

Modelos de Regressão

Prof. Gustavo Willam Pereira

Introdução

- Os problemas de machine learning podem ser divididos em aprendizagem supervisionada e não-supervisionada.
- A classificação supervisionada é realizada quando temos disponíveis dados de entrada e conhecemos, para cada entrada, o resultado da saída.
- Podemos dividir a aprendizagem supervisionada em dois grupos: regressão e classificação.
- Na regressão utilizamos dados de entrada para prever valores contínuos.
- Na classificação os dados de entrada são utilizados para prever classes, que são variáveis categóricas.
- Na aprendizagem não-supervisionada os algoritmos utilizam os próprios dados de entrada para encontrar padrões/relações entre esses dados. Nesse caso os dados não apresentam rótulos pré-definidos.

Regressão Linear Simples

- Para fazer a previsão de valores contínuos iremos utilizar a regressão.
- Podemos ter regressão linear ou não-linear.
- Temos vários tipos de algoritmos para regressão, a seguir vamos descrever alguns mais importantes para *machine learning*.
- **Regressão Linear Simples**
 - A regressão linear simples pode ser explicada como a estimativa de valores de uma variável dependente em função de uma variável independente.
 - A Figura 1 a seguir é uma representação gráfica da variável independente altura no eixo X versus a variável dependente massa corporal no eixo Y.

Regressão Linear Simples

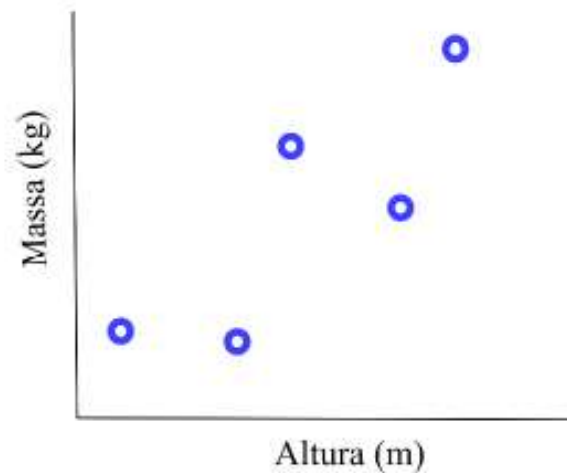


Figura 1 – Gráfico da variável independente altura versus a variável dependente massa.

Regressão Linear Simples

- Na Figura 2 foram projetados os valores das massas no eixo Y e determinado a média das massas.
- A média seria uma estimativa para a massa para qualquer valor no eixo X (altura).
- Poderíamos calcular a diferença entre cada valor de massa até a média das massas.
- Para que os valores negativos não anulem valores positivos, elevamos a diferença ao quadrado.
- A soma da diferença ao quadrado é a nossa Soma de quadrado total (SQT), conforme apresentado na Figura 2.

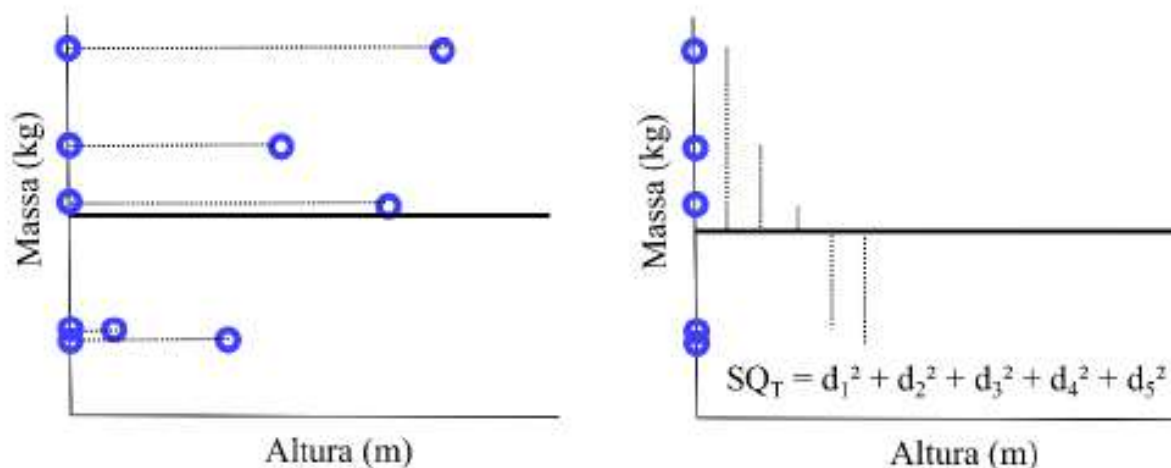


Figura 2 – Estimativa dos valores pela média da massa.

