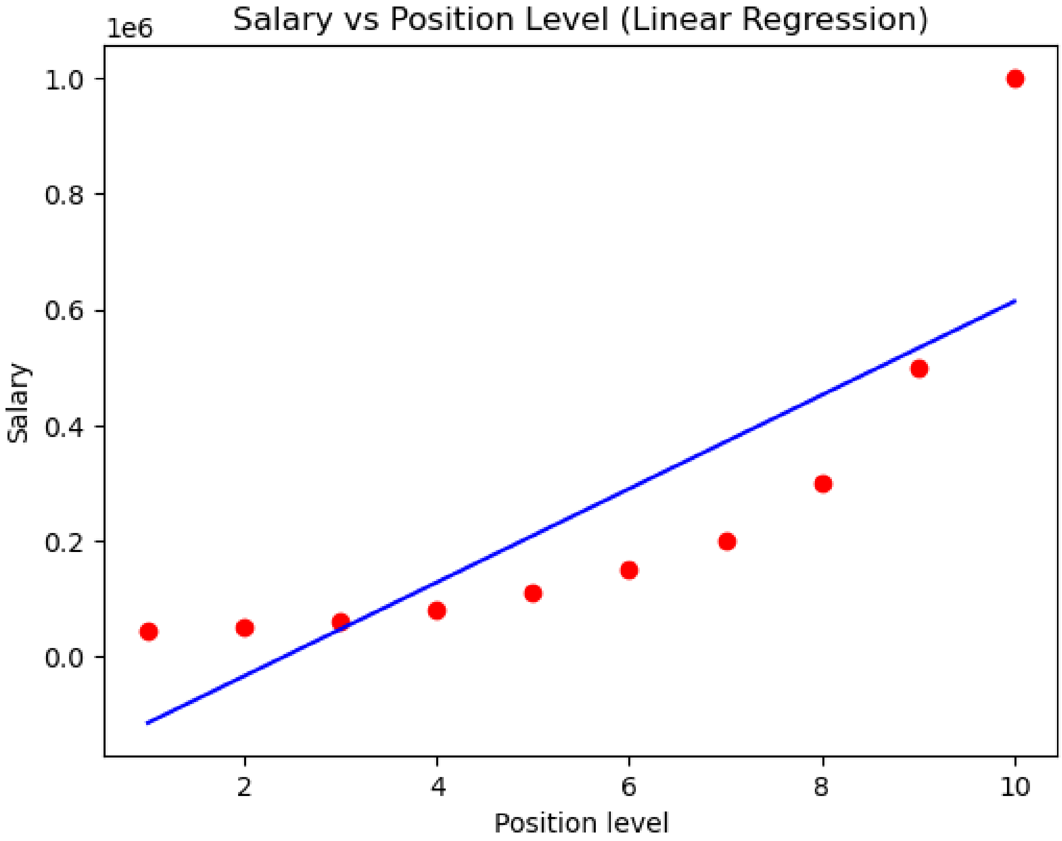
[96]:

```
# Implementing the Polynomial Regression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
```

```
# For Simple Linear Regression
lin_reg = LinearRegression()
lin_reg.fit(X, y)
y_pred = lin_reg.predict(X)
```

[97]:

```
# Linear Regression on Scatter Plot
plt.scatter(X, y, color = 'red')
plt.plot(X, y_pred, color = 'blue')
plt.title('Salary vs Position Level (Linear Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```



[98]:

```
# Evaluating the model
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y, y_pred)
r2 = r2_score(y, y_pred)
print(f"The Mean Squared Error is {mse} and the R2 Score is {r2}")
```

```
The Mean Squared Error is 26695878787.878788 and the R2 Score is
0.6690412331929895
```

[99]:

```
from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree = 3)
X_poly = poly_reg.fit_transform(X)
```

[100]:

```
X
```

[100]:

```
array([[ 1],
       [ 2],
       [ 3],
       [ 4],
       [ 5],
       [ 6],
       [ 7],
       [ 8],
       [ 9],
       [10]], dtype=int64)
```

[101]:

```
X_poly
```

[101]:

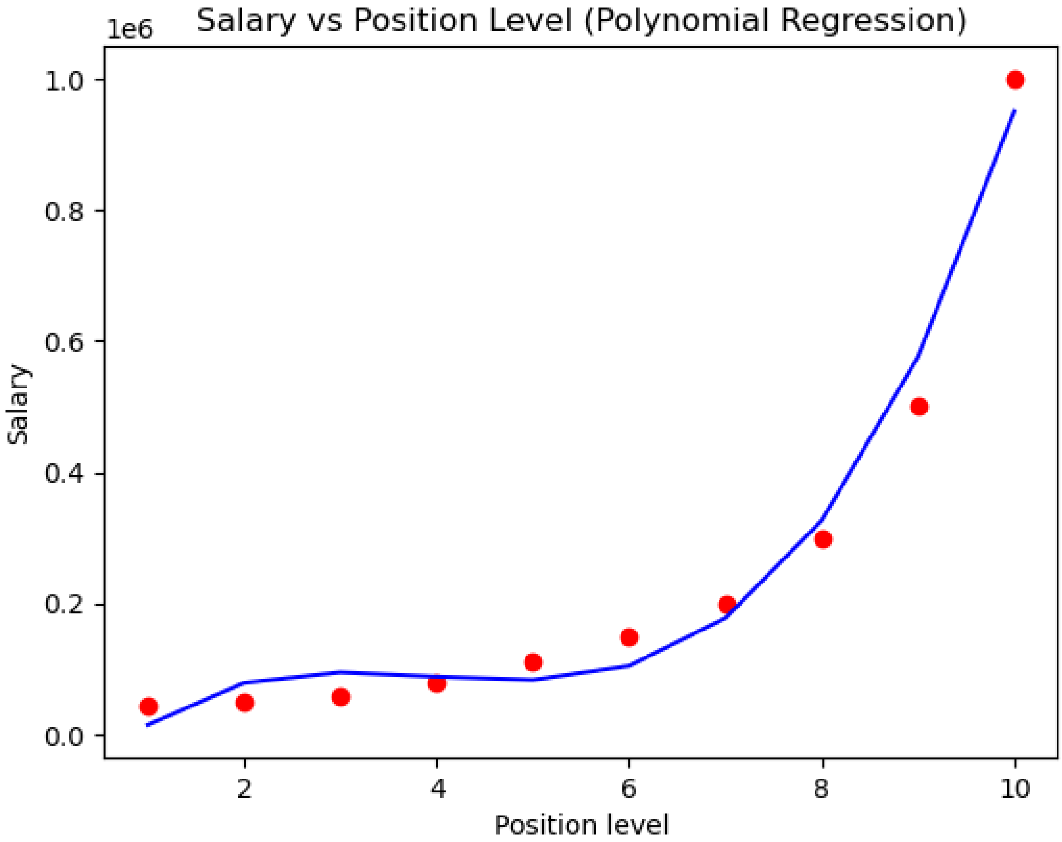
```
array([[   1.,    1.,    1.,    1.],
       [   1.,    2.,    4.,    8.],
       [   1.,    3.,    9.,   27.],
       [   1.,    4.,   16.,   64.],
       [   1.,    5.,   25.,  125.],
       [   1.,    6.,   36.,  216.],
       [   1.,    7.,   49.,  343.],
       [   1.,    8.,   64.,  512.],
       [   1.,    9.,   81.,  729.],
       [   1.,   10.,  100., 1000.]])
```

[102]:

```
lin_reg_2 = LinearRegression()
lin_reg_2.fit(X_poly, y)
y_pred_2 = lin_reg_2.predict(X_poly)
```

[103]:

```
# Poly Regression on Scatter Plot
plt.scatter(X, y, color = 'red')
plt.plot(X, y_pred_2, color = 'blue')
plt.title('Salary vs Position Level (Polynomial Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```



```
[104]: # Evaluating the model
       from sklearn.metrics import mean_squared_error, r2_score
       mse = mean_squared_error(y, y_pred_2)
       r2 = r2_score(y, y_pred_2)
       print(f"The Mean Squared Error is {mse} and the R2 Score is {r2}")
```

```
The Mean Squared Error is 1515662004.6620033 and the R2 Score is
0.9812097727913367
```

