

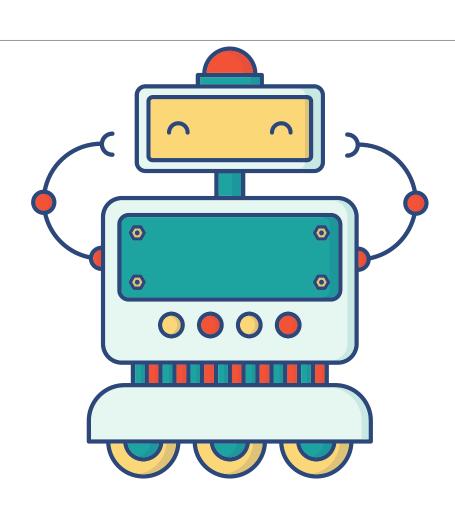
# APRENDIZADO DE MÁQUINA



# Aprendizado de Máquina

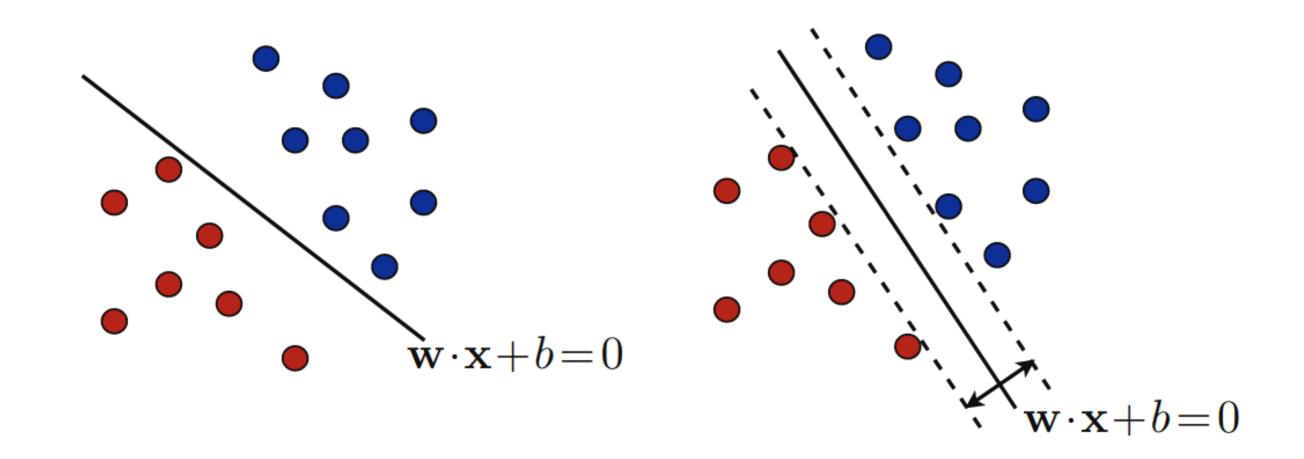
**COM SCIKIT-LEARN** 

- 1.0 que é Machine Learning
- 2. Problemas e Ferramentas
- 3. Modelos lineares
- 4. Modelos baseados em probabilidade e similaridade
- 5. Árvores e Florestas aleatórias
- 6. Recapitulação



### Diferenças e melhoria dos modelos

Fundamentos e hiperparâmetros



Aprendizado Complexibilidade Interpretabilidade Seleção de modelos

Imagem: Foundations of Machine Learning

### Modelos de classificação

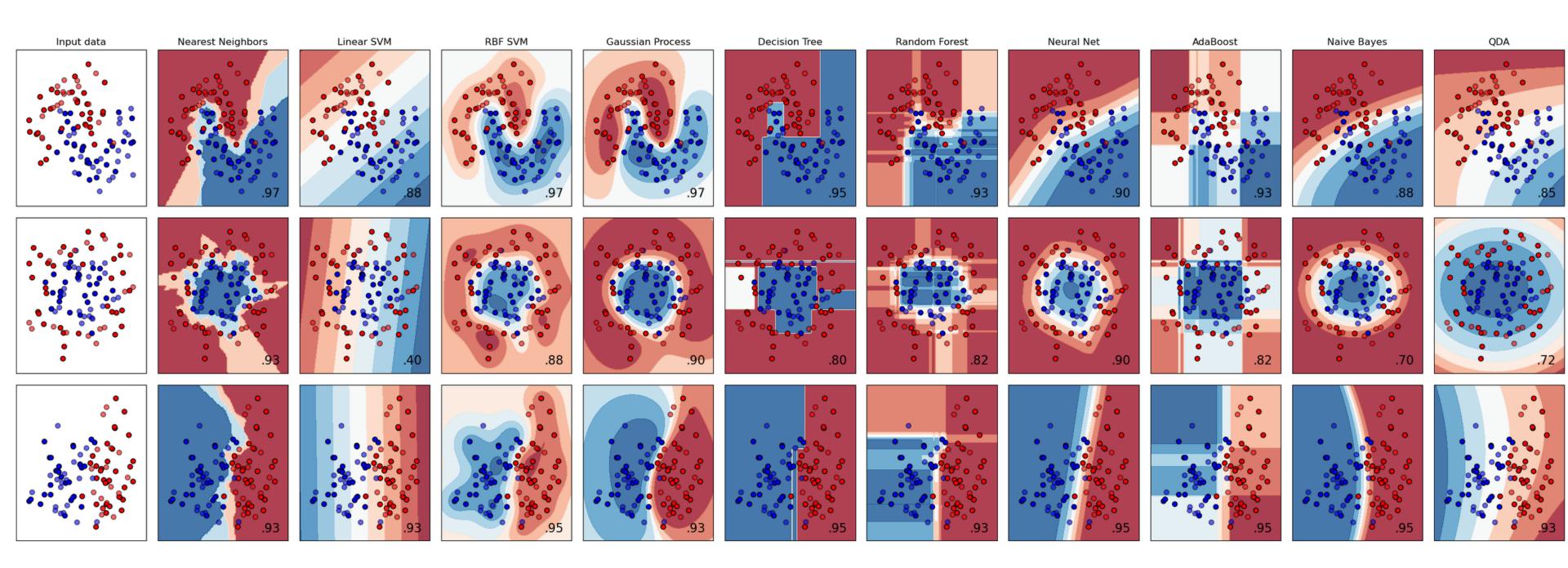
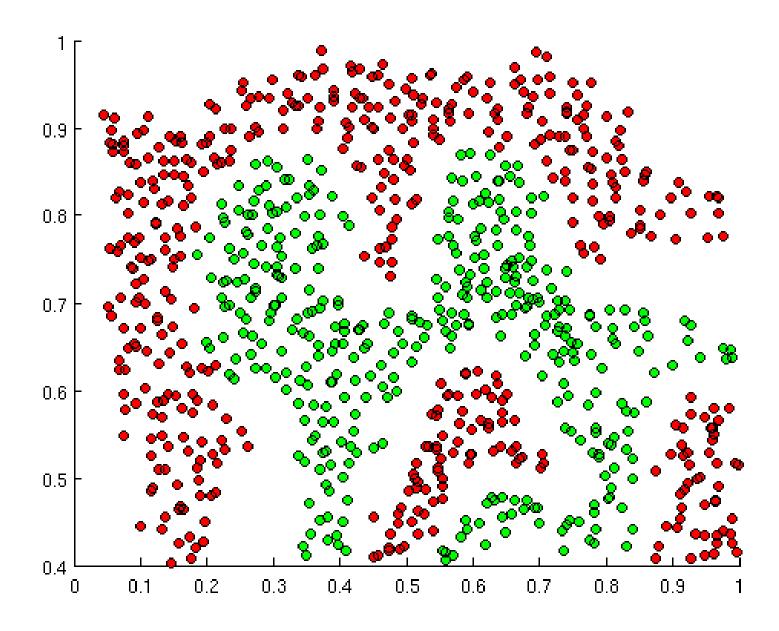


