AI & CHATBOT

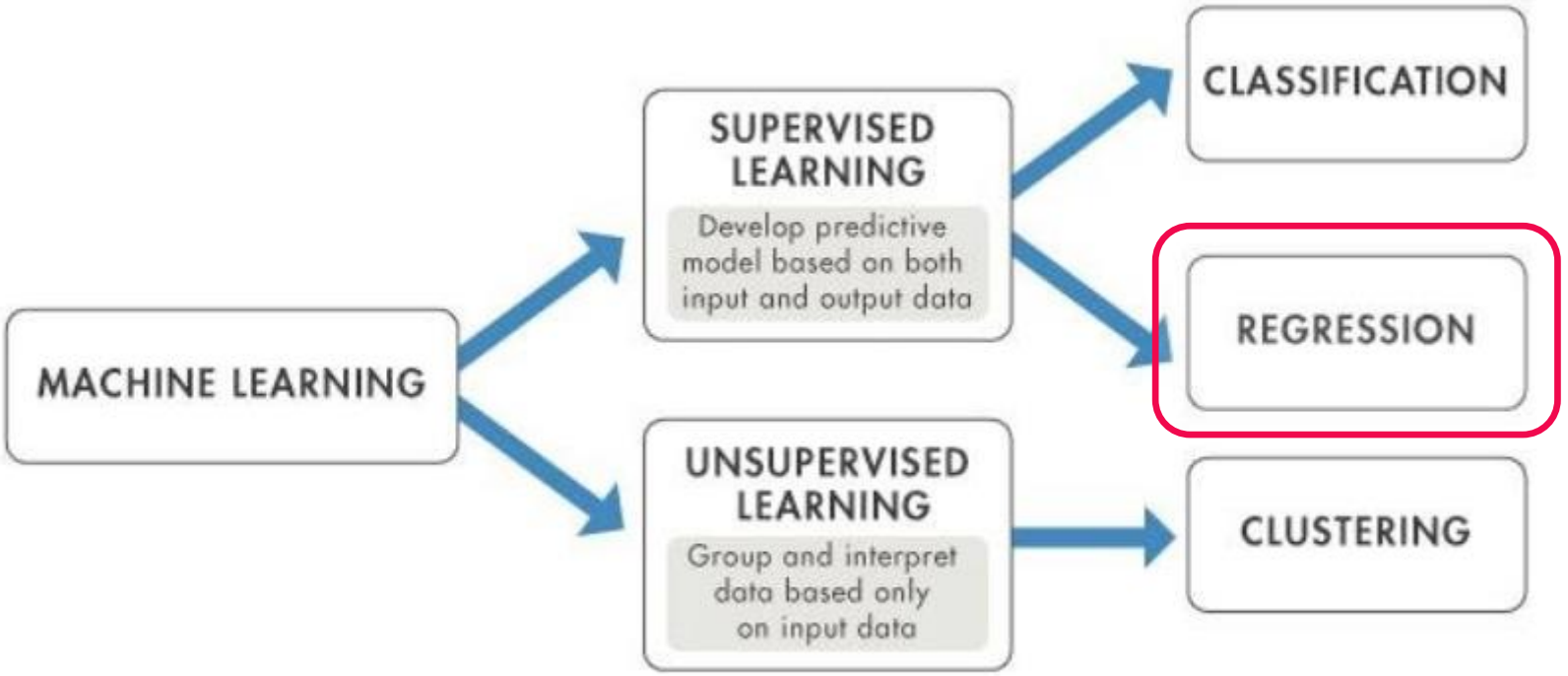
Aula 18 – Aprendizado de Máquina Supervisionado:

Algoritmos de Regressão

Prof. Henrique Ferreira
Prof. Miguel Bozer
Prof. Guilherme Aldeia
Prof. Michel Fornaciali
Prof. Daniel Gomes
Prof. Daniel Petrini
Prof. Vinicius Holanda

ML Supervisionado - Regressão

A tarefa de regressão é agrupada dentro dos algoritmos de aprendizado de máquina supervisionada.

