10. Data Science – Machine Learning – Polynomial Features

Contents

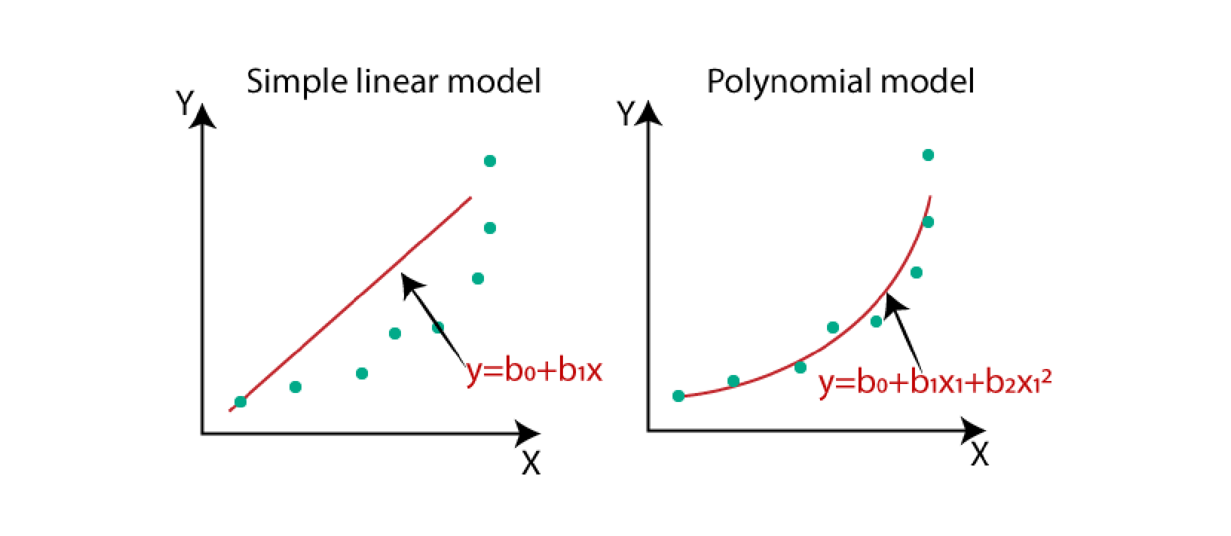
10. Data Science – Machine Learning – Polynomial Features

1. Polynomial Features for machine learning

- ✓ As such, polynomial features are a type of feature engineering means creating new input features based on the existing features.
- ✓ For example, if a dataset had one input feature X, then a polynomial feature would be the addition of a new feature (column) where values were calculated by squaring the values in X, e.g. X^2.
- ✓ This process can be repeated for each input variable in the dataset, creating a transformed version of each.
- ✓ The "degree" of the polynomial is used to control the number of features added.

2. Need of Polynomial Features

- ✓ If we apply a linear model on a linear dataset, then it provides us a good result as we have seen in Simple Linear Regression.
- ✓ If we apply the same model without any modification on a non-linear dataset, then it will produce wrong results
- ✓ Due to this,
  - o The error rate will be high etc



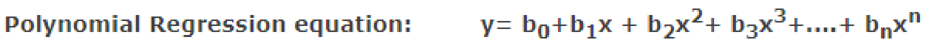
3. Equations

Simple Linear Regression equation

- ✓ y = b0+b1x

Multiple Linear Regression equation

- ✓ y= b0+b1x+ b2x2+ b3x3+....+ bnxn



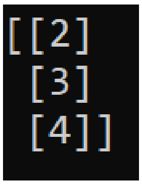
Polynomial Regression equation: $y= b_0+b_1x + b_2x^2+ b_3x^3+....+ b_nx^n$

Program Name

Creating an array
demo1.py

```
from numpy import asarray

data = asarray([[2], [3], [4]])
print(data)
```

Output

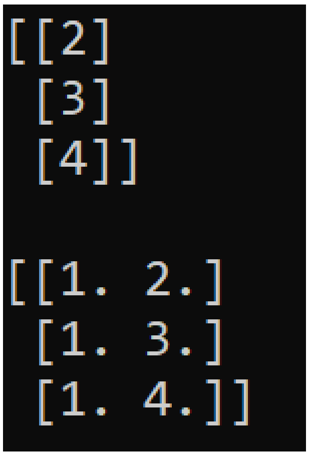


```
[[2]
 [3]
 [4]]
```

Program Name   Creating feature from existing feature
demo2.py

```
from numpy import asarray
from sklearn.preprocessing import PolynomialFeatures

data1 = asarray([[2],[3],[4]])

trans = PolynomialFeatures(degree = 1)
data2 = trans.fit_transform(data1)

print(data1)
print()
print(data2)
```