Imagem: Scikit-Learn

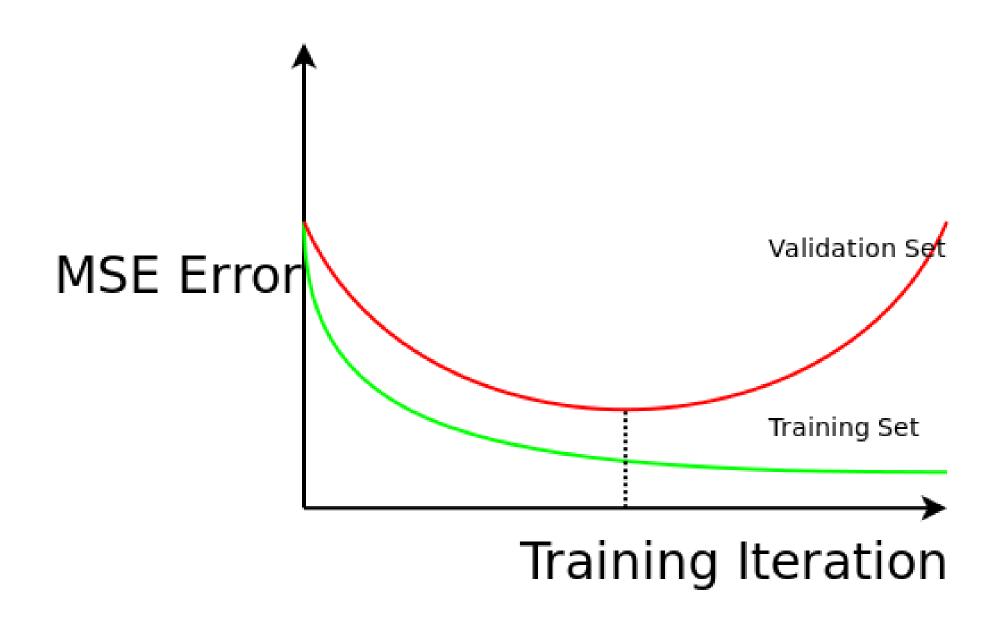
### Aprendizado

- Diferentes suposições para se aprender um padrão pelos dados
- Minimizar o erro
- Calibrar o modelo



### Aprendizado

- Diferentes suposições para se aprender um padrão pelos dados
- Minimizar o erro
- Calibrar o modelo



### Aprendizado

- Diferentes suposições para se aprender um padrão pelos dados
- Minimizar o erro
- Calibrar o modelo

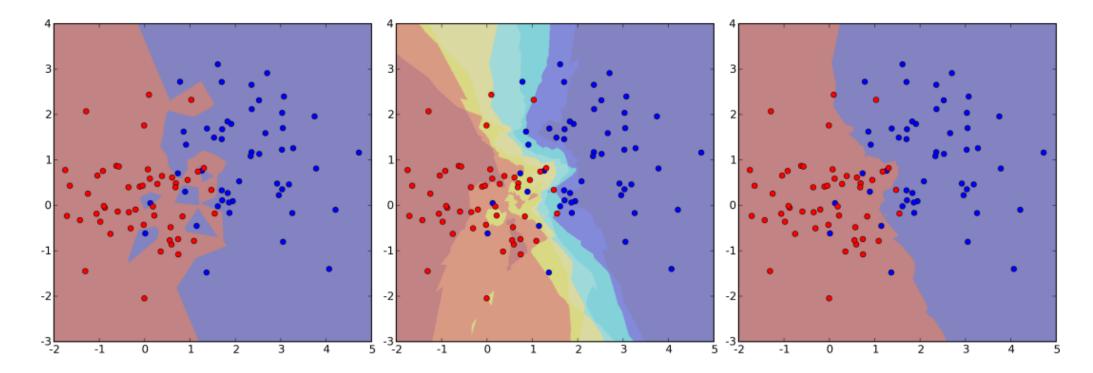
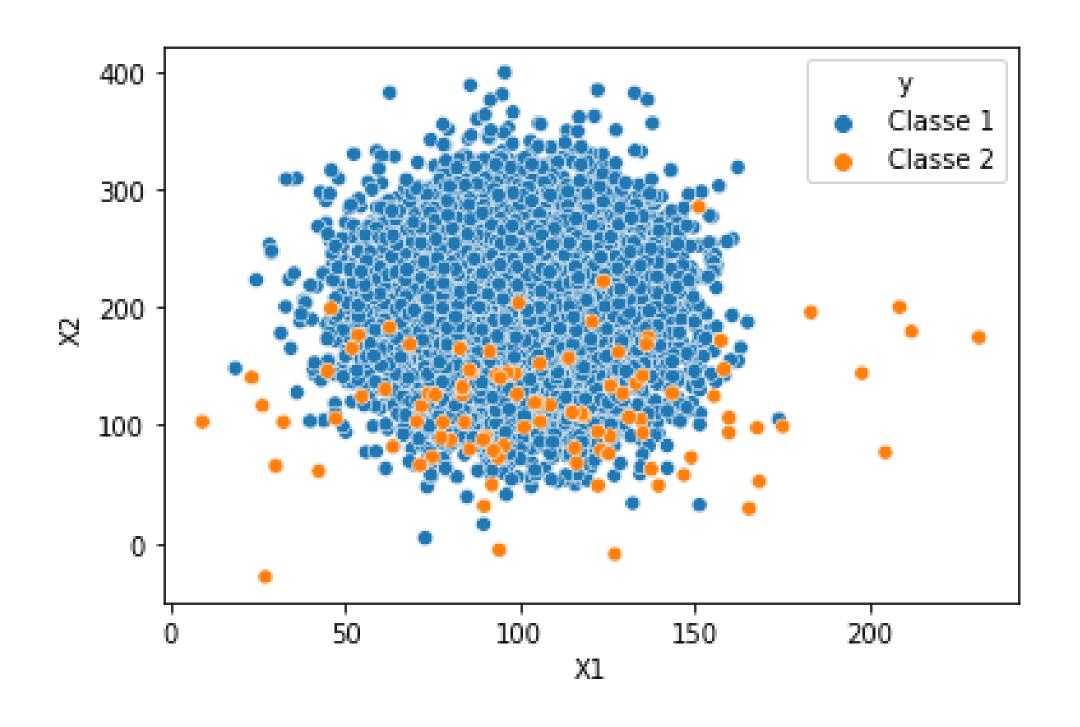


Fig. 1.18. k-Nearest neighbor classifiers using Euclidean distances. Left: decision boundaries obtained from a 1-nearest neighbor classifier. Middle: color-coded sets of where the number of red / blue points ranges between 7 and 0. Right: decision boundary determining where the blue or red dots are in the majority.