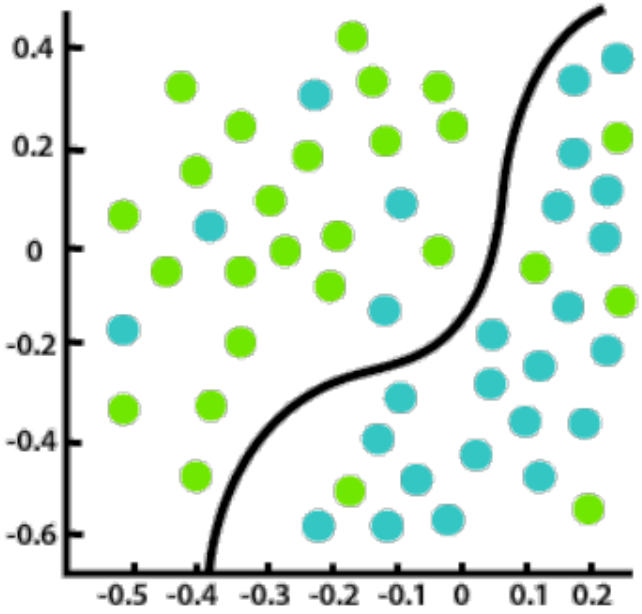


https://www.researchgate.net/profile/Rory-Bunker/publication/319937079/figure/fig1/AS:542578279366656@1506372150206/Supervised-Learning-versus-Unsupervised-Learning-Mathworks-nd.png

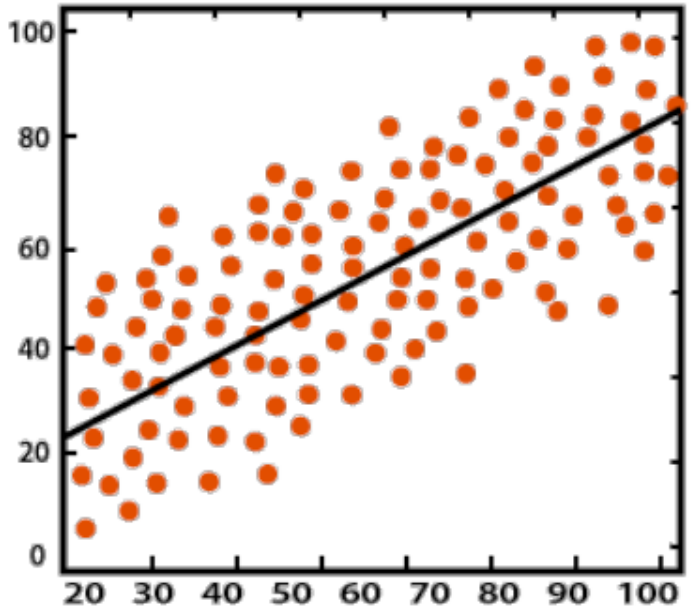
Regressão vs Classificação

- Na classificação nosso atributo alvo é um objeto (string); A curva obtida pelo algoritmo funciona como fronteira de separação de classes;
- Na regressão nosso atributo alvo é um número (float); A curva obtida pelo algoritmo funciona como uma linha de tendência dos dados;

