Curso Data Science

Skill Lab

Prof Me Eng Marcelo Bianchi
Data Scientist | Data Engineer | Data Architect

Aula 8 – Polynomial Regression

Machine Learning

Polynomial Regression

Regressions

Simple Linear Regression

$$y = b_0 + b_1 x_1$$

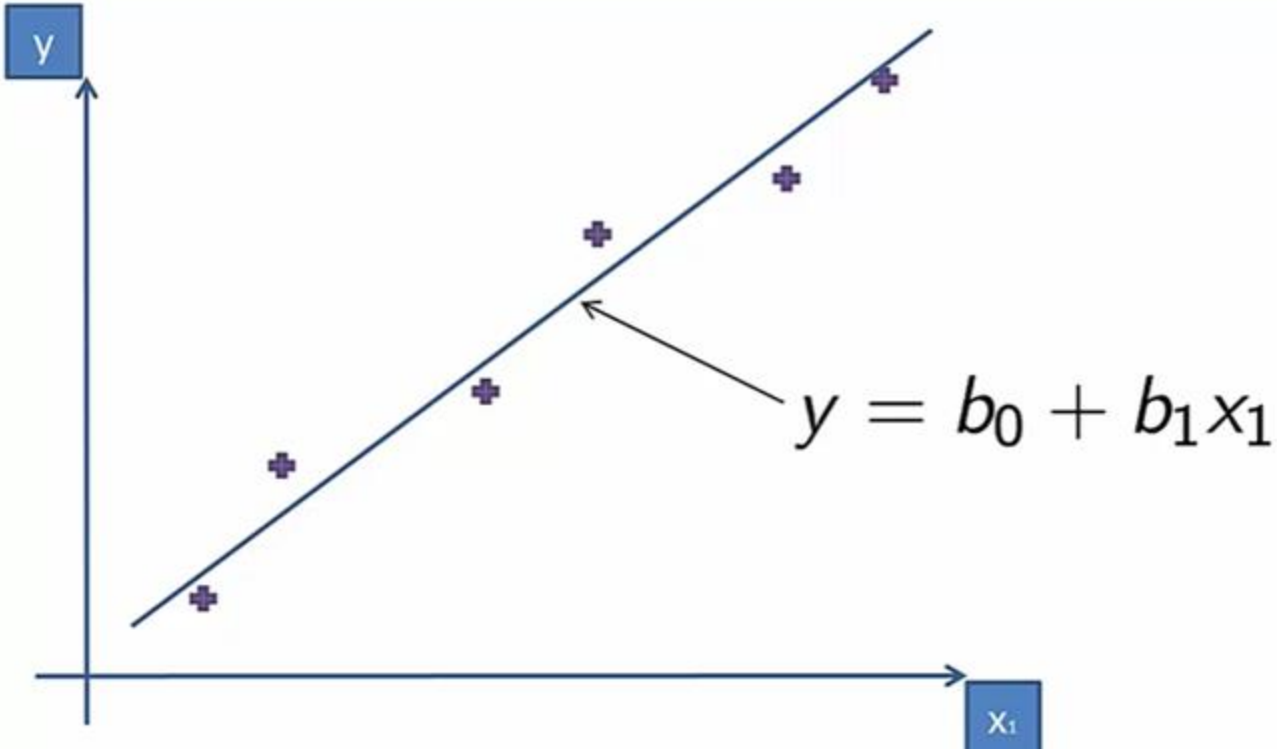
Multiple Linear Regression

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$$