### Baseline

### Classe majoritária

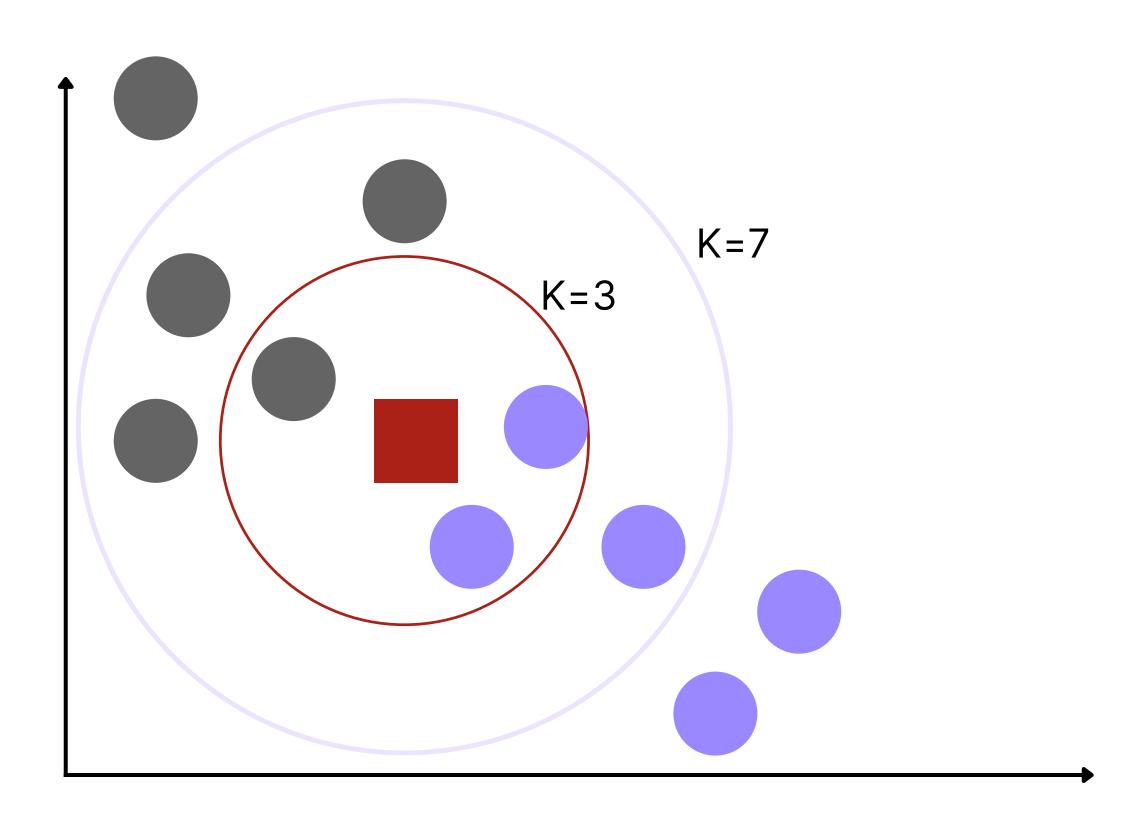
### Aleatório



## K-Nearest Neighbors KNN

### KNN

| Legenda         |  |  |  |
|-----------------|--|--|--|
| Classe negativa |  |  |  |
| Classe positiva |  |  |  |
| Novo exemplo    |  |  |  |