Regressão Linear Simples

- Podemos obter uma soma dos quadrados dos resíduos menor que aquela obtida na Figura 2 modificando o ângulo da reta.
- Veja na Figura 3 que para diferentes ângulos seria obtido um valor menor da soma dos quadrados (SQ_1 , SQ_2) até chegar em um valor mínimo (SQ_R).

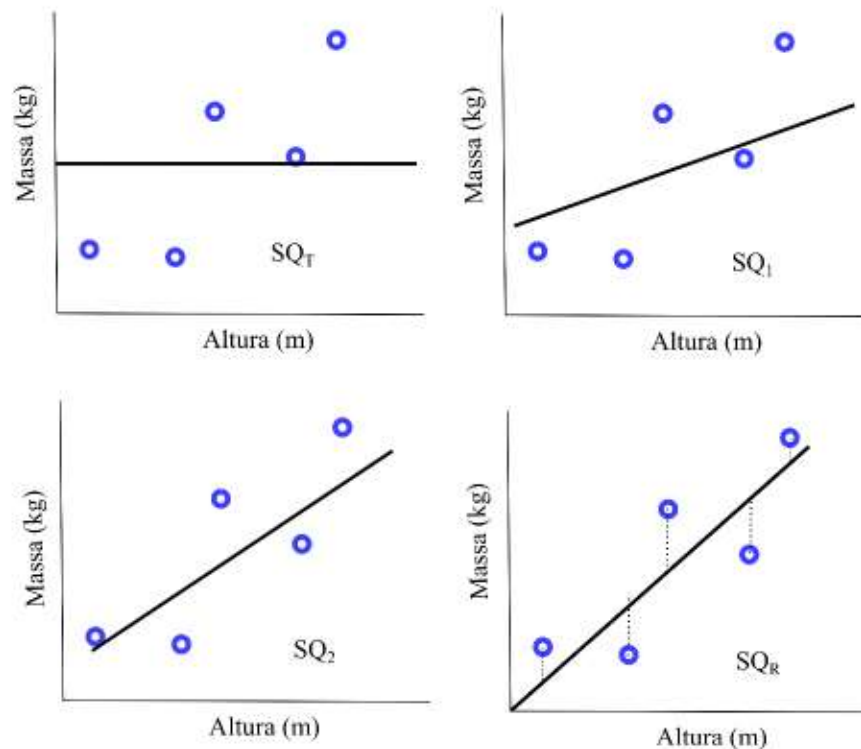


Figura 3. Modificação do ângulo da reta até chegar em um valor da soma dos quadrados dos resíduos mínimo

Regressão Linear Simples

- O modelo para a regressão linear simples pode ser representado pela equação:

$$y = b_0 + b_1 * x_1$$

- Na equação acima, y é a variável dependente e x1 é a variável independente. O parâmetro b0 é o intercepto e b1 é a inclinação da reta.
- Conforme Figura 2 e 3, de forma intuitiva, com os valores da soma dos quadrados total (SQ_T) e o valor da soma dos quadrados da regressão (SQ_R) podemos obter o coeficiente de determinação R².

Regressão Linear Simples

- O coeficiente de determinação, equação abaixo, é muito utilizado para avaliar a qualidade do modelo.
- O coeficiente de determinação nos indica quantos por cento a massa corporal é explicada pela altura.

$$R^2 = \frac{SQ_T - SQ_R}{SQ_T}$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (x_i - \hat{x}_i)^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

- O RMSE é um valor absoluto que está na mesma unidade da variável (y), descreve a precisão do modelo, o que significa quão próximos os valores previstos estão dos valores reais

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2}$$

Regressão Linear Múltipla

- A regressão linear simples pode ser considerada como um caso particular de uma regressão linear múltipla.
- A seguir é apresentado a equação geral para um modelo de regressão linear múltipla. Na regressão múltipla temos várias variáveis.