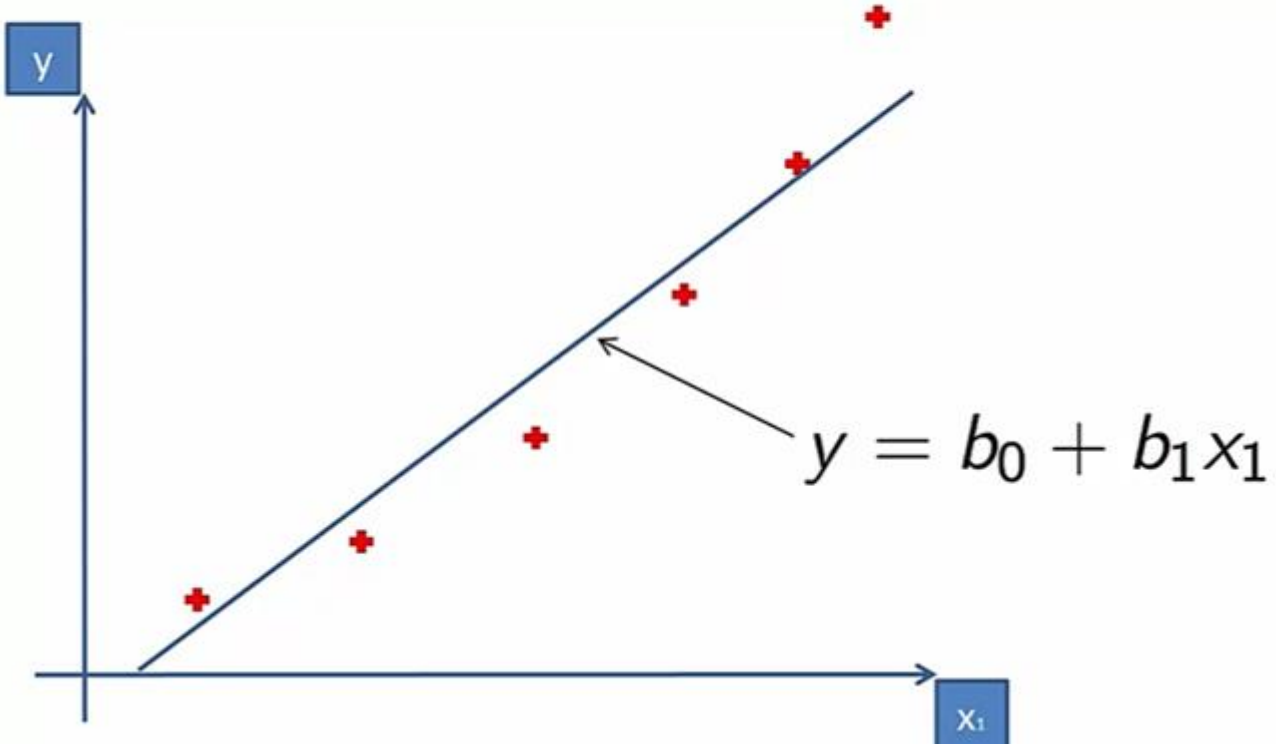
Polynomial Linear Regression

$$y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$$

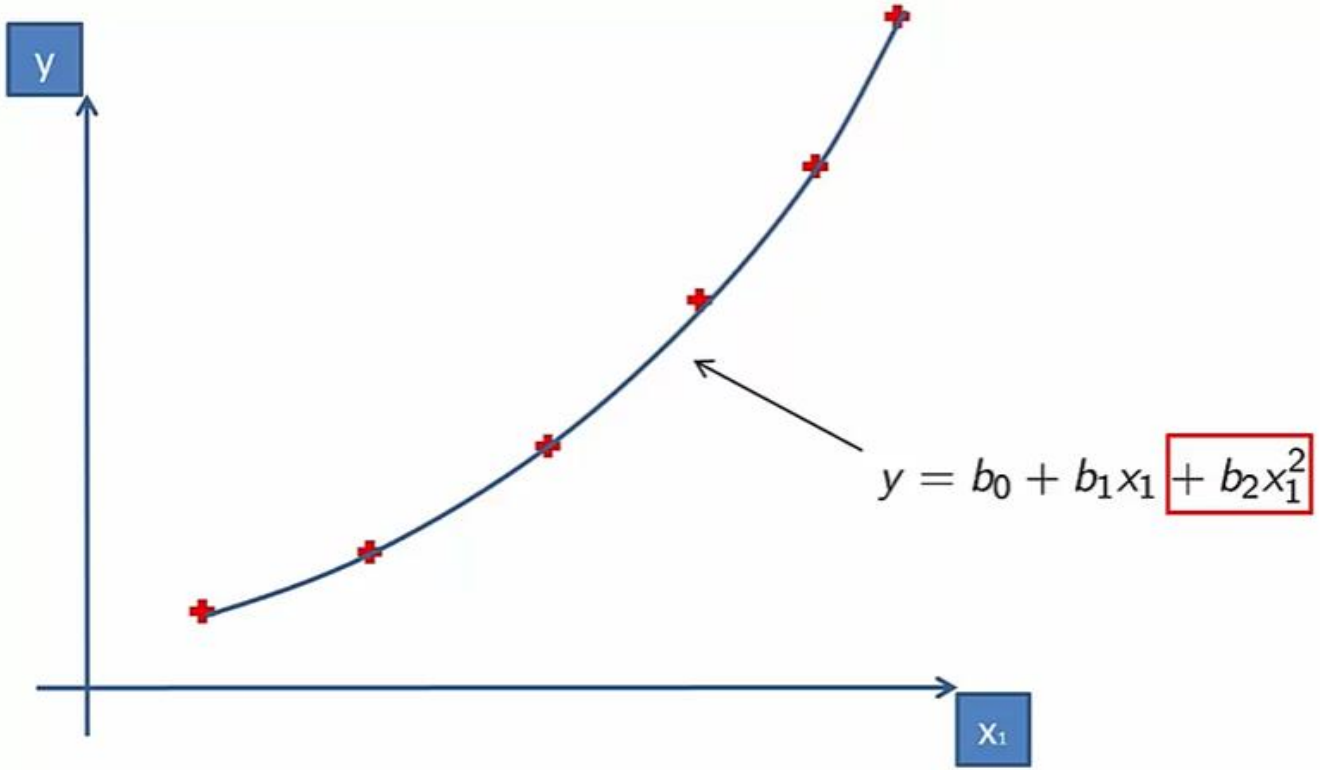
Simple Linear Regression



Simple Linear Regression



Polinomial Regression



Polinomial Regression

One Question: Why “Linear”?

Polinomial Regression

Polynomial Linear Regression

$$y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$$