Classificação: predição de classe

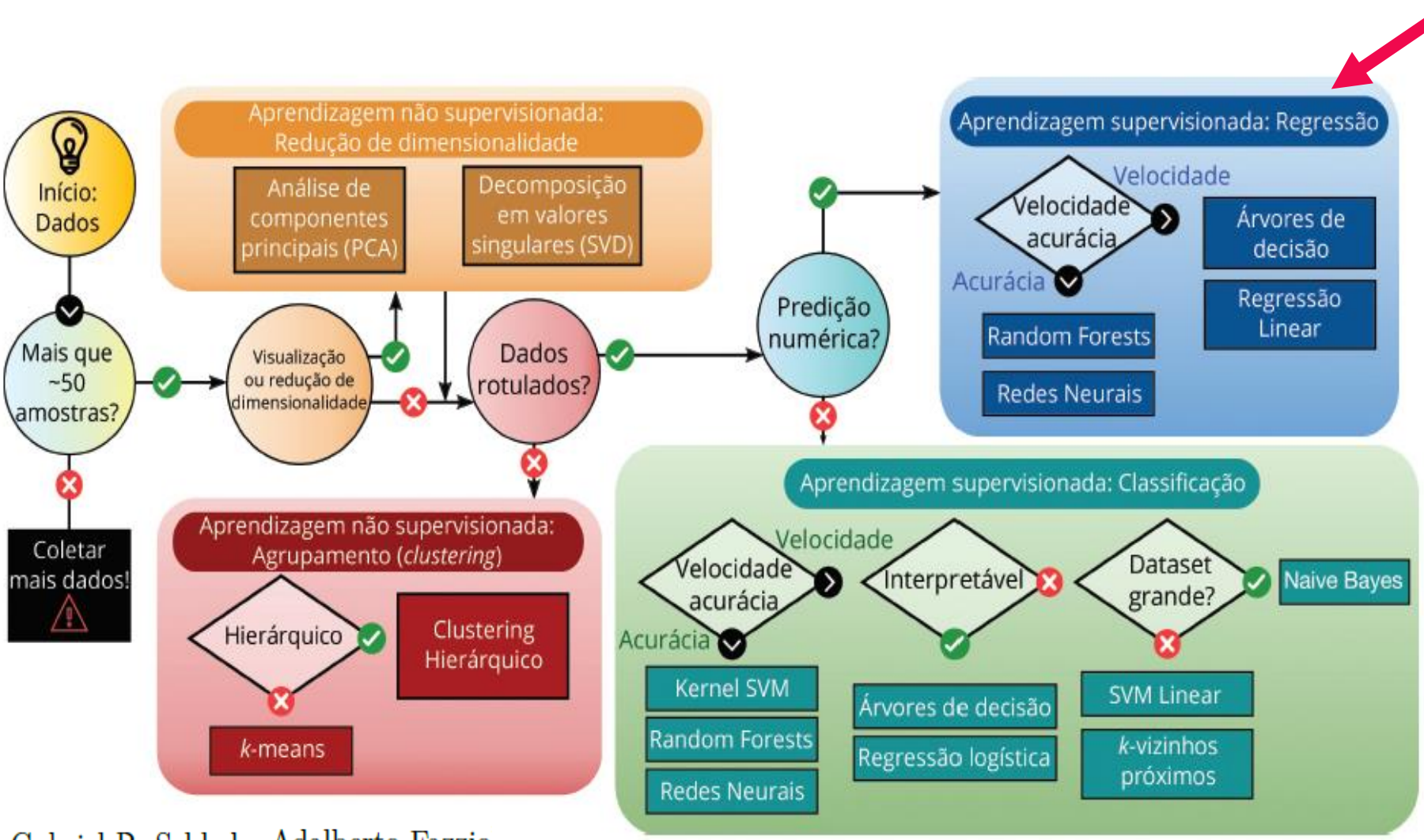


Regressão: predição de valor



https://static.javatpoint.com/tutorial/machine-learning/images/regression-vs-classification-in-machine-learning.png

Regressão



Gabriel R. Schleder Adalberto Fazzio Revista Brasileira de Ensino de Física, vol. 43, suppl. 1, e20200407 (2021)

Regressão vs Classificação

- Sempre conseguimos transformar os rótulos de **regressão** (valores) para rótulos de **classificação** (classes), definindo as faixas de valores de cada classe.

  Exemplo: valor de um imóvel à faixa de valor do imóvel

- A volta não é sempre verdadeira! Nem sempre temos informação numérica que nos permita aplicar regressão.

  Exemplo: imóvel acima ou abaixo do preço de mercado e sem a informação do valor em si.

Regressão

Um modelo matemático de regressão tem como objetivo predizer um valor de uma variável dependente através de uma função das variáveis independentes.

Atributo alvo = variável resposta (ou dependente)
Atributos descritivos = variáveis regressoras (ou independentes)

Abordagem Paramétrica:
assumimos uma classe de funções que descrevem o relacionamento entre as variáveis regressoras e de resposta;

Abordagem Não Paramétrica:
não assumimos nenhuma classe de função a rigor, deixando o algoritmo regredir a função sozinho (funcionando como uma caixa preta);

Regressão

FIAP

A regressão, diferentemente da classificação, opera sobre dados numéricos $\mathbb{R}^N$.

Matematicamente, a regressão tentar estimar a função $f(x_1, x_2, \ldots, x_N)$ que relaciona os atributos $x_1, x_2, \ldots, x_N$ com um atributo alvo y.

Uma forma de realizar isso automaticamente é realizando minimização da função erro (ou função custo) $E(f(x), y)$.

A regressão é uma tarefa supervisionada, que opera sobre um conjunto de k exemplos;

Lembrando que exemplos são as linhas na tabela de dados e atributos são as colunas.