Output



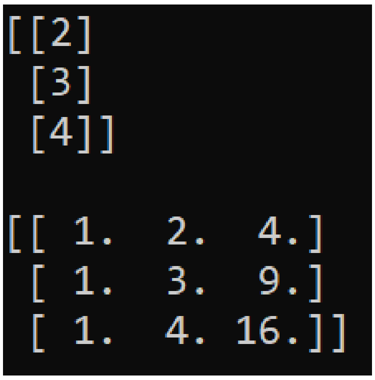
```
[[2]
 [3]
 [4]]

[[1. 2.]
 [1. 3.]
 [1. 4.]]
```

Program Name

Creating feature from existing feature
demo3.py

```
from numpy import asarray
from sklearn.preprocessing import PolynomialFeatures

data1 = asarray([[2],[3],[4]])

trans = PolynomialFeatures(degree = 2)
data2 = trans.fit_transform(data1)

print(data1)
print()
print(data2)
```

Output



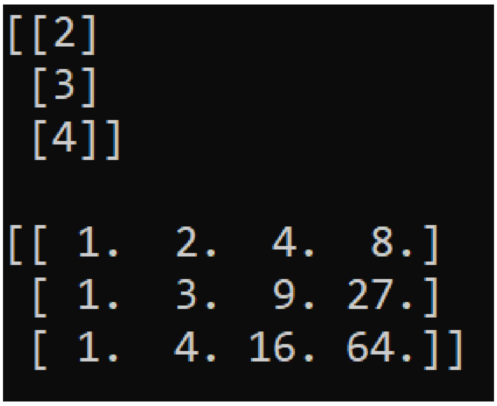
```
[[2]
 [3]
 [4]]

[[ 1.  2.  4.]
 [ 1.  3.  9.]
 [ 1.  4. 16.]]
```

Program Name

Creating feature from existing feature
demo4.py

```
from numpy import asarray
from sklearn.preprocessing import PolynomialFeatures

data1 = asarray([[2],[3],[4]])

trans = PolynomialFeatures(degree = 3)
data2 = trans.fit_transform(data1)

print(data1)
print()
print(data2)
```

Output



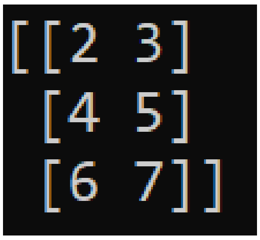
```
[[2]
 [3]
 [4]]

[[ 1.  2.  4.  8.]
 [ 1.  3.  9. 27.]
 [ 1.  4. 16. 64.]]
```

Program Name

Creating feature from existing feature
demo5.py

```
from numpy import asarray

data1 = asarray([[2, 3],[4, 5],[6, 7]])

print(data1)
```

Output



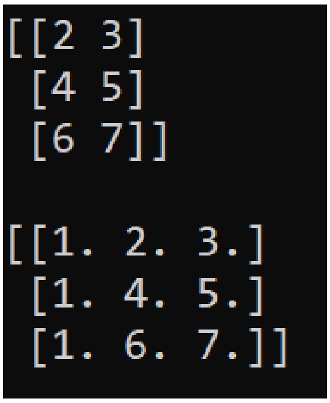
```
[[2 3]
 [4 5]
 [6 7]]
```

Program Name

Creating feature from existing feature
demo6.py

```
from numpy import asarray
from sklearn.preprocessing import PolynomialFeatures

data1 = asarray([[2, 3],[4, 5],[6, 7]])


trans = PolynomialFeatures(degree = 1)
data2 = trans.fit_transform(data1)

print(data1)
print()
print(data2)
```