### Outliers

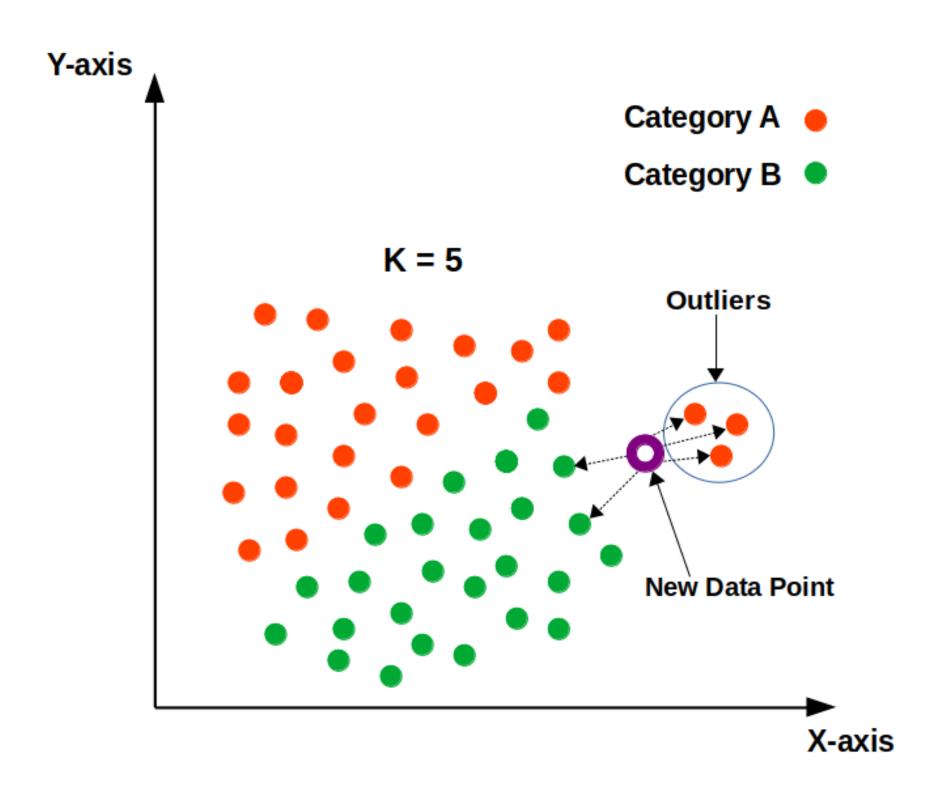


Imagem: <u>analyticsvidhya</u>

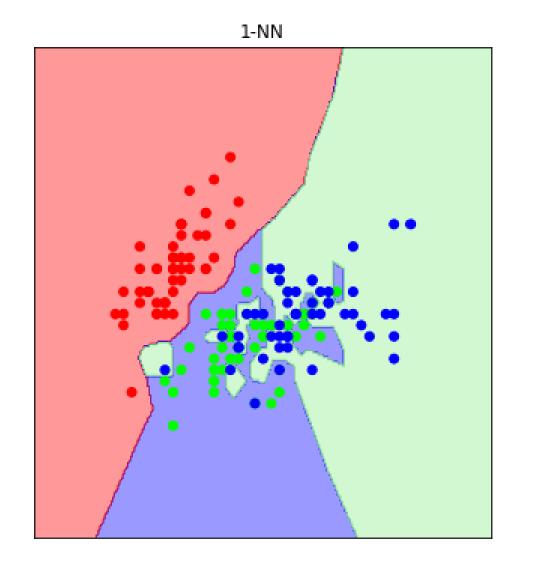
### Valor de K

Overfitting

Underfitting

K baixo: sobreajuste

• K alto: subajuste



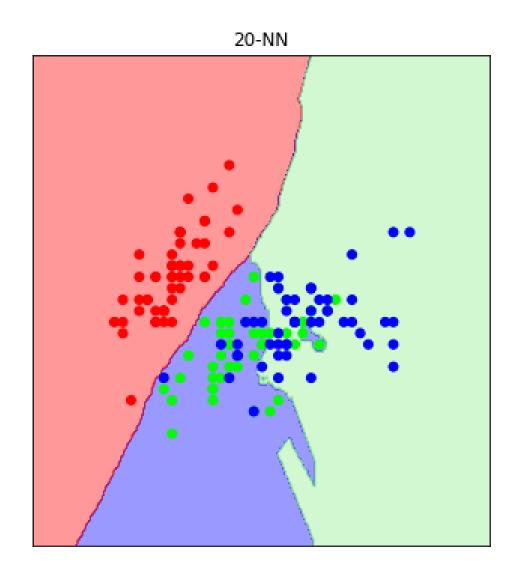


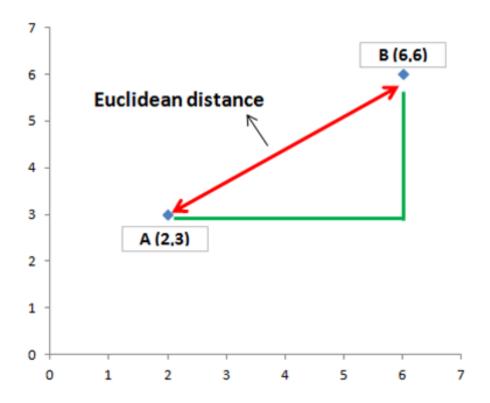
Imagem: winder.ai

### Variáveis numéricas

### Necessidade de escala

$$D(a,b) = \sqrt{\sum_{i=1}^{n} (b_i - a_i)^2}$$

"Semelhança" e distância



## Documentação

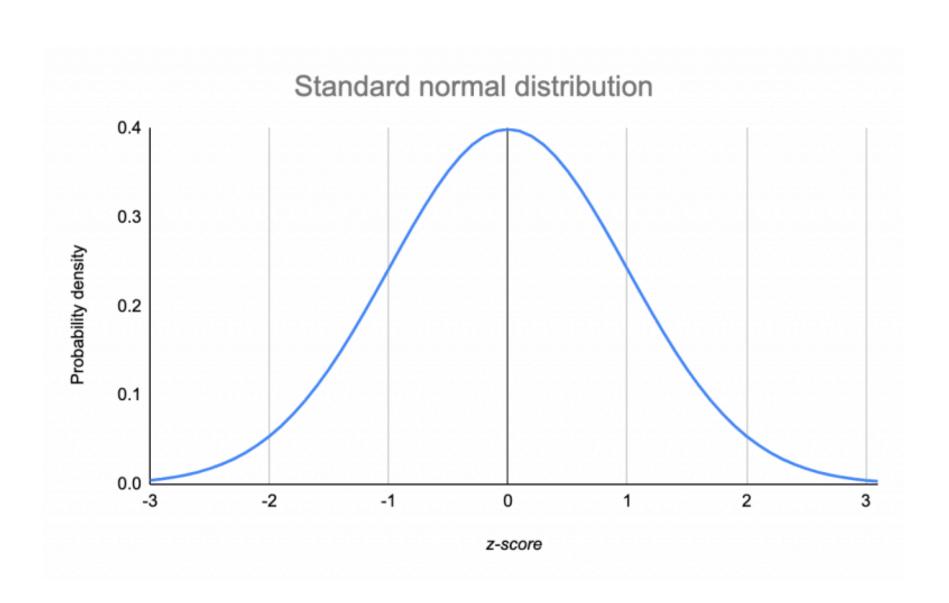


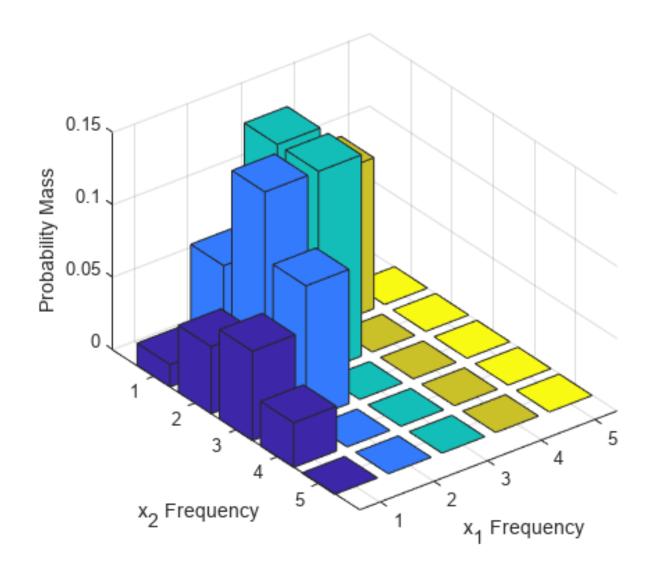
# Código



# Naive Bayes Multinomial e Gaussiano

# Distribuições de Probabilidade

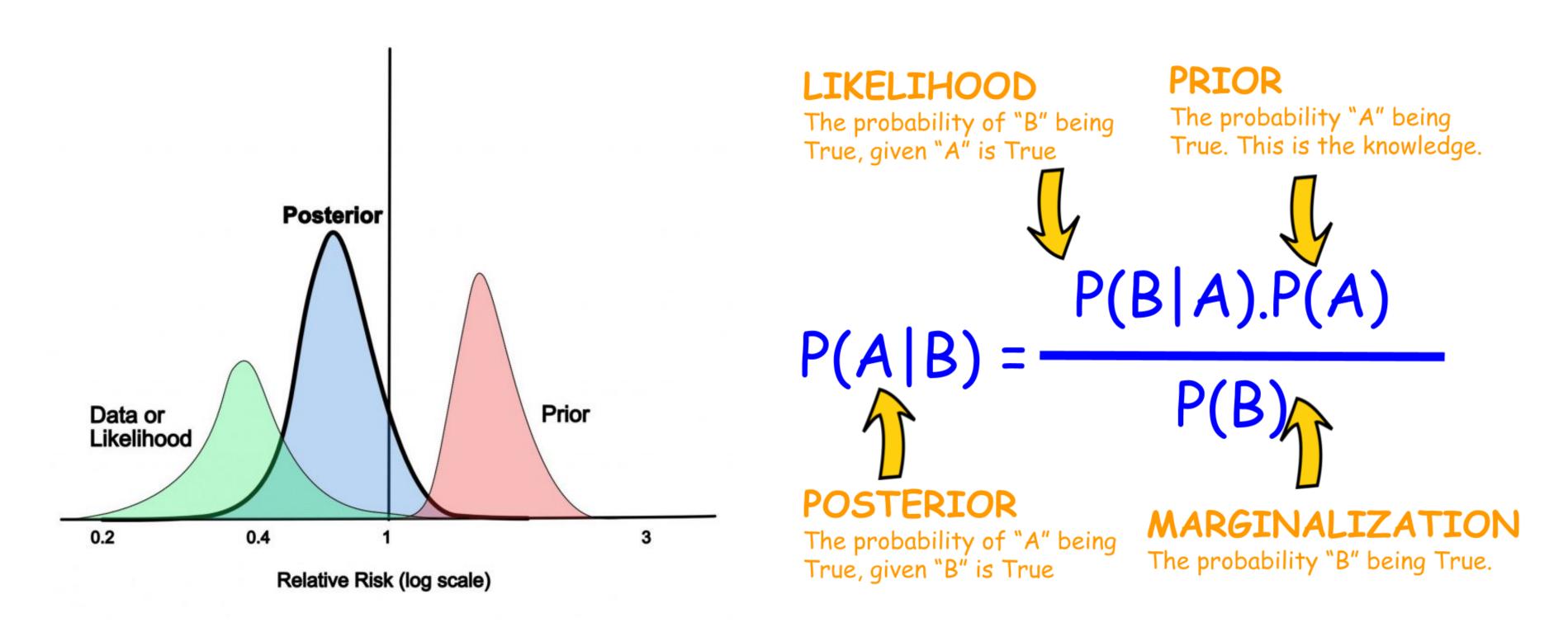




Normal: scribbr

Multinomial: mathworks

### Teorema de Bayes



### Naive Bayes

| Dinheiro | Convite | Amigo | Parabéns | Classe      |
|----------|---------|-------|----------|-------------|
| 2        | 0       | 0     | 1        | Spam        |
| 1        | 1       | 2     | Ο        | Spam        |
| 0        | 0       | 2     | 1        | Spam        |
| 0        | 2       | 0     | 1        | Não<br>Spam |
| 0        | 1       | 2     | 0        | Não<br>Spam |
| 1        | 1       | 0     | 0        | Não<br>Spam |