Regressão

Regressão é uma técnica supervisionada!

$$f(x_1, x_2, \ldots, x_N)$$

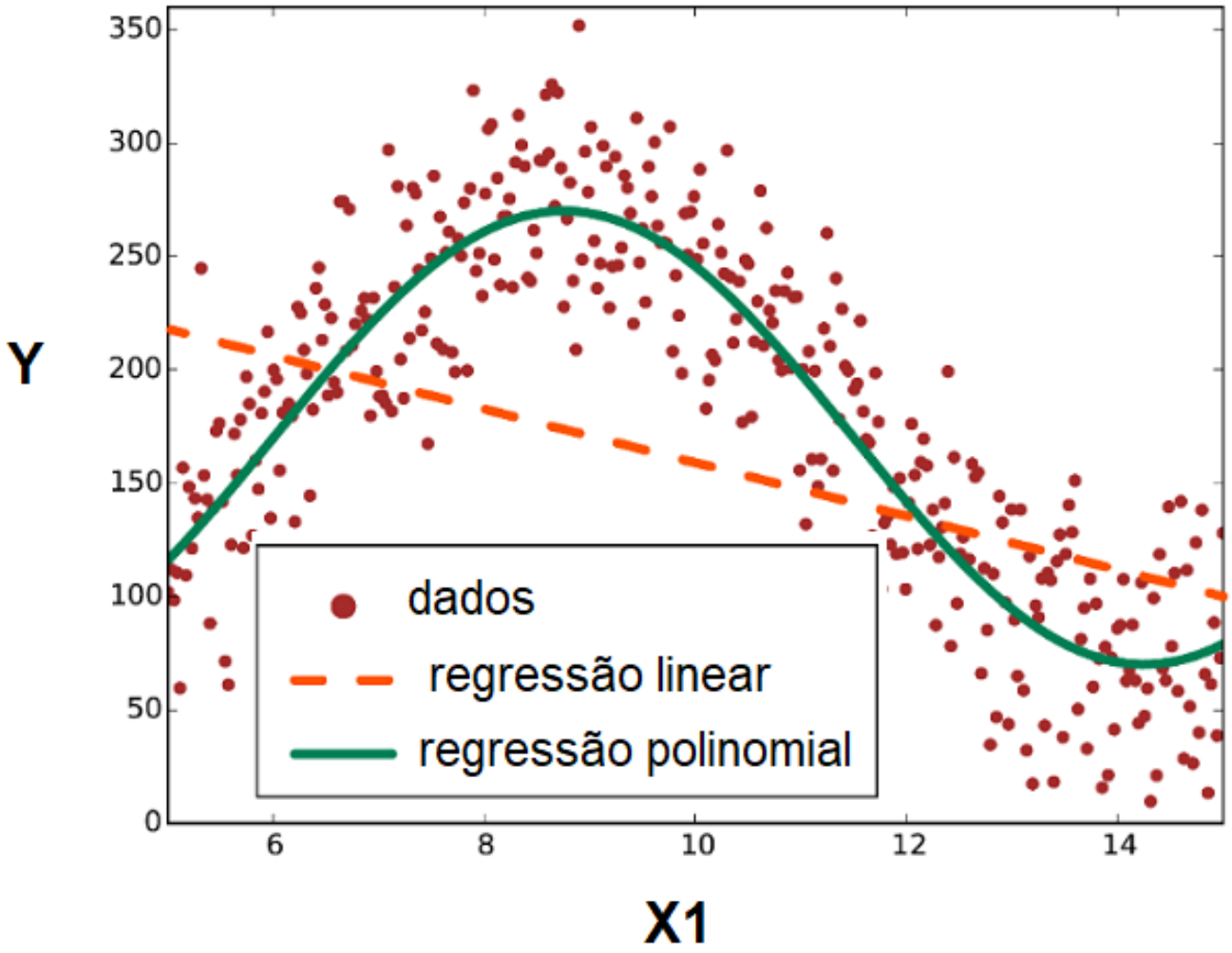
Atributos descritivos

Atributo alvo

Índice da linha	x1	x2	...	xn	y
1	548.4	-9789		0.4875	-7595.28
2	689.4	-10235		-0.358	-7468.82
3	3154.8	-1031858	...	-0.1458	-1019232
...	...			...	
k	803.54	-20000		1.054	-16791.4

Regressão

O objetivo da regressão é encontrar a regra que relaciona as variáveis, mostrando uma “linha de tendência”:

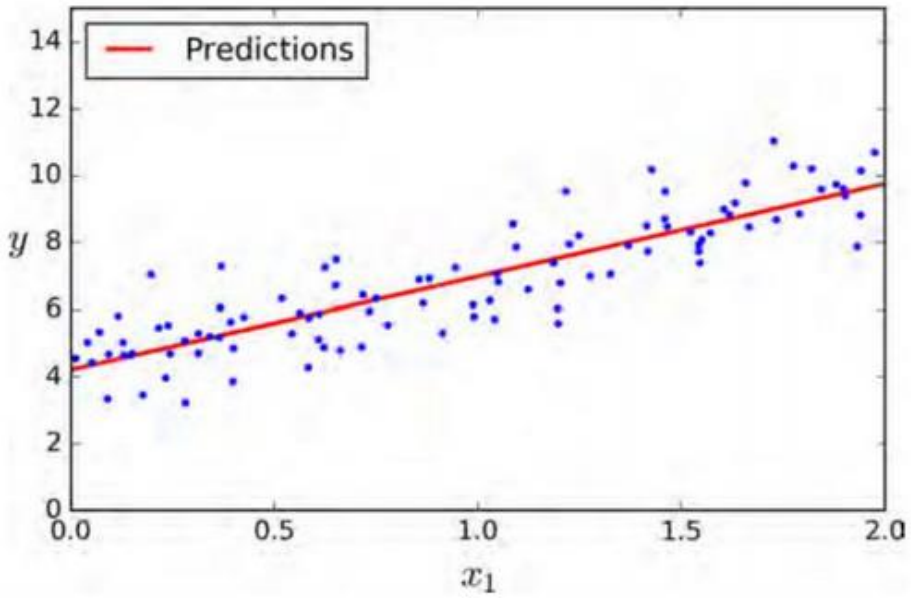


ALGORITMOS DE
DE REGRESSÃO

Regressão Linear (paramétrica)

Função linear:

$$f(x) = \hat{y} = ax + b$$



O algoritmo de regressão pretende determinar quais são os coeficientes a e b do modelo matemático.

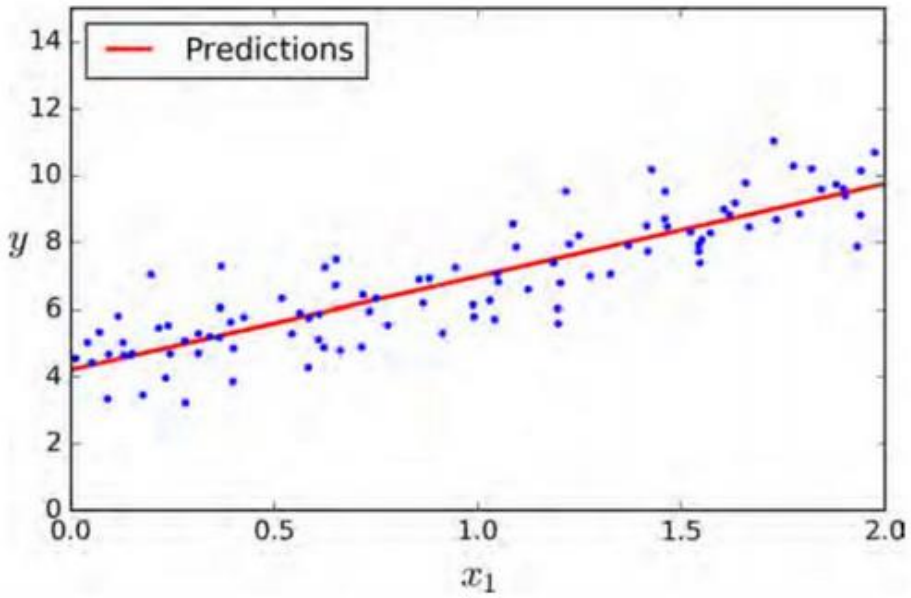
Uma vez conhecidos esses valores, o modelo pode generalizar a saída y para uma data entrada arbitrária x.

Como funciona?

FIAP

Função linear:

$$f(x) = \hat{y} = ax + b$$



O algoritmo de regressão pretende determinar quais são os coeficientes a e b do modelo matemático.

Uma vez conhecidos esses valores, o modelo pode generalizar a saída y para uma data entrada arbitrária x.

Como funciona?

Chuta valores para os **coeficientes da equação** e então calcula o erro médio obtido para os dados de treinamento. Altera os valores dos coeficientes e recalcula o erro. Repete esse passo até chegar em um erro mínimo aceitável.