DataSet

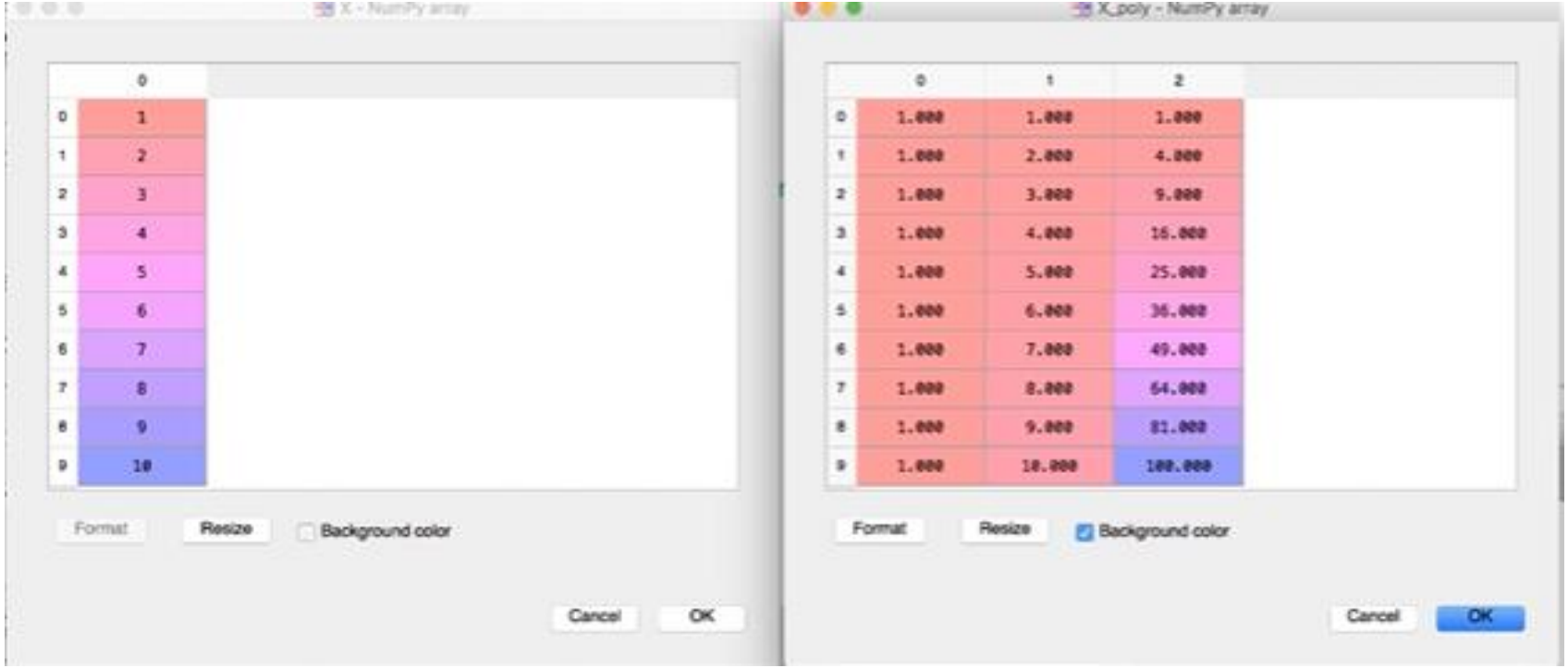
	A	B	C	D	E
1	Position	Level	Salary		
2	Business Analyst	1	45000		
3	Junior Consultant	2	50000		
4	Senior Consultant	3	60000		
5	Manager	4	80000		
6	Country Manager	5	110000		
7	Region Manager	6	150000		
8	Partner	7	200000		
9	Senior Partner	8	300000		
10	C-level	9	500000		
11	CEO	10	1000000		
12					
13					