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

- **P-Value**

- Como regra geral, um bom modelo de regressão é aquele que tem um elevado R^2 com menor número de variáveis, ou seja, devemos utilizar variáveis realmente importantes para o modelo.
- Modelos de regressão com variáveis pouco importantes poderá gerar modelos ruins, primeiro pelo pior ajuste e segundo pela complexidade.

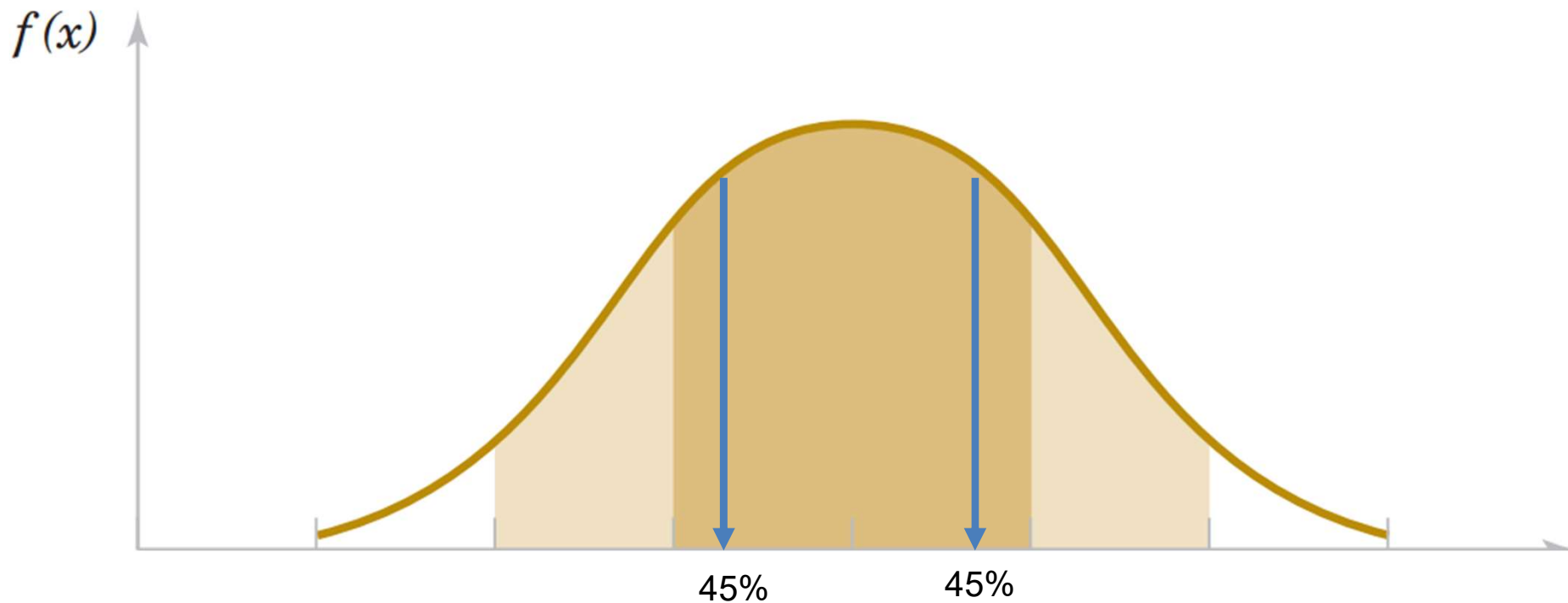
P-Value

- Então, quando estamos gerando um modelo de regressão múltipla, devemos verificar quais as variáveis são realmente importantes no modelo.
- Para verificar se uma determinada variável é realmente importante para o modelo, podemos testar se o parâmetro beta da variável é diferente de zero.
- Se beta for igual a zero significa que o parâmetro em questão não é importante para o modelo. Em um modelo de regressão, nossa hipótese nula é que uma determinada variável, x_1 por exemplo, não afeta o modelo ($\beta = 0$).