Exemplo - queremos determinar uma função de uma variável que é linear:

$$f(x) = \hat{y} = ax + b$$

Treinar o modelo significa encontrar os valores de a e b

			Erro
1	0	1	-1
5	8	5	3
...	...	...	...
10	18	10	8

Chutamos $a = 1$ e $b = 0$

$$MSE = \frac{(-1)^2 + (3)^2 + \cdots + 8^2}{N}$$

Regressão Linear (paramétrica)

FIAP

- Nosso problema pode envolver muitas features (colunas), ou seja, diferentes variáveis de entrada x_i.
- O modelo de Regressão Linear então deverá ter mais coeficientes para cada coluna de entrada;
- Ele pode ser entendido como a combinação linear das entradas:

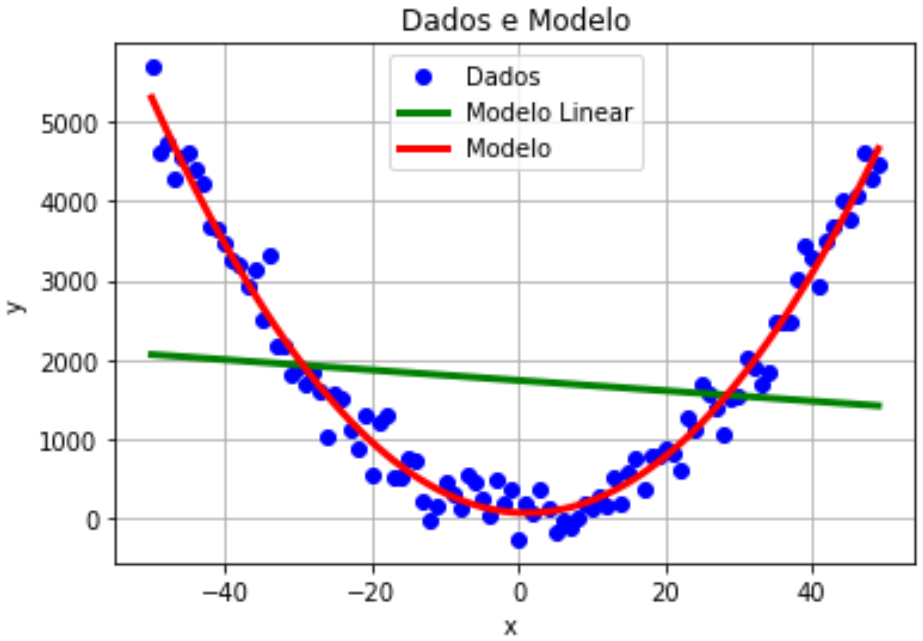
$$f(x_1, x_2, x_3, \dots, x_n) = a_1x_1 + a_2x_2 + \cdots + a_nx_n + a_0$$

- O algoritmo de regressão pretende determinar quais são os coeficientes a_i que produzem o menor erro médio em cima dos dados de treinamento.

Regressão - Quadrática

Função do segundo grau:

$$f(x) = ax^2 + bx + c$$



O algoritmo de regressão pretende determinar quais são os coeficientes a, b e c do modelo matemático.

Uma vez conhecidos esses valores, o modelo pode generalizar a saída y para uma data entrada arbitrária x.

Regressão - Polinomial

Polinômio de grau i:

$$f(x) = a_i x^i + a_{i-1} x^{i-1} + a_{i-2} x^{i-2} + \cdots + a_1 x + a_0$$

É possível realizar a regressão de qualquer série de dados através de um polinômio de grau elevado;
Isso é possível pois, dado que existe uma função $f(x)$ infinitamente diferenciável, podemos aproximá-la através de uma série de Taylor:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n.$$

Regressão – Mudança de variável

FIAP

- Sempre é possível criar um algoritmo para outras funções gerais.
- Funções polinomiais podem ser reduzidas em funções lineares de varias variáveis;
- Na prática é como se houvesse uma nova coluna com o valor de x^2;