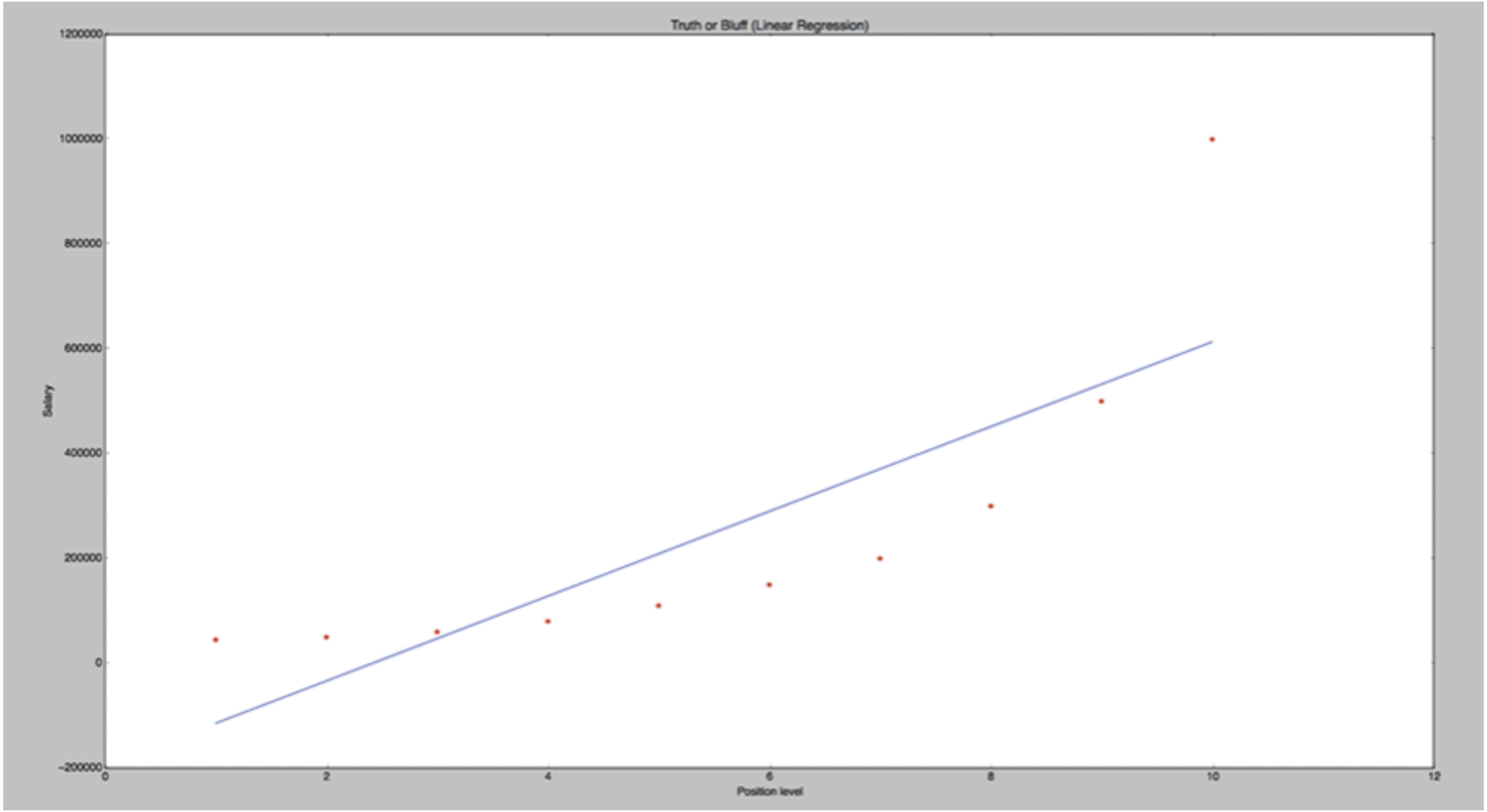
Our Challenge

- In HR Team, will be necessary to hire a new employee.
- The employee informed a earlier salary in the previous company
- The mission is try to create a bluffing detector, if the new potential employee is lying about the earlier salary
- Our machine learning model needs to find out the correlation between the level and the salary to predict if the employee is bluffing the salary or not.