P-Value

- Em estatística, assumimos que a hipótese nula é verdadeira, a não ser que a nossa amostra seja tão “estranha” que nos leve a pesar que podemos considerar que a hipótese nula é falsa, ou seja, rejeitar a hipótese nula.
- Para verificar o quão estranha é nossa amostra, determinamos o p-valor.
- O p-valor é a probabilidade de obtermos uma amostra como a nossa ou mais extrema que a nossa, se a hipótese nula é verdadeira.
- Então, se o p-valor for de 0,90, ele indica que temos 90% de chance de pegarmos uma amostra, como a nossa ou mais extrema, se a hipótese nula é verdadeira, ou seja, nossa amostra não é nada “estranha” com 90% de probabilidade.
- Dessa forma, com p-valor de 90% não temos evidências para rejeitar a hipótese nula

P-Value



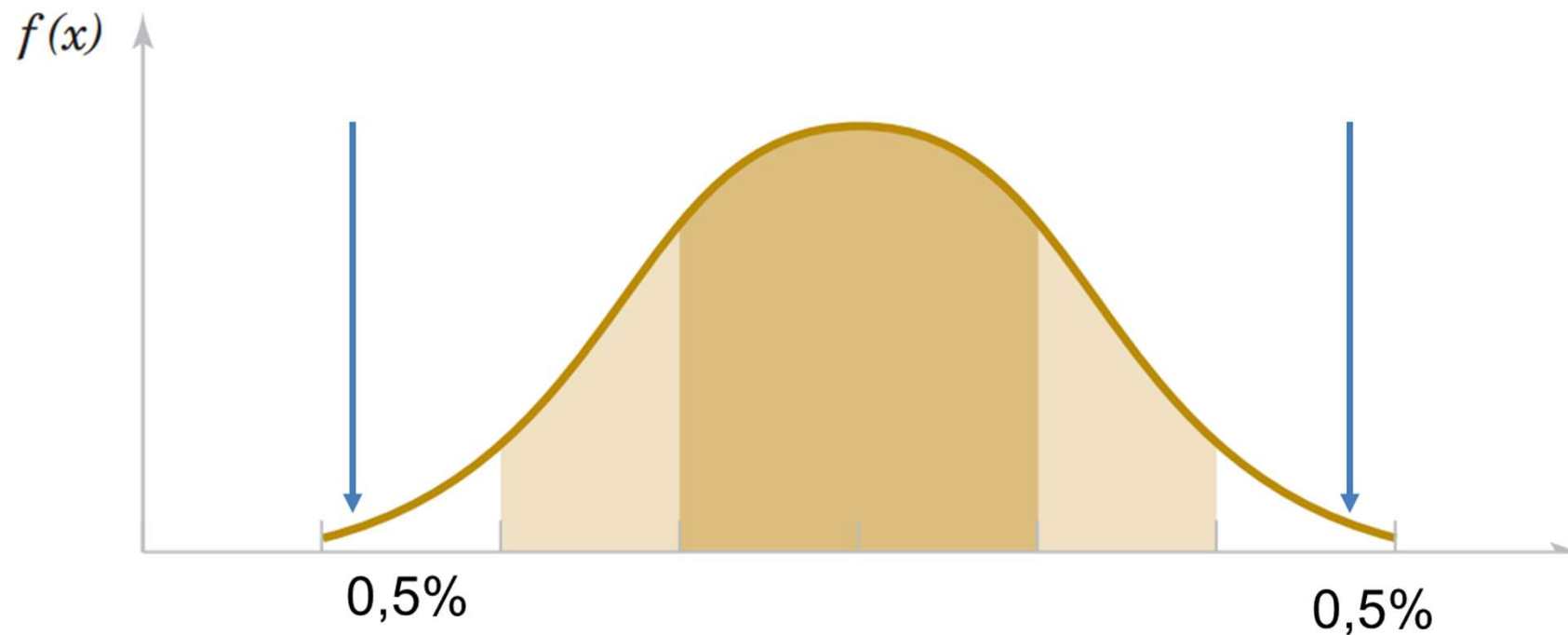
P-value = 90%

Com p-valor de 90% não temos evidências para rejeitar a hipótese nula

P-Value

- Se o p-valor for de 0,01, isso indica que temos probabilidade de 1% de pegarmos uma amostra como a nossa ou mais extrema, se a hipótese nula é verdadeira.
- Podemos assim, considerar que essa amostra é “estranha”.
- Assim, começamos a pensar que com um valor muito baixo de probabilidade, podemos pensar que talvez a hipótese nula seja falsa.
- Normalmente consideramos que p-valor abaixo de 5% poderíamos rejeitar a hipótese nula.