Output



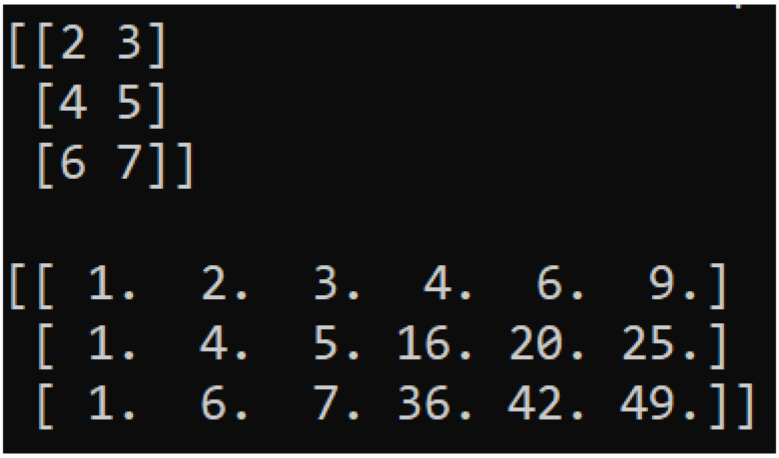
```
[[2 3]
 [4 5]
 [6 7]]

[[1. 2. 3.]
 [1. 4. 5.]
 [1. 6. 7.]]
```

Program Name

Creating feature from existing feature
demo7.py

```
from numpy import asarray
from sklearn.preprocessing import PolynomialFeatures

data1 = asarray([[2, 3],[4, 5],[6, 7]])


trans = PolynomialFeatures(degree = 2)
data2 = trans.fit_transform(data1)

print(data1)
print()
print(data2)
```

Output



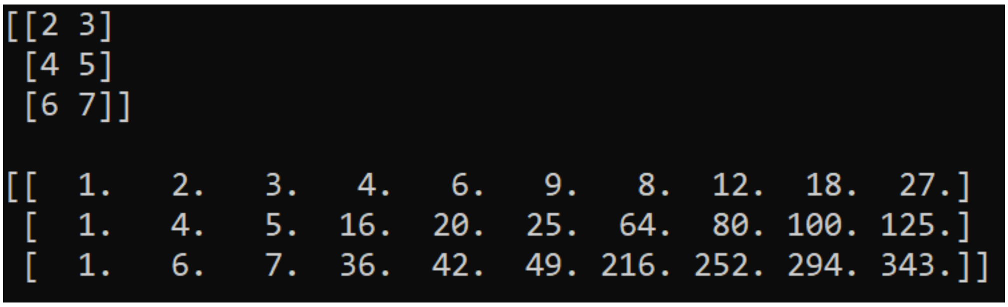
```
[[2 3]
 [4 5]
 [6 7]]

[[ 1.  2.  3.  4.  6.  9.]
 [ 1.  4.  5. 16. 20. 25.]
 [ 1.  6.  7. 36. 42. 49.]]
```

Program Name

Creating feature from existing feature
demo8.py

```
from numpy import asarray
from sklearn.preprocessing import PolynomialFeatures

data1 = asarray([[2, 3],[4, 5],[6, 7]])


trans = PolynomialFeatures(degree = 3)
data2 = trans.fit_transform(data1)

print(data1)
print()
print(data2)
```

Output



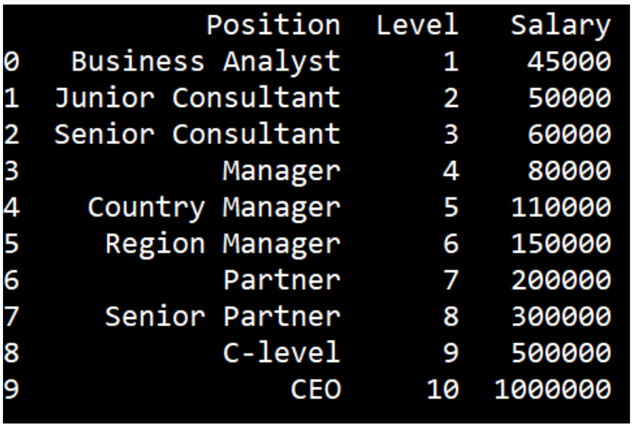
```
[[2 3]
 [4 5]
 [6 7]]

[[  1.   2.   3.   4.   6.   9.   8.  12.  18.  27.]
 [  1.   4.   5.  16.  20.  25.  64.  80. 100. 125.]
 [  1.   6.   7.  36.  42.  49. 216. 252. 294. 343.]]
```

Program Name

Loading dataset
demo9.py

```
import pandas as pd

df = pd.read_csv("poly_dataset.csv")

print(df)
```