P(Spam | [Convite, Dinheiro])

- = P(Spam)\*P(Convite | Spam)\*P(Dinheiro | Spam)
- = 0.5\*(1/10)\*(3/10)
- = 0,015

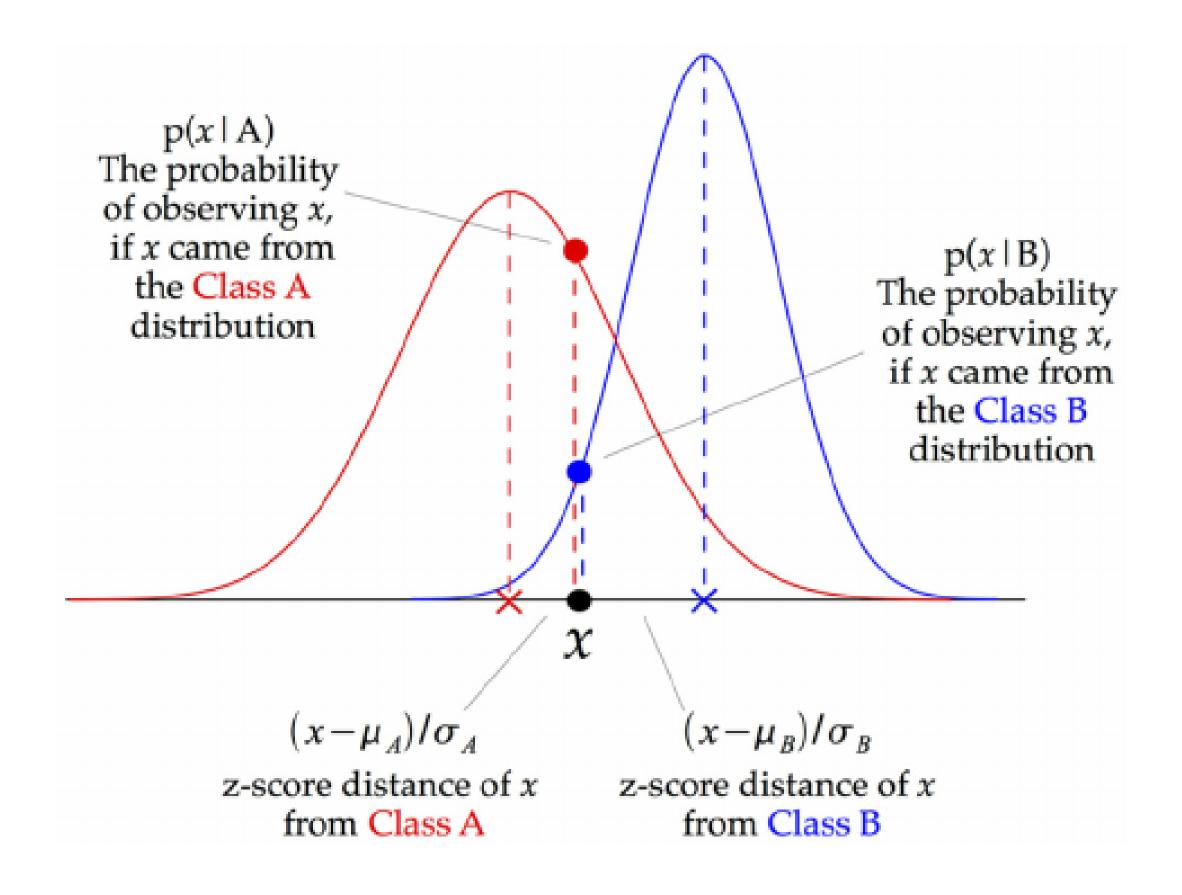
P(Não Spam | [Convite, Dinheiro])

- = P(N Spam)\*P(Convite N Spam)\*P(Dinheiro N Spam)
- = 0.5\*(4/8)\*(1/8)
- = 0,03125

```
P(Não Spam | [Convite, Dinheiro])
```

P(Spam | [Convite, Dinheiro])

### Gaussian Naive Bayes



Imagens: <u>opengenus</u>

### Naive Bayes

Utiliza uma distribuição esperada dos dados Problemas com probabilidade 0 (alfa / laplace) - Multionomial