## 2 Advertising

```
[105]: # Practice on Advertisemnt Dataset
       url = r"D:\Supervised Machine Learning lab (SMLL)\2\Assignment 2 Advertising.
        ↪csv"
       dataset = pd.read_csv(url)
       dataset.head()
```

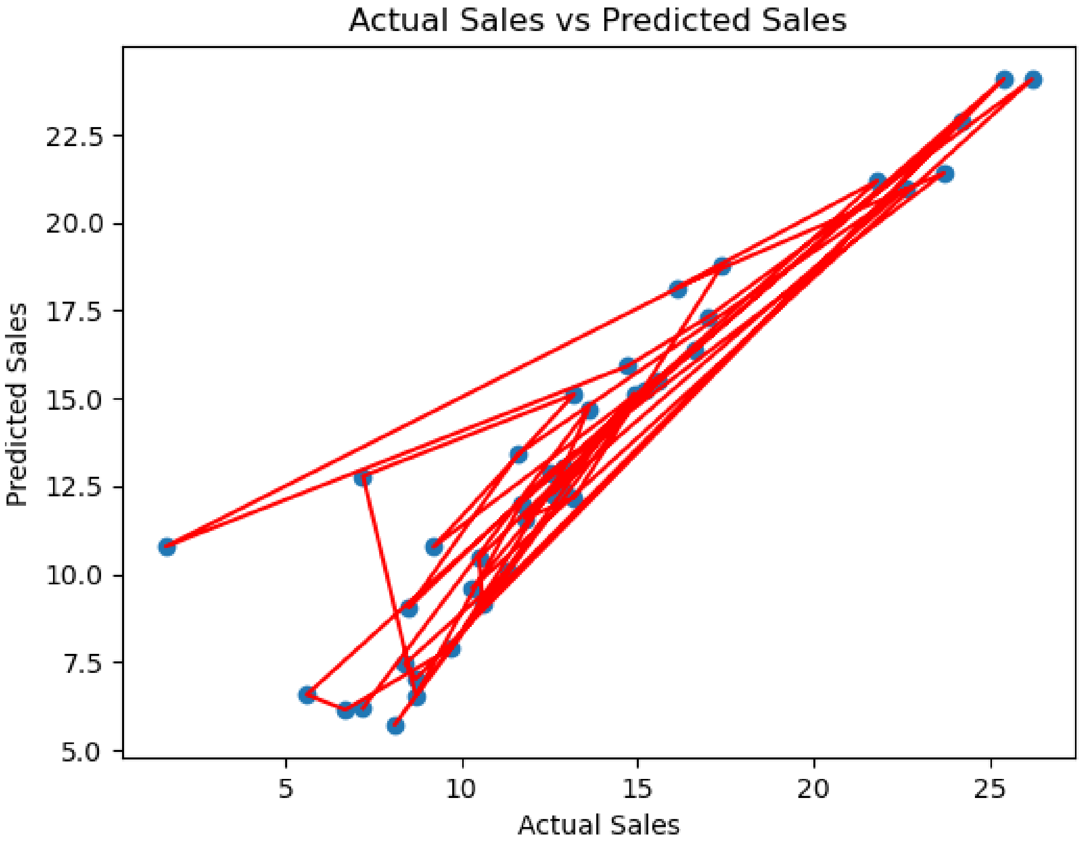
```
[105]:    Unnamed: 0     TV  Radio  Newspaper  Sales
       0           1  230.1   37.8       69.2   22.1
```

```
1            2   44.5   39.3       45.1   10.4
2            3   17.2   45.9       69.3    9.3
3            4  151.5   41.3       58.5   18.5
4            5  180.8   10.8       58.4   12.9
```