P-Value



P-value = 1%

Com p-valor de 1% podemos pensar que a hipótese nula seja falsa

Regressão Polinomial

- Podemos considerar que a regressão polinomial é um caso particular da regressão múltipla.
- A diferença é que na regressão polinomial iremos utilizar uma variável e as demais “variáveis” serão geradas pela variável original elevada a uma potência.
- $$y = b_0 + b_1 * x_1 + b_2 * x_1^2 + \dots + b_n * x_1^n$$
- Vamos utilizar o Scikit Learn para fazer um modelo de regressão linear polinomial.
- No código a seguir vamos importar o banco de dados e criar as variáveis X e y.

Regressão Polinomial

```
8 import pandas as pd
9
10 dados = pd.read_csv('Position_Salaries.csv')
11
12 ### Separar dados de treinamento e dados de teste.
13
14 X = dados[['Level']] #slice no dataframe e retorna dataframe
15 y = dados[['Salary']]
```

- Veja que temos uma variável ('Level'). Quanto maior o nível maior será o salário.
- Então vamos gerar um modelo polinomial para essa regressão e plotar os gráficos.

Regressão Polinomial

```
17 from sklearn.linear_model import LinearRegression
18 import matplotlib.pyplot as plt
19
20 #Regressão linear simples
21 regressor = LinearRegression() #esse método já considera a constante
22
23 regressor.fit(X, y)
24
25 # Predicting the Test set results
26 y_pred = regressor.predict(X)
27
28 plt.plot(X['Level'], y_pred)
29
30 plt.scatter(X['Level'], y['Salary'])
```

Regressão Polinomial

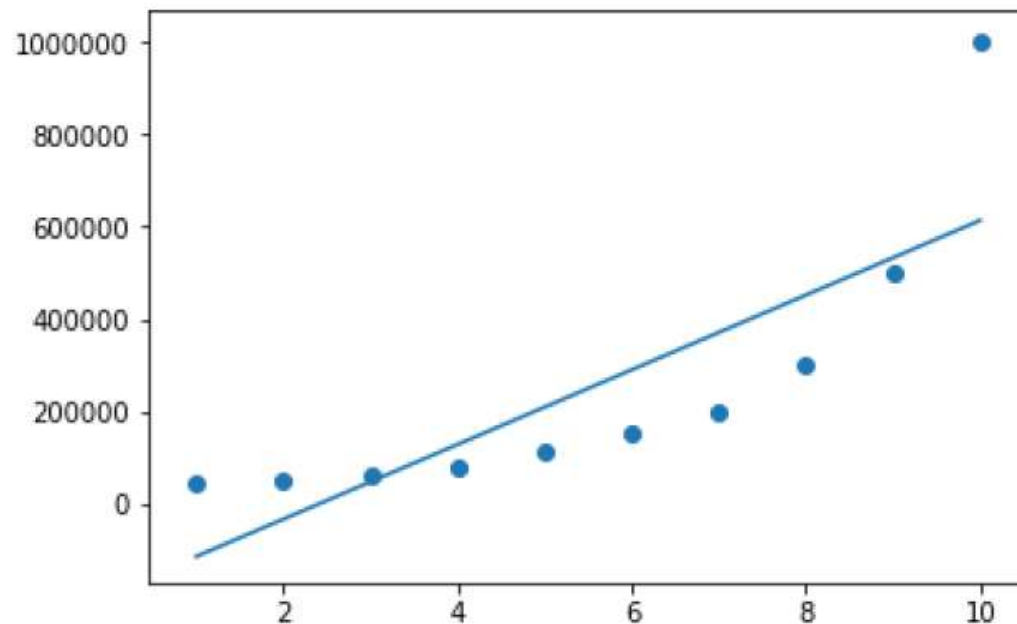


Figura 7. Regressão linear

Nesse código foi ajustado um modelo de regressão linear simples. Agora vamos criar uma variável. Essa nova variável será a variável 'Level' elevada ao quadrado (grau 2). Veja o código e o resultado gráfico.