Uma variável x

$$f(x) = ax^2 + bx + c$$

$$x^2 = x_1$$
$$x = x_2$$

Substituição de variáveis

Duas variáveis x_1 e x_2

$$f(x_1, x_2) = ax_1 + bx_2 + c$$

Regressão – Outras funções

FIAP

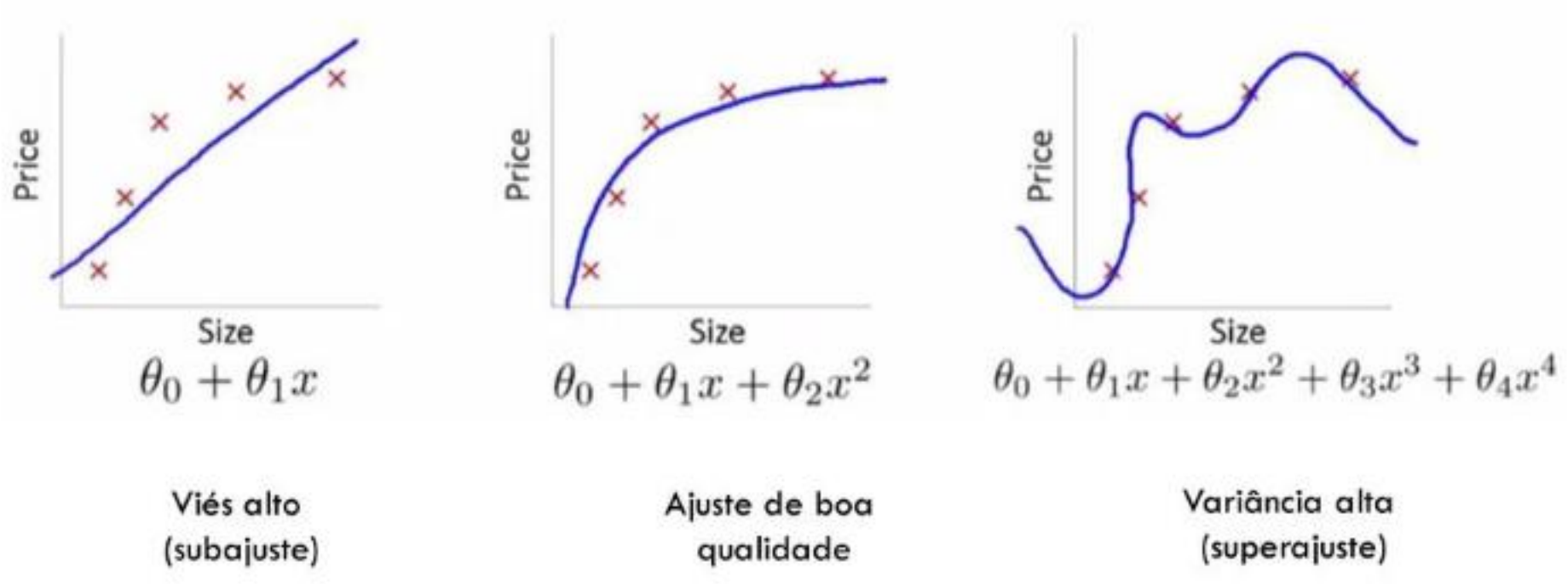
Usando a abordagem paramétrica você pode testar toda uma vasta classe de funções:

$$f(x) = a\frac{1}{x} + be^{x-1} - \sqrt{2x} + \sin(2x)\log_{10}(3x+2) + \cdots$$

- Se você quer modelos simplificados e com explicações, vale a pena estudar funções diferentes para saber como seus dados se comportam realmente;
- Existem algoritmos avançados chamados de **Regressão Simbólica** que tentam encontrar a melhor função que descreve os dados;
- Apesar de ser possível aproximar qualquer função por uma série infinita, como a série de Taylor ou a série de Fourier, essas aproximações podem não corresponder a coeficientes observáveis e acabam gerando overfitting;