Todas as variáveis tem o mesmo peso

Rápido e enviesado para as distribuições

## Documentação



### 1.9. Naive Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the val...

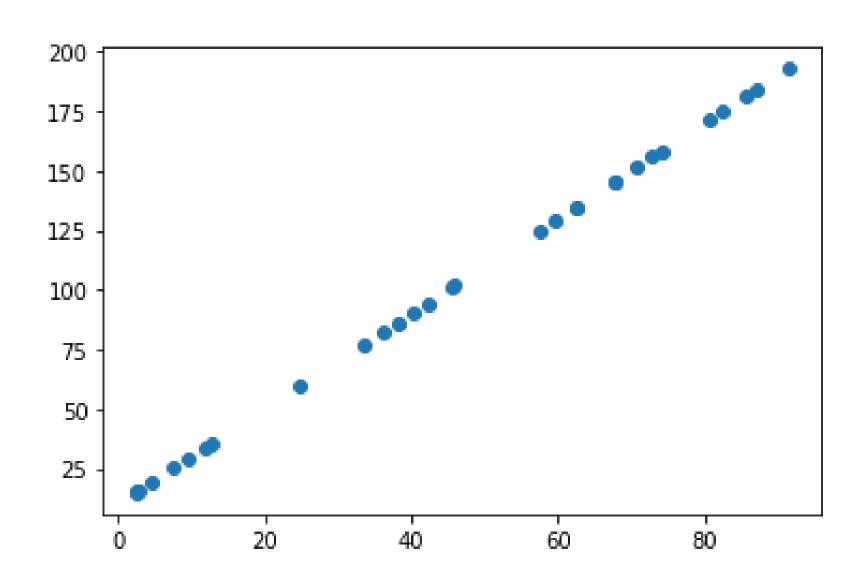
scikit-learn

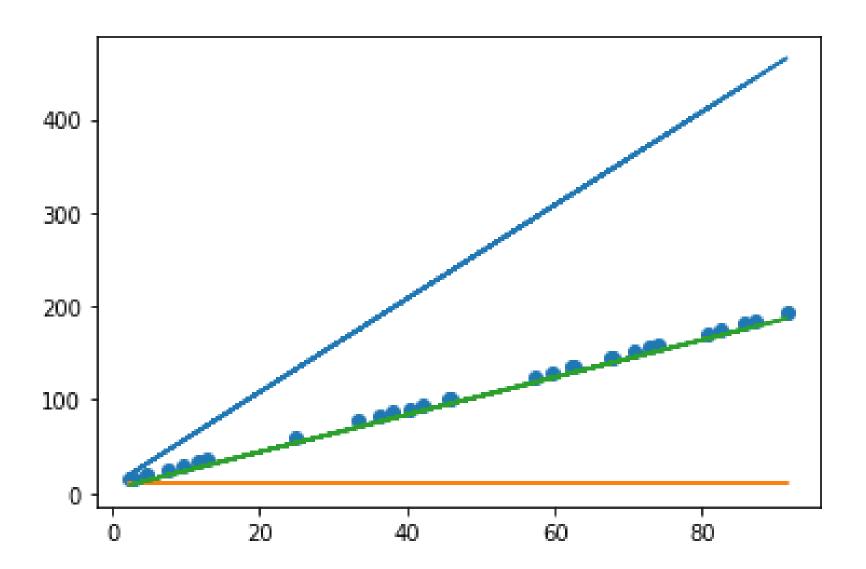
# Código



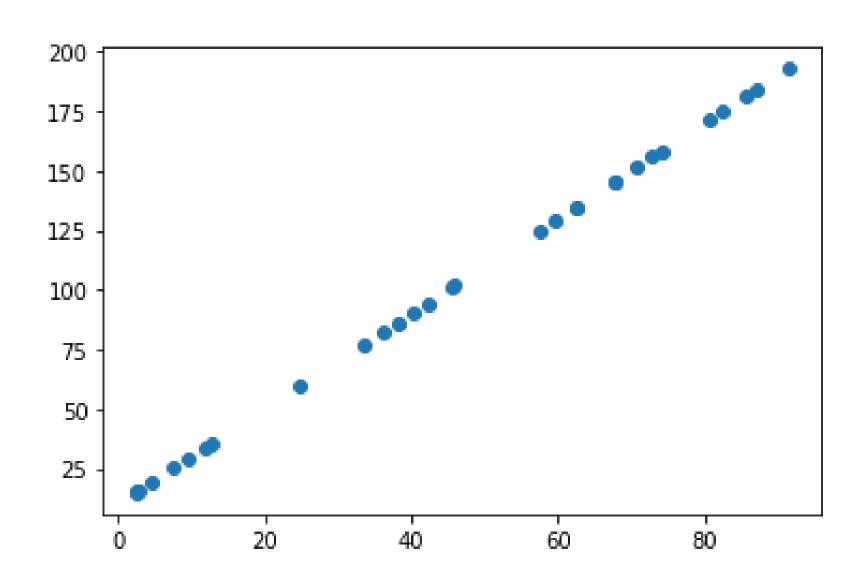
## Regressão linear

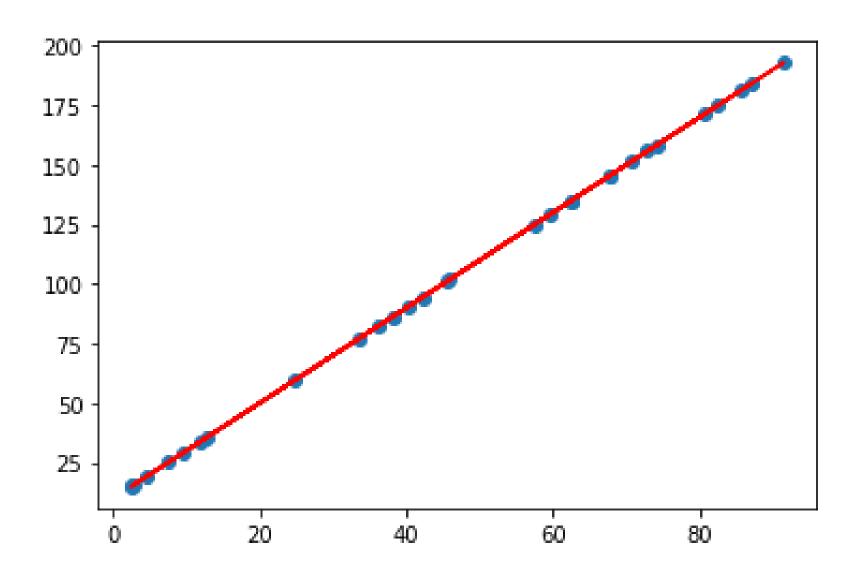
## Regressão Linear





# Regressão Linear

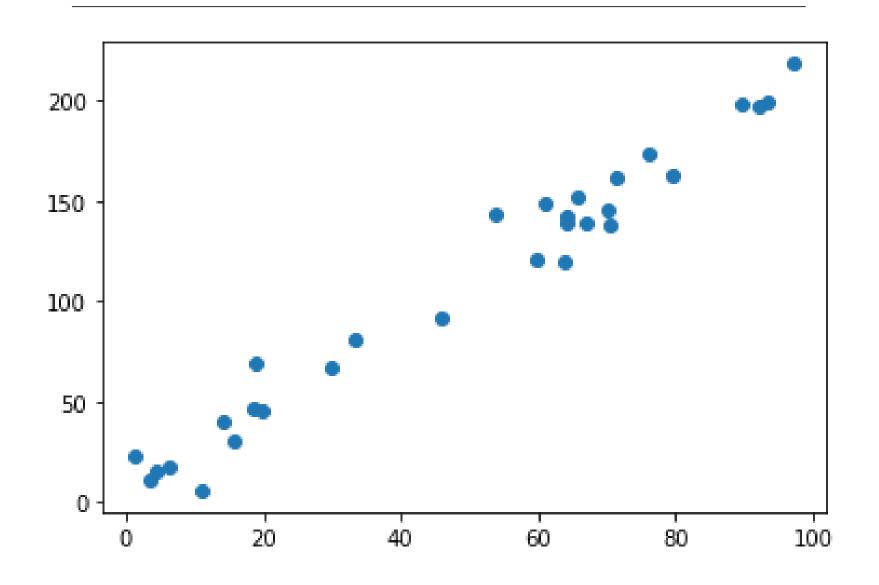




### Ruído

### Não linearidade

Como os dados se comportam na realidade. Tendências lineares não são incoumns Apesar de minimizar um erro, segue-se uma suposição do comportamento dos dados



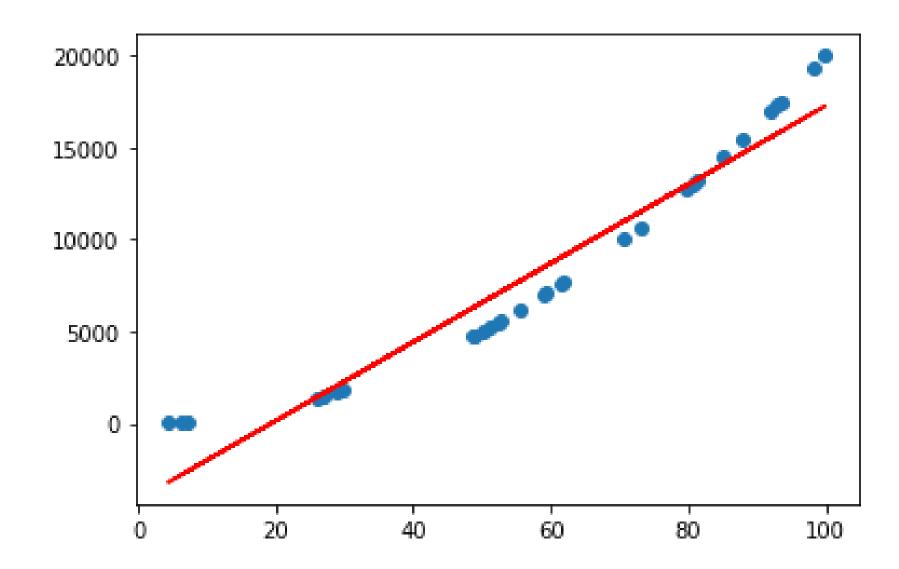
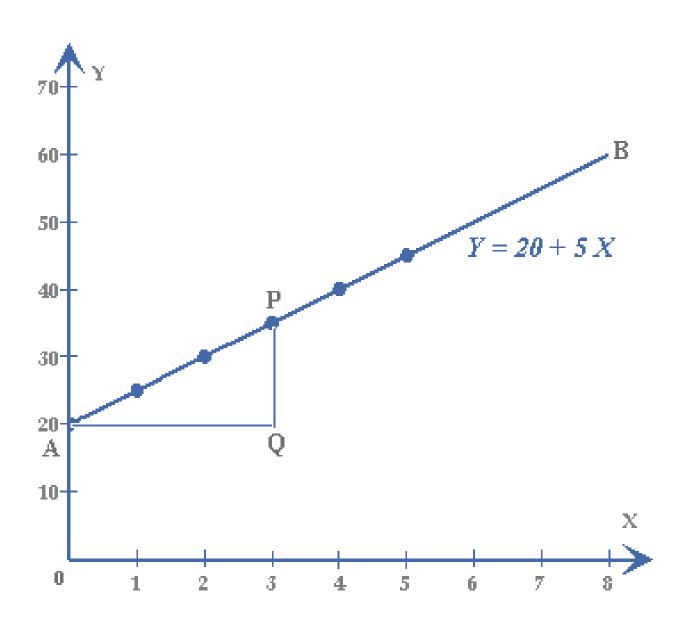