Output



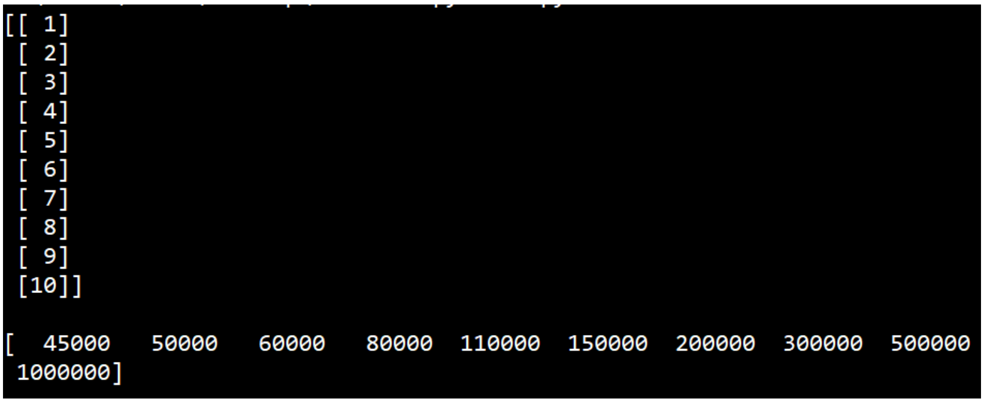
```
            Position  Level   Salary
0   Business Analyst      1    45000
1  Junior Consultant      2    50000
2  Senior Consultant      3    60000
3            Manager      4    80000
4    Country Manager      5   110000
5     Region Manager      6   150000
6            Partner      7   200000
7     Senior Partner      8   300000
8            C-level      9   500000
9                CEO     10  1000000
```

Program Name

Data preparation
demo10.py

```
import pandas as pd

df = pd.read_csv("poly_dataset.csv")

# Data preparation
X = df.iloc[:, 1:2].values
y = df.iloc[:, 2].values

print(X)
print()
print(y)
```

Output



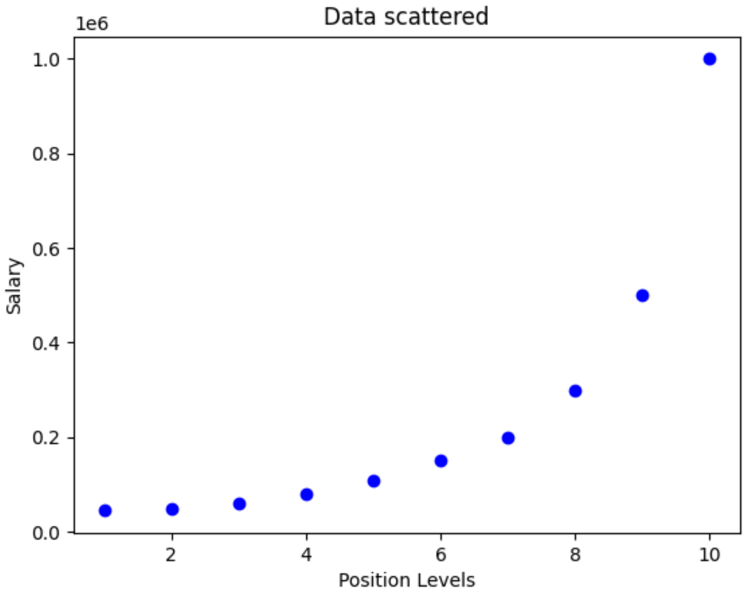
```
[[ 1]
 [ 2]
 [ 3]
 [ 4]
 [ 5]
 [ 6]
 [ 7]
 [ 8]
 [ 9]
 [10]]

[  45000   50000   60000   80000  110000  150000  200000  300000  500000
 1000000]
```

Program Name

Plotting the dataset
demo11.py

```
import pandas as pd
import matplotlib.pyplot as plt

df=pd.read_csv("poly_dataset.csv")

X = df.iloc[:, 1:2].values
y = df.iloc[:, 2].values

plt.scatter(X, y, color="blue")

plt.title("Data scattered")
plt.xlabel("Position Levels")
plt.ylabel("Salary")

plt.show()
```

Output



Program Name

Model training
demo12.py

```
import pandas as pd
from sklearn.linear_model import LinearRegression

# Loading the dataset
df = pd.read_csv("poly_dataset.csv")

# Data preparation
X = df.iloc[:, 1:2].values
y = df.iloc[:, 2].values

# Model training
lin_regs= LinearRegression()
lin_regs.fit(X, y)

print("Model got trained")
```