0
0 1
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 10
Format
Resize
Background color
0
0 45000
1 50000
2 60000
3 80000
4 110000
5 150000
6 200000
7 300000
8 500000
9 1000000
Format
Resize
Background color

X_poly Matrix (Column Square)




Truth or Bluff (Linear Regression)
1200000
1000000
800000
600000
400000
200000
0
−200000
Salary
0
2
4
6
8
10
12
Position level

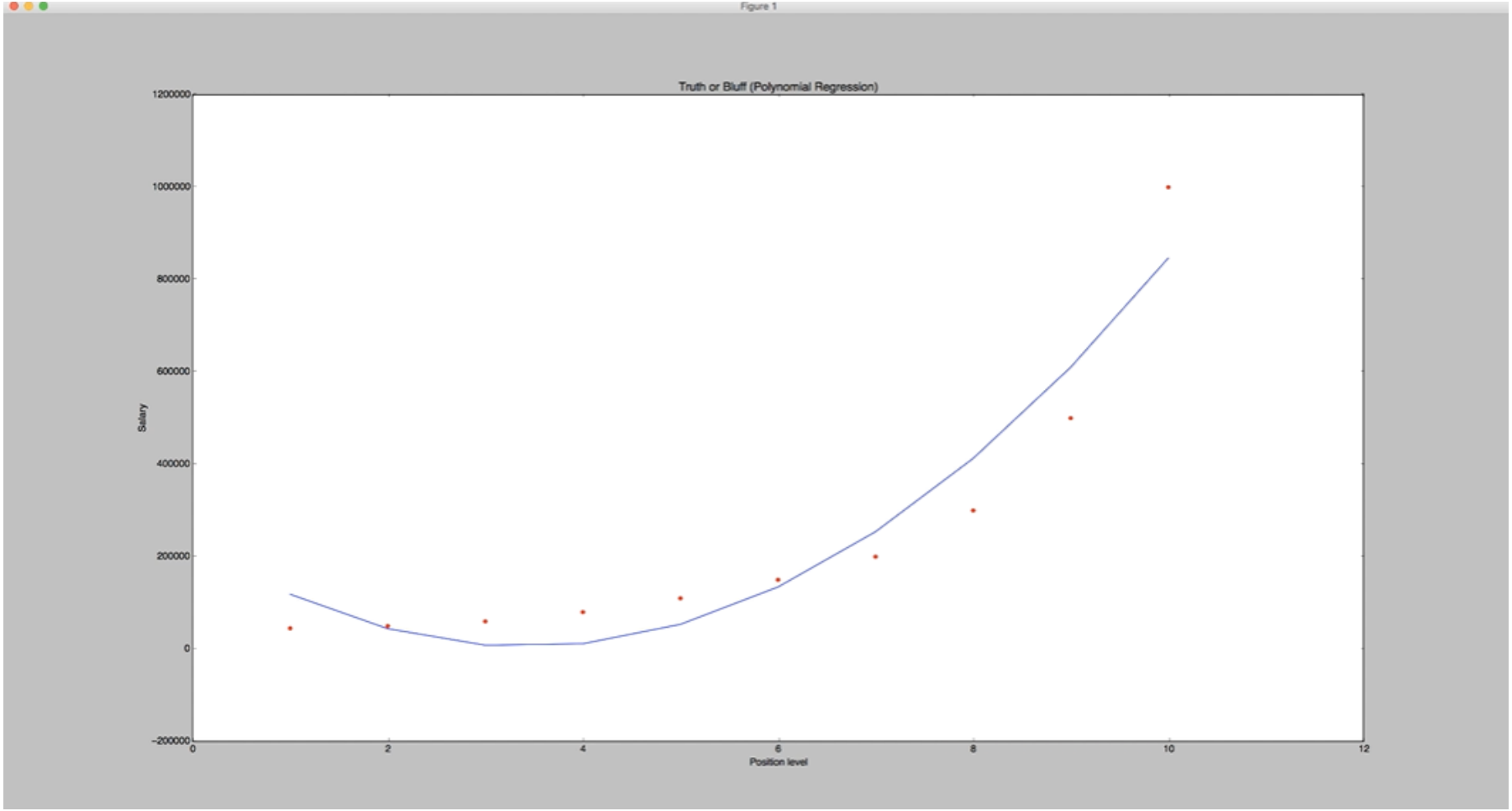
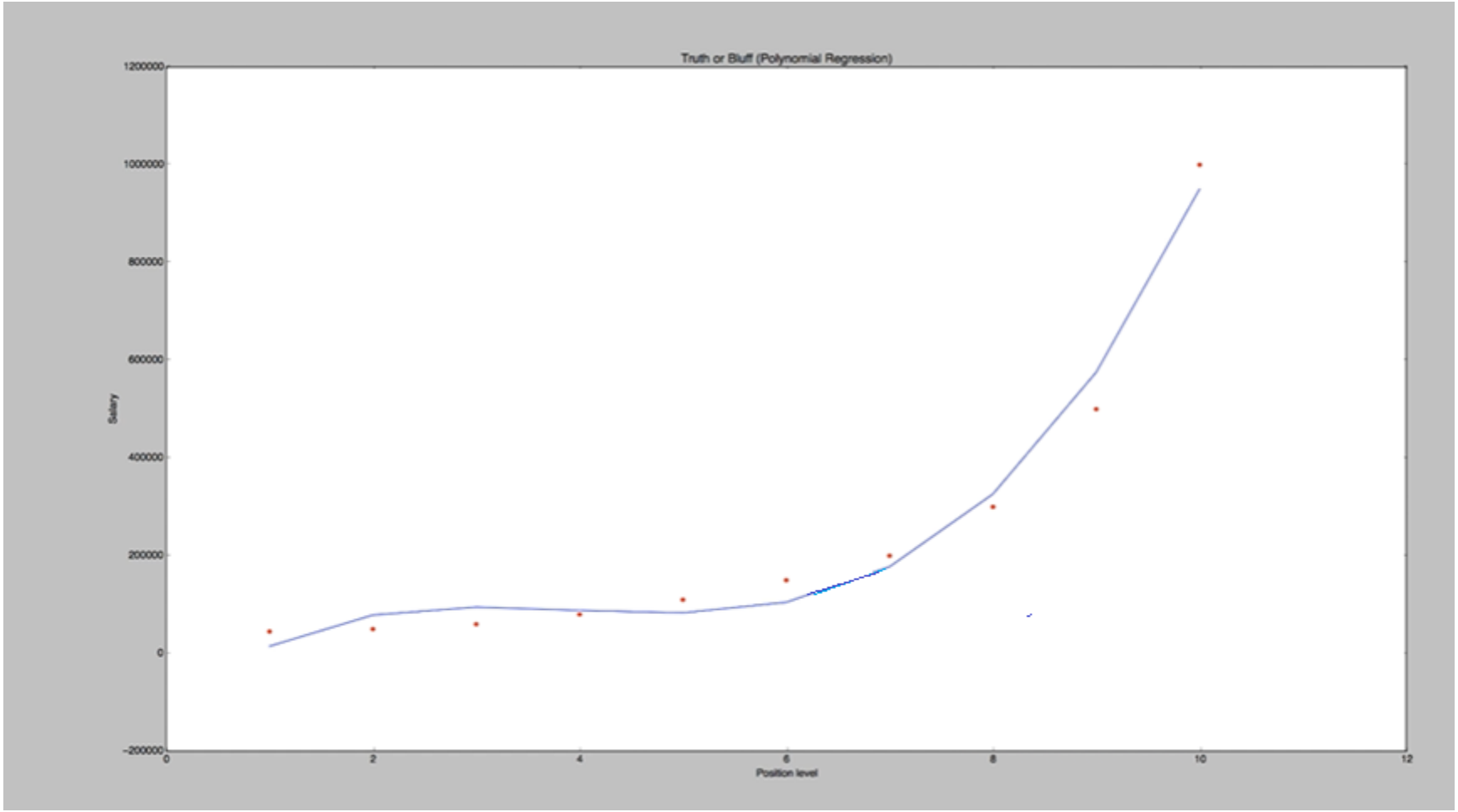
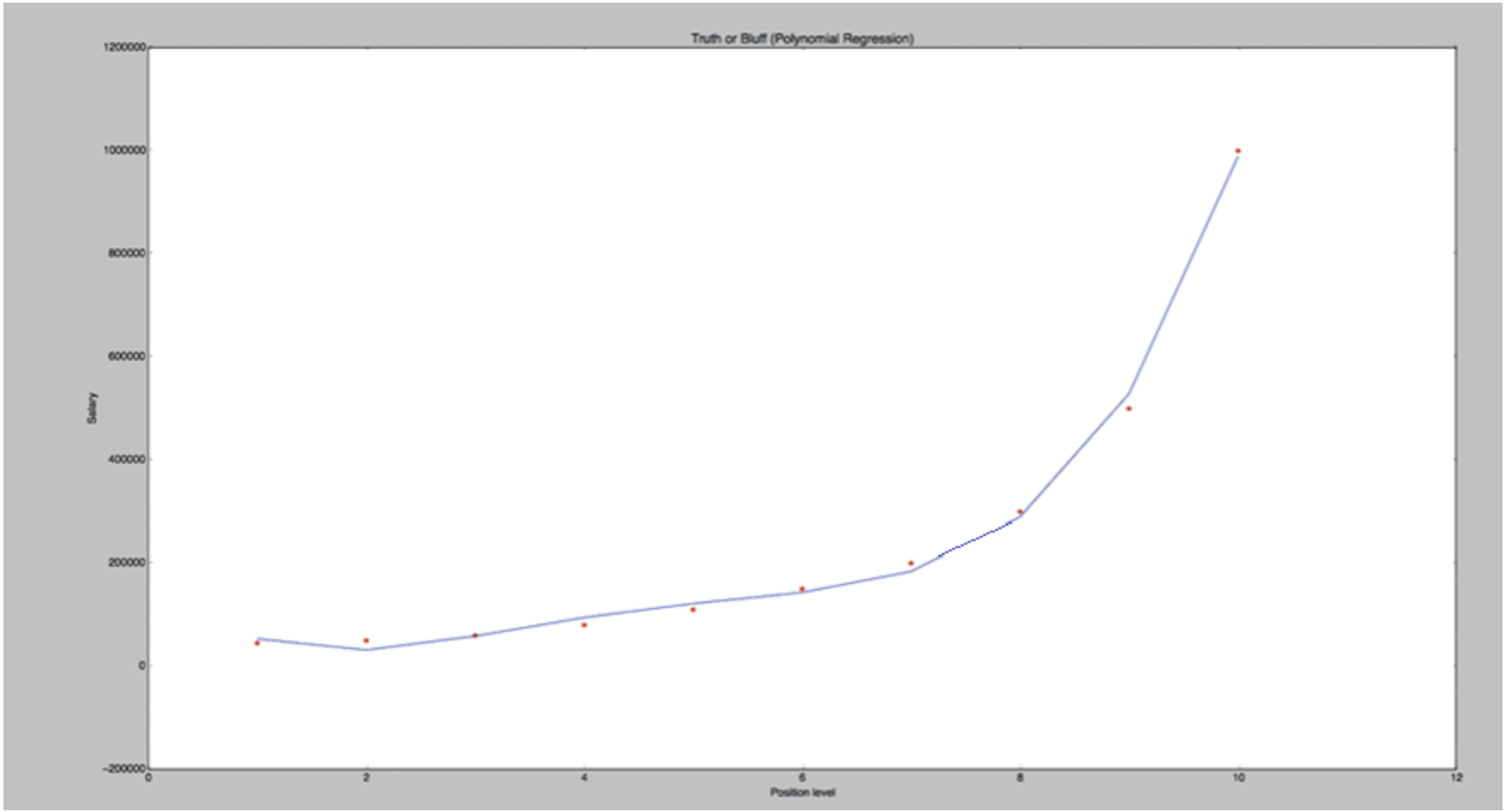