Imagem: Kaggle

### Modelo Linear



$$y = mx + b$$

Imagem: <u>emathzone</u>

# Minimização do erro

- Encontrar a combinação de pesos que minimize a RSS
- RSS: soma dos quadrados dos resíduos
- Fórmula fechada para uma variável independente
- Estimação por gradiente descendente

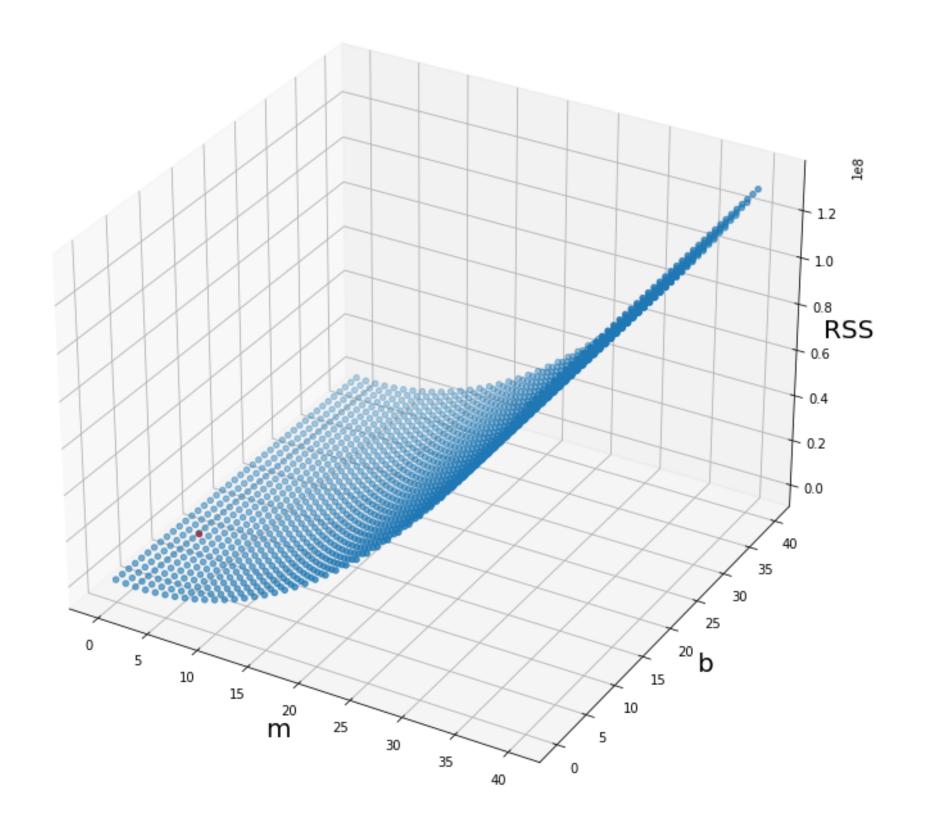


Imagem: <u>kaggle</u>

## Documentação

### sklearn.linear\_model.LinearRegression

Examples using sklearn.linear\_model.LinearRegression:
Principal Component Regression vs Partial Least Squares...