Output

```
Model got trained
```

Program Name

Model prediction
demo13.py

```
import pandas as pd
from sklearn.linear_model import LinearRegression

# Loading the dataset
df = pd.read_csv("poly_dataset.csv")

# Data preparation
X = df.iloc[:, 1:2].values
y = df.iloc[:, 2].values

# Model training
lin_regs= LinearRegression()
lin_regs.fit(X, y)

print("Model got trained")
print(lin_regs.predict([[6.5]])  )
```

Output

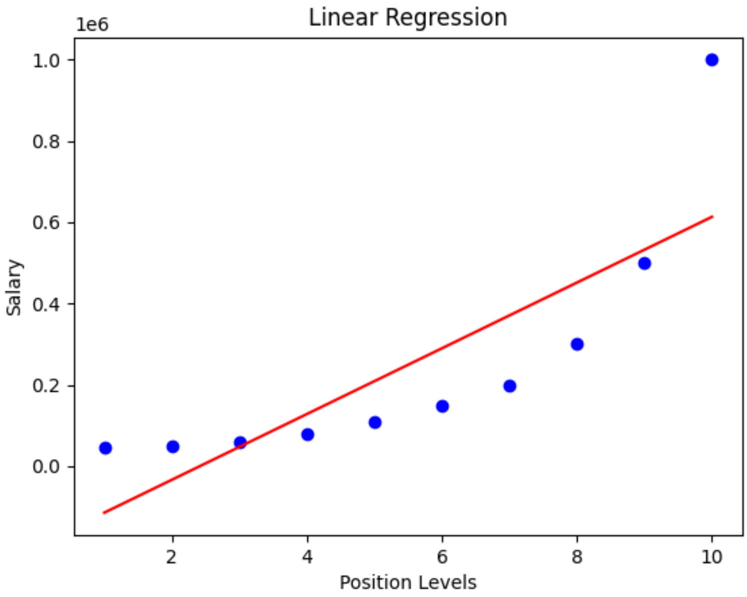
```
[330378.78787879]
```

Program Name

Plotting the dataset
demo14.py

```
import pandas as pd
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

# Loading the dataset
df=pd.read_csv("poly_dataset.csv")

# Data preparation
X = df.iloc[:, 1:2].values
y = df.iloc[:, 2].values

# Model training
lin_regs= LinearRegression()
lin_regs.fit(X, y)

plt.scatter(X, y, color="blue")

plt.plot(X, lin_regs.predict(X), color = "red")

plt.title("Linear Regression")
plt.xlabel("Position Levels")
plt.ylabel("Salary")

plt.show()
```

Output


Linear Regression
1e6
1.0
0.8
0.6
0.4
0.2
0.0
Salary
2
4
6
8
10
Position Levels

4. Lets create Polynomial features

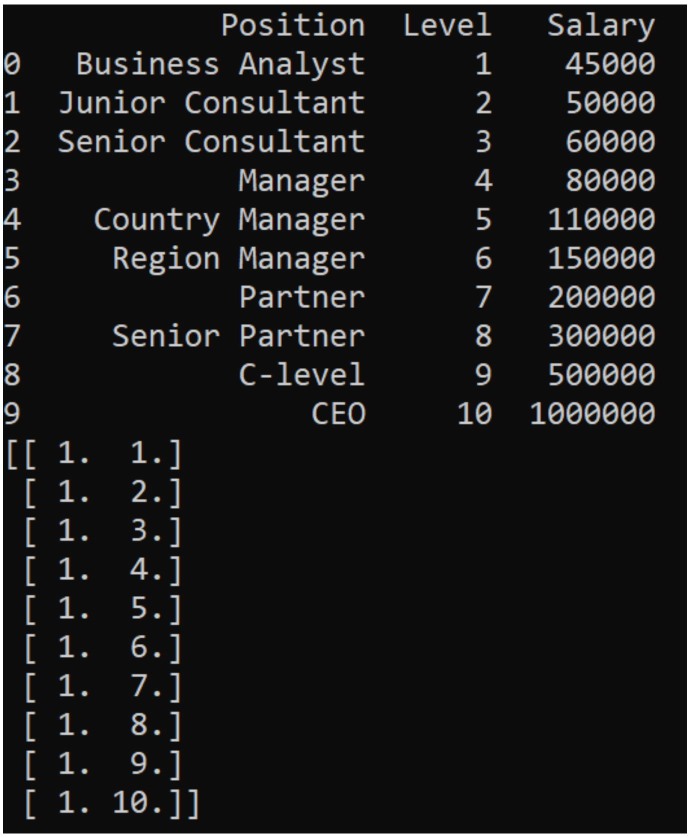
- ✓ We need to use PolynomialFeatures class to get polynomial features.