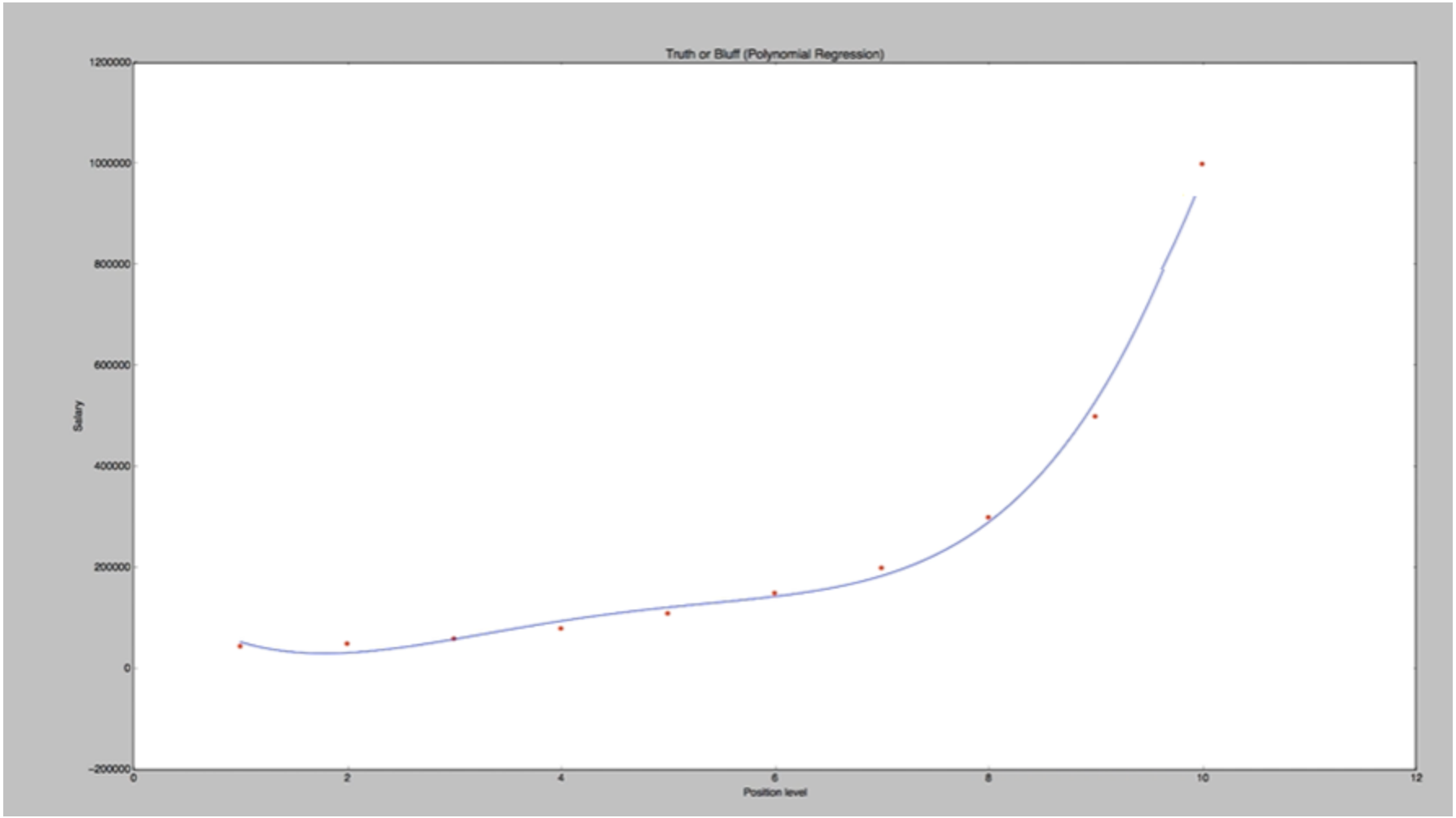
Figure 1
Truth or Bluff (Polynomial Regression)
Salary
Position level
1200000
1000000
800000
600000
400000
200000
0
-200000
0
2
4
6
8
10
12