[106]:

```python
# Using Multiple Linear Regression
X = dataset[['TV', 'Radio', 'Newspaper']]
y = dataset['Sales']

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 = 0)

from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)
y_pred = lin_reg.predict(X_test)

from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"The Mean Squared Error is {mse} and the R2 Score is {r2}")

# Scatter Plot
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Sales")
plt.plot(y_test, y_pred, color = 'red')
plt.ylabel("Predicted Sales")
plt.title("Actual Sales vs Predicted Sales")
plt.show()
```

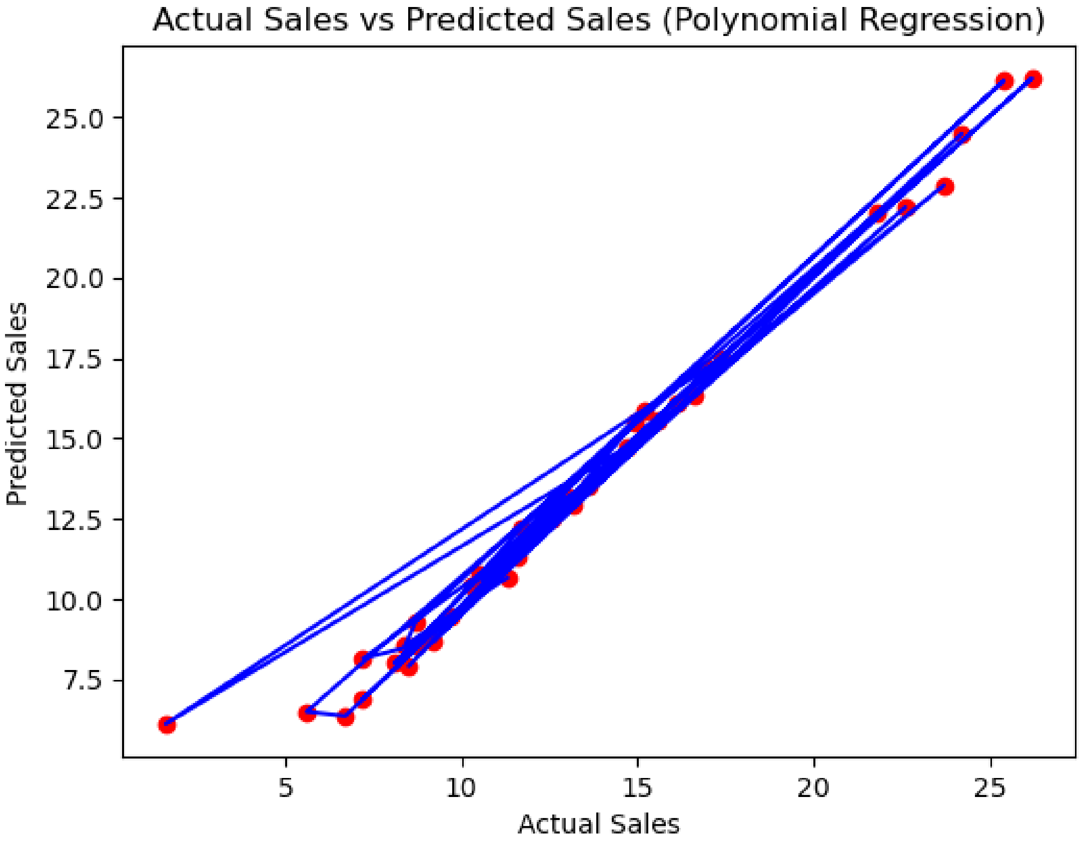
```
The Mean Squared Error is 4.402118291449685 and the R2 Score is
0.8601145185017868
```



[107]:

```
# Using Polynomial Regression
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(degree = 3)
X_train_poly = poly.fit_transform(X_train)
X_test_poly = poly.fit_transform(X_test)
y_train = np.array(y_train).reshape(-1, 1)

lin_reg_2 = LinearRegression()
lin_reg_2.fit(X_train_poly, y_train)
y_pred_2 = lin_reg_2.predict(X_test_poly)

# Poly Regression on Scatter Plot
plt.scatter(y_test, y_pred_2, color = 'red')
plt.plot(y_test, y_pred_2, color = 'blue')
plt.title('Actual Sales vs Predicted Sales (Polynomial Regression)')
plt.xlabel('Actual Sales')
plt.ylabel('Predicted Sales')
plt.show()

# Evaluating the model
```