Program Name

Fitting the Polynomial regression to the dataset
demo15.py

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures


# Loading the dataset
df=pd.read_csv("poly_dataset.csv")

# Data preparation
X = df.iloc[:, 1:2].values
y = df.iloc[:, 2].values


poly_regs= PolynomialFeatures(degree = 1)

x_poly= poly_regs.fit_transform(X)

print(x_poly)
```

Output



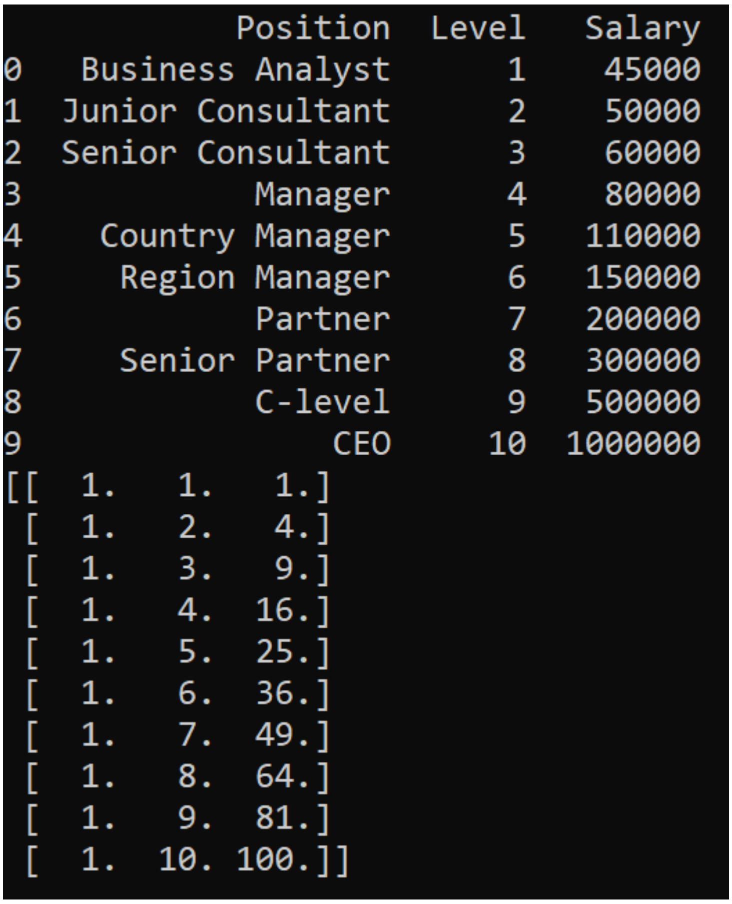
```
             Position  Level   Salary
0   Business Analyst      1    45000
1  Junior Consultant      2    50000
2  Senior Consultant      3    60000
3            Manager      4    80000
4    Country Manager      5   110000
5     Region Manager      6   150000
6            Partner      7   200000
7     Senior Partner      8   300000
8            C-level      9   500000
9                CEO     10  1000000
[[ 1.  1.]
 [ 1.  2.]
 [ 1.  3.]
 [ 1.  4.]
 [ 1.  5.]
 [ 1.  6.]
 [ 1.  7.]
 [ 1.  8.]
 [ 1.  9.]
 [ 1. 10.]]
```

Program Name

Fitting the Polynomial regression to the dataset
demo16.py

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures


# Loading the dataset
df=pd.read_csv("poly_dataset.csv")

# Data preparation
X = df.iloc[:, 1:2].values
y = df.iloc[:, 2].values


poly_regs= PolynomialFeatures(degree = 2)

x_poly= poly_regs.fit_transform(X)

print(x_poly)
```

Output



```
             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
5     Region Manager      6   150000
6            Partner      7   200000
7     Senior Partner      8   300000
8            C-level      9   500000
9                CEO     10  1000000
[[  1.    1.    1.]
 [  1.    2.    4.]
 [  1.    3.    9.]
 [  1.    4.   16.]
 [  1.    5.   25.]
 [  1.    6.   36.]
 [  1.    7.   49.]
 [  1.    8.   64.]
 [  1.    9.   81.]
 [  1.   10.  100.]]
```