Truth or Bluff (Polynomial Regression)
Salary
Position level
1200000
1000000
800000
600000
400000
200000
0
-200000
0
2
4
6
8
10
12


Truth or Bluff (Polynomial Regression)
Salary
Position level
1200000
1000000
800000
600000
400000
200000
0
-200000
0
2
4
6
8
10
12

Adjusting the degree
= 2,3,4 , etc

```
8 # Fitting Polynomial Regression to the dataset
9 from sklearn.preprocessing import PolynomialFeatures
0 poly_reg = PolynomialFeatures(degree = 4)
```


Truth or Bluff (Polynomial Regression)
Salary
Position level
1200000
1000000
800000
600000
400000
200000
0
-200000
0
2
4
6
8
10
12

```
# Polynomial Regression

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

# Importing the dataset
dataset = pd.read_csv('Position_Salaries.csv')
X = dataset.iloc[:, 1:2].values
y = dataset.iloc[:, 2].values

# Splitting the dataset into the Training set and Test set
"""from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)"""

# Feature Scaling
"""from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)"""
```

```
# Fitting Linear Regression to the dataset
from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(X, y)

# Fitting Polynomial Regression to the dataset
from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree = 4)
X_poly = poly_reg.fit_transform(X)
poly_reg.fit(X_poly, y)
lin_reg_2 = LinearRegression()
lin_reg_2.fit(X_poly, y)

# Visualising the Linear Regression results
plt.scatter(X, y, color = 'red')
plt.plot(X, lin_reg.predict(X), color = 'blue')
plt.title('Truth or Bluff (Linear Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

```
# Visualising the Polynomial Regression results
plt.scatter(X, y, color = 'red')
plt.plot(X, lin_reg_2.predict(poly_reg.fit_transform(X)), color = 'blue')
plt.title('Truth or Bluff (Polynomial Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()

# Visualising the Polynomial Regression results (for higher resolution and smoother curve)
X_grid = np.arange(min(X), max(X), 0.1)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X, y, color = 'red')
plt.plot(X_grid, lin_reg_2.predict(poly_reg.fit_transform(X_grid)), color = 'blue')
plt.title('Truth or Bluff (Polynomial Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()

# Predicting a new result with Linear Regression
lin_reg.predict(6.5)

# Predicting a new result with Polynomial Regression
lin_reg_2.predict(poly_reg.fit_transform(6.5))
```

Adjusting the Code

- lin_reg.predict([[6.5]])
- lin_reg_2.predict(poly_reg.fit_transform([[6.5]]))

Skill Lab

Muito obrigado!