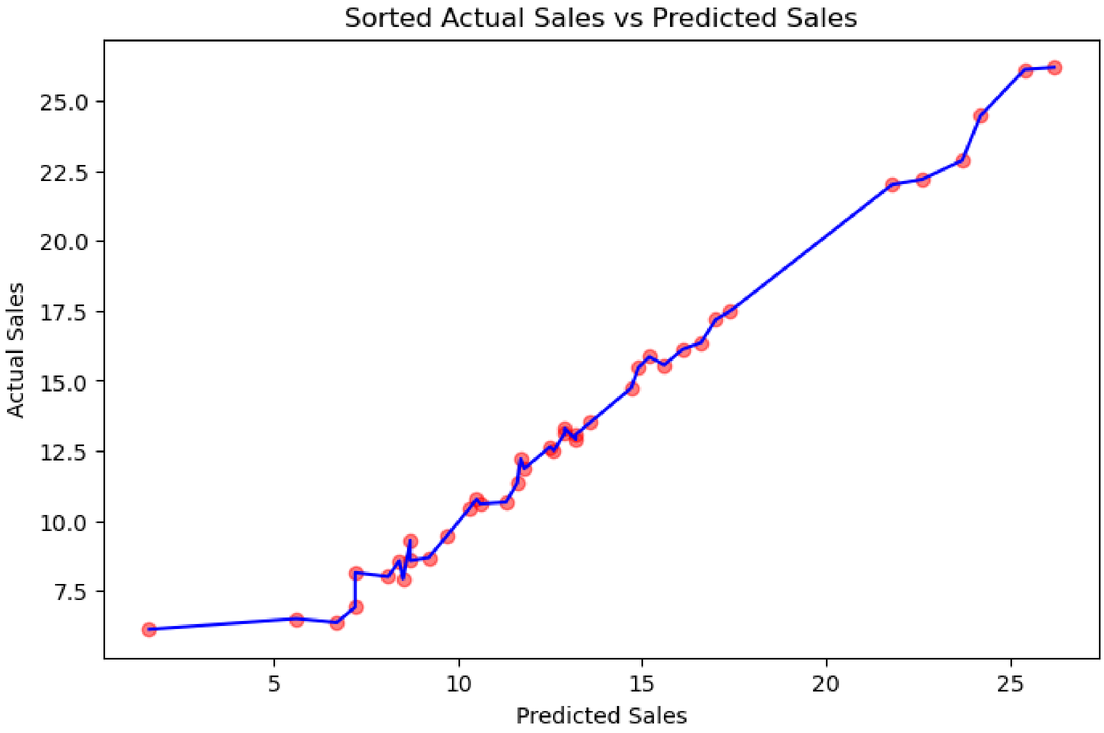
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred_2)
r2 = r2_score(y_test, y_pred_2)
print(f"The Mean Squared Error is {mse} and the R2 Score is {r2}")
```



```
The Mean Squared Error is 0.6721344417962923 and the R2 Score is
0.978641680255429
```

[109]:

```
import numpy as np
import matplotlib.pyplot as plt
sorted_idx = np.argsort(y_test)
y_test_sorted = np.array(y_test)[sorted_idx]
y_pred_sorted = y_pred_2[sorted_idx]

# Scatter Plot (Actual vs Predicted)
plt.figure(figsize=(8, 5))
plt.scatter(y_test_sorted, y_pred_sorted, color = 'red', alpha=0.5)
plt.plot(y_test_sorted, y_pred_sorted, color = 'blue')
plt.title('Sorted Actual Sales vs Predicted Sales')
plt.xlabel('Predicted Sales')
plt.ylabel('Actual Sales')
plt.show()
```


Sorted Actual Sales vs Predicted Sales
Actual Sales
Predicted Sales