Program Name

Fitting the Polynomial regression to the dataset
demo17.py

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures

# Loading the dataset
df=pd.read_csv("poly_dataset.csv")

# Data preparation
X = df.iloc[:, 1:2].values
y = df.iloc[:, 2].values


poly_regs= PolynomialFeatures(degree = 2)
x_poly= poly_regs.fit_transform(X)

model = LinearRegression()
model.fit(x_poly, y)

print("Fitting the Polynomial regression to the dataset ")
```

Output

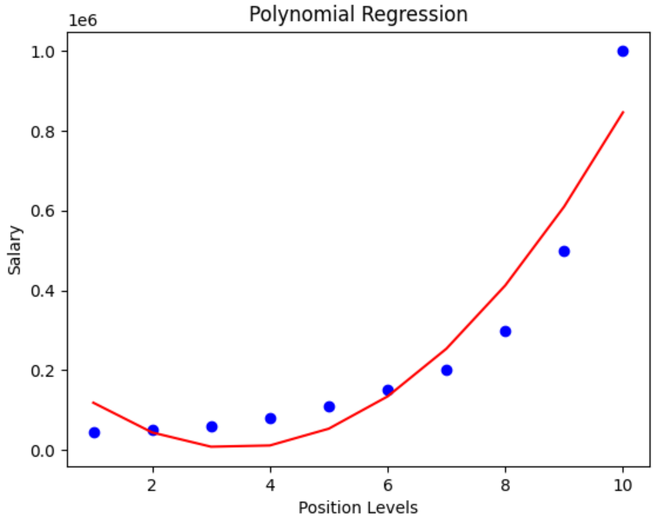
```
Fitting the Polynomial regression to the dataset
```

Program Name

Plotting Polynomial Regression features
demo18.py

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
import matplotlib.pyplot as plt

# Loading the dataset
df=pd.read_csv("poly_dataset.csv")

# Data preparation
X = df.iloc[:, 1:2].values
y = df.iloc[:, 2].values

#Fitting the Polynomial regression to the dataset
poly_regs= PolynomialFeatures(degree = 2)
x_poly = poly_regs.fit_transform(X)
model =LinearRegression()
model.fit(x_poly, y)

# Plotting Polynomial Regression

plt.scatter(X, y, color="blue")

plt.plot(X, model.predict(x_poly), color="red")

plt.title("Polynomial Regression")
plt.xlabel("Position Levels")
plt.ylabel("Salary")

plt.show()
```

Output

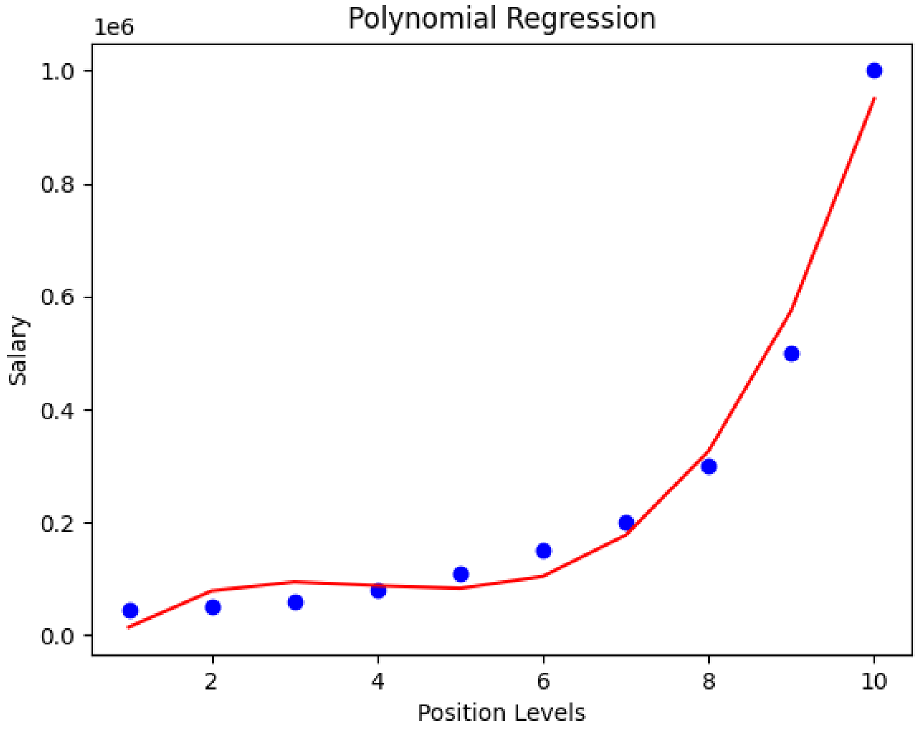

Polynomial Regression
1e6
1.0
0.8
0.6
0.4
0.2
0.0
Salary
2
4
6
8
10
Position Levels