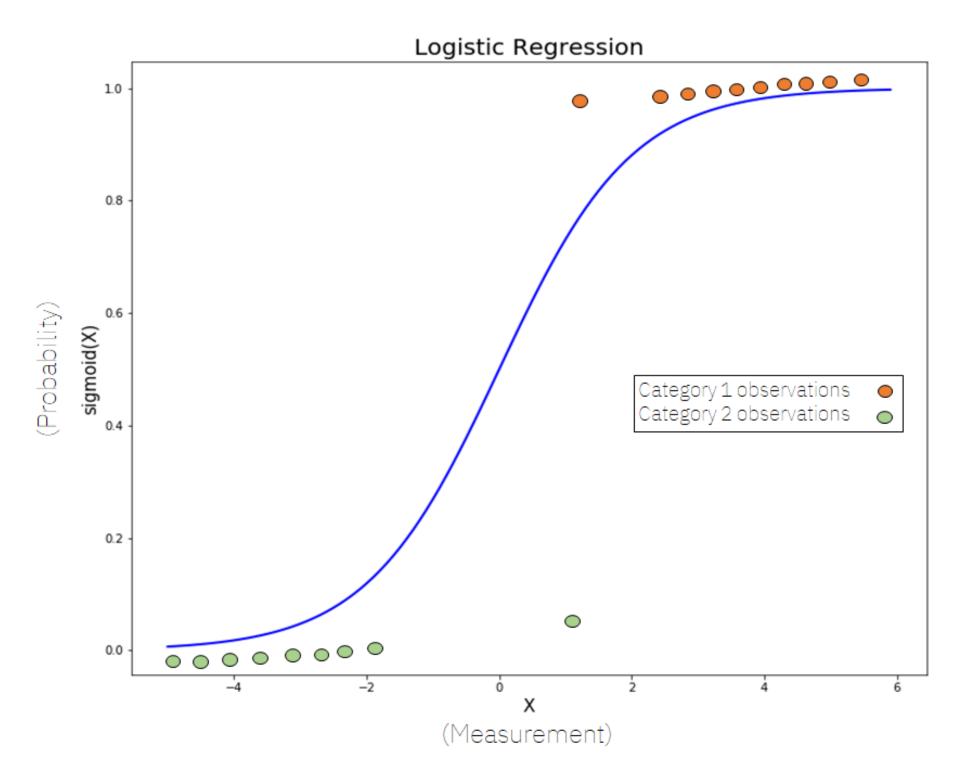


# Código

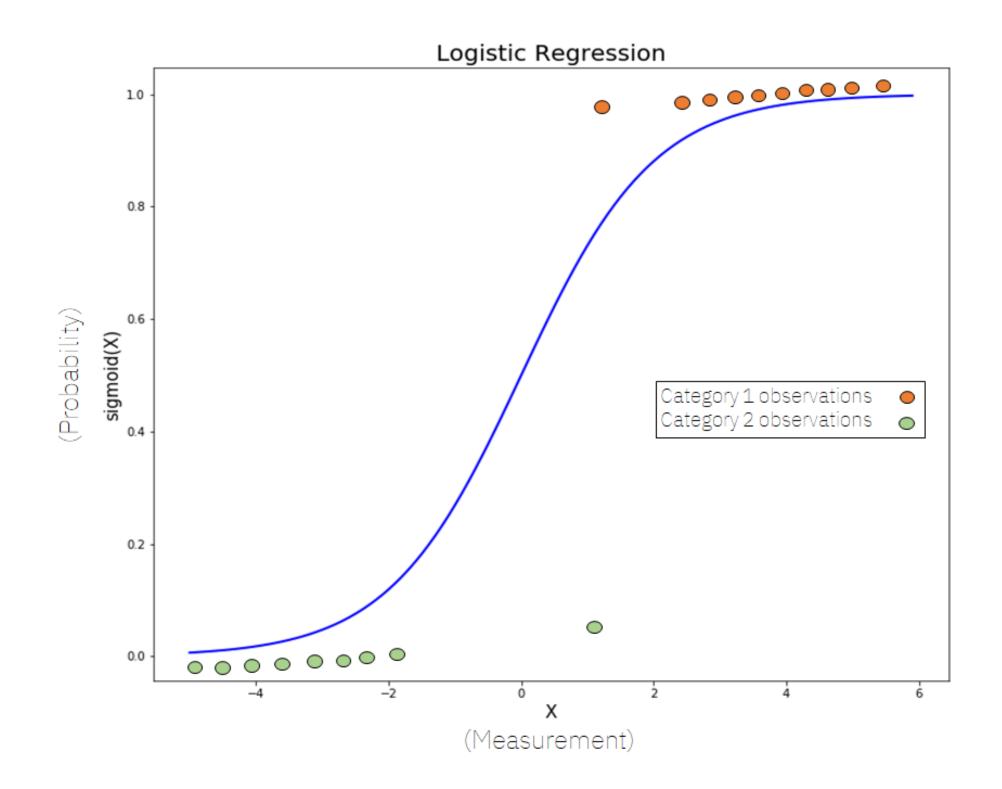


# Regularizações L1 e L2

# Regressão Logística



# Regressão Logistica



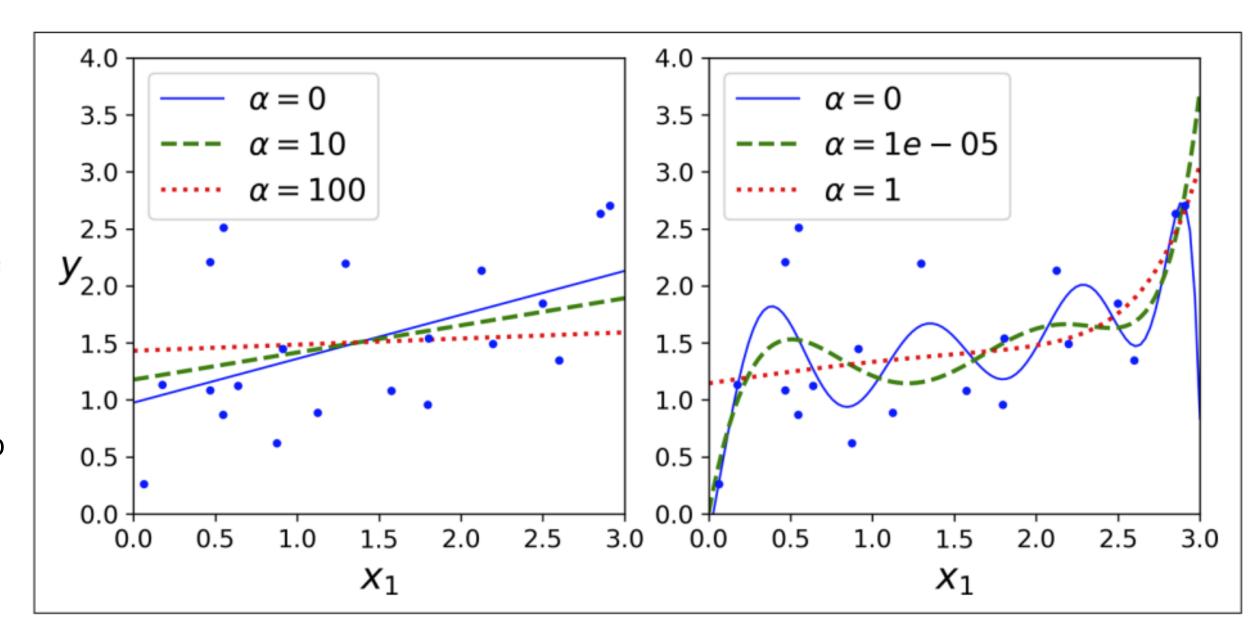
$$f(x)=rac{1}{1+e^{-x}}$$

# Regularização em Regressões

L2 Ridge

$$LossFunction = \frac{1}{N} \sum_{i=1}^{N} (\hat{Y} - Y)^2 + \lambda \sum_{i=1}^{N} \theta_i^2$$

Alguns coeficientes se aproximam de zero

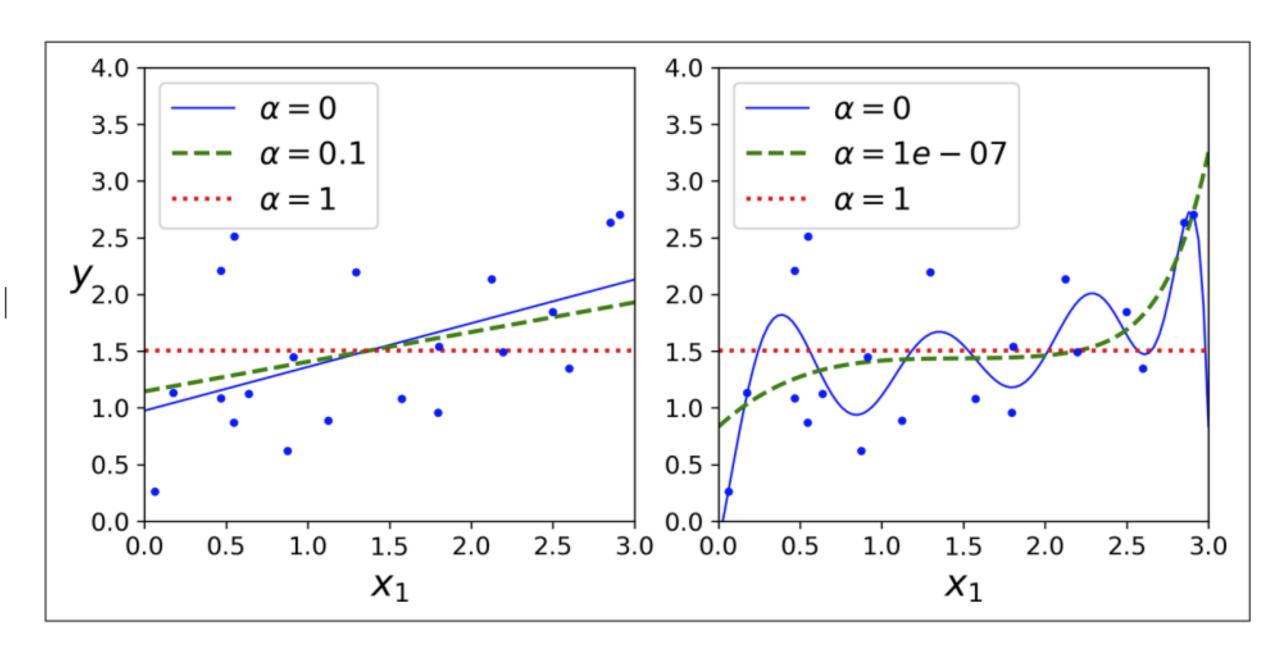


# Regularização em Regressões

L1 Lasso

$$LossFunction = rac{1}{N} \sum_{i=1}^{N} (\hat{Y} - Y)^2 + \lambda \sum_{i=1}^{N} \mid heta_i \mid$$

Alguns coeficientes se chegam a zero



## Documentação

### sklearn.linear\_model.LogisticRegression

Examples using sklearn.linear\_model.LogisticRegression: Release Highlights for scikit-learn 1.1 Release Highlights...



# Código





# APRENDIZADO DE MÁQUINA