Regressão Polinomial

```
32 #Regressão Polinomial (grau 2)
33 X.insert(0, 'Level2', X['Level']**2)
34
35 poly_model = LinearRegression()
36 poly_model.fit(X,y)
37
38 y_pred = poly_model.predict(X)
39
40 plt.plot(X['Level'], y_pred)
41
42 plt.scatter(X['Level'], y['Salary'])
```

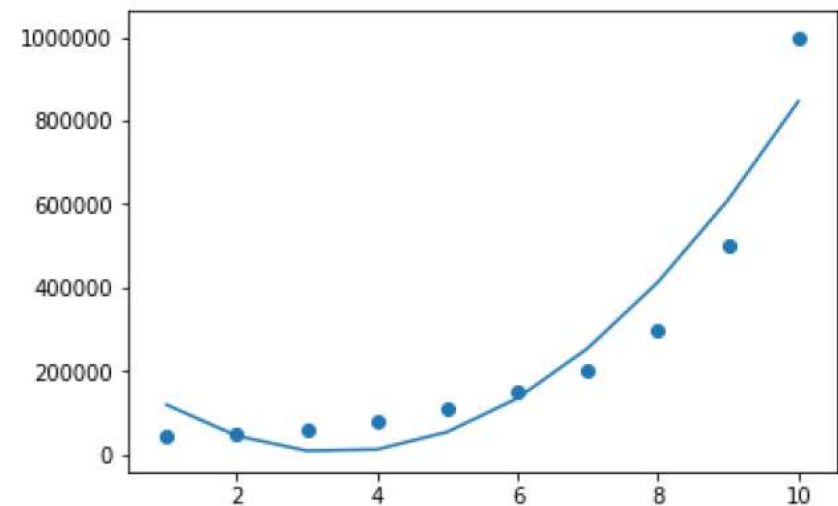


Figura 8. Regressão polinomial de grau 2.

Agora vamos acrescentar uma outra variável que será a variável ‘Level’ com potência 3 (grau 3).

Veja o código e o resultado gráfico.

Regressão Polinomial

```
44 #Regressão Polinomial (grau 3)
45 X.insert(0, 'Level3', X['Level']**3)
46
47 poly_model2 = LinearRegression()
48 poly_model2.fit(X,y)
49
50 y_pred = poly_model2.predict(X)
51
52 plt.plot(X['Level'], y_pred)
53
54 plt.scatter(X['Level'], y['Salary'])
55
56 poly_model2.coef_
```

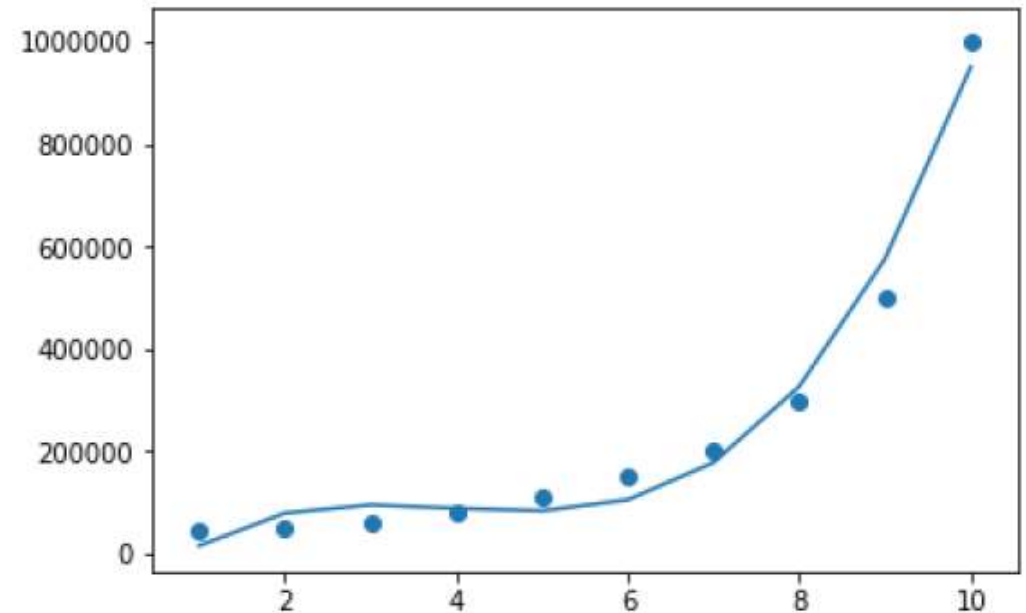


Figura 8. Modelo de regressão polinomial de grau 3



INSTITUTO FEDERAL
Sudeste de Minas Gerais