Program Name

Plotting Polynomial Regression features
demo19.py

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
import matplotlib.pyplot as plt

# Loading the dataset
df=pd.read_csv("poly_dataset.csv")

# Data preparation
X = df.iloc[:, 1:2].values
y = df.iloc[:, 2].values

#Fitting the Polynomial regression to the dataset
poly_regs= PolynomialFeatures(degree = 3)
x_poly = poly_regs.fit_transform(X)
model =LinearRegression()
model.fit(x_poly, y)

# Plotting Polynomial Regression

plt.scatter(X, y, color="blue")

plt.plot(X, model.predict(x_poly), color="red")

plt.title("Polynomial Regression")
plt.xlabel("Position Levels")
plt.ylabel("Salary")

plt.show()
```

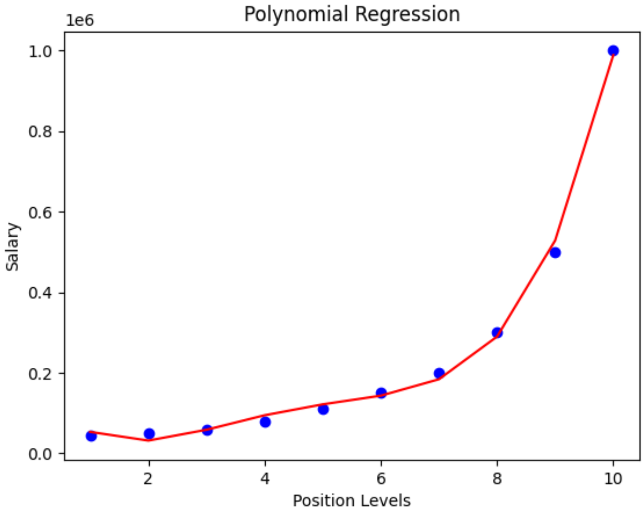
Output


Polynomial Regression
1e6
1.0
0.8
0.6
0.4
0.2
0.0
Salary
2
4
6
8
10
Position Levels

Program Name	Plotting Polynomial Regression features demo20.py

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
import matplotlib.pyplot as plt

# Loading the dataset
df=pd.read_csv("poly_dataset.csv")

# Data preparation
X = df.iloc[:, 1:2].values
y = df.iloc[:, 2].values

#Fitting the Polynomial regression to the dataset
poly_regs= PolynomialFeatures(degree = 4)
x_poly = poly_regs.fit_transform(X)
model =LinearRegression()
model.fit(x_poly, y)

# Plotting Polynomial Regression

plt.scatter(X, y, color="blue")

plt.plot(X, model.predict(x_poly), color="red")

plt.title("Polynomial Regression")
plt.xlabel("Position Levels")
plt.ylabel("Salary")

plt.show()
```

Output


Polynomial Regression
1e6
1.0
0.8
0.6
0.4
0.2
0.0
Salary
2
4
6
8
10
Position Levels

Program Name

Predicting result
demo21.py

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
import matplotlib.pyplot as plt

# Loading the dataset
df=pd.read_csv("poly_dataset.csv")

# Data preparation
X = df.iloc[:, 1:2].values
y = df.iloc[:, 2].values

#Fitting the Polynomial regression to the dataset
poly_regs= PolynomialFeatures(degree = 4)
x_poly= poly_regs.fit_transform(X)
model =LinearRegression()
model.fit(x_poly, y)

# Prediction with Polynomial Regression
poly_pred = model.predict(poly_regs.fit_transform([[6.5]]))
print(poly_pred)
```

Output

```
[158862.45265155]
```

Note

- ✓ LinearRegression predicted output is : [330378.78787879]
- ✓ Polynomial Regression predicted output is : [158862.45265155]

Conclusion

- ✓ Polynomial Regression predicted output is the accurate one according to the discussion