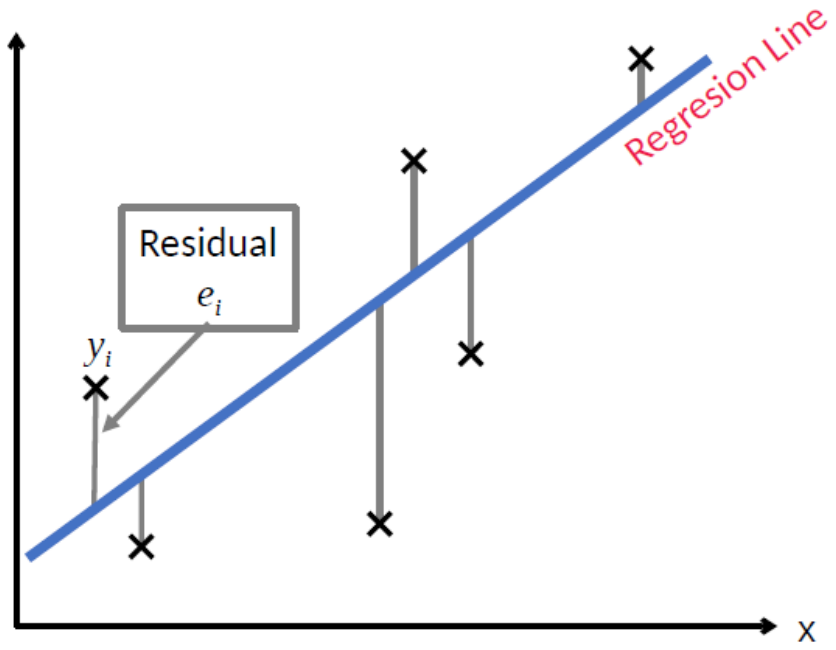
Regressão – Overfitting

Um modelo com **overfitting** tem mais coeficientes do que o necessário. É um modelo com pouca capacidade de generalização: ele terá alta acurácia para os dados de treinamento e acurácia extremamente baixa para os dados de teste.



Regressão – Função Custo

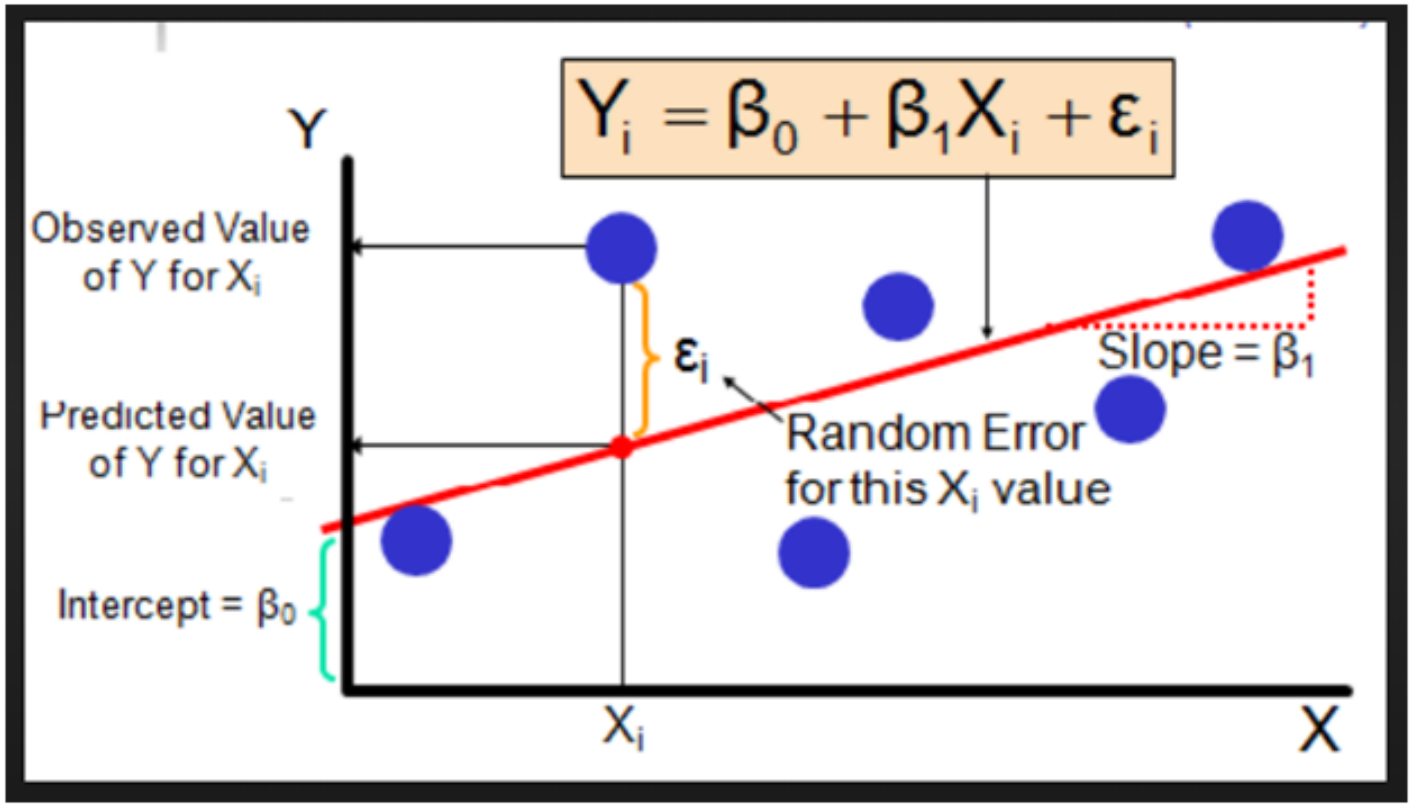
Função custo ou função erro é uma função $E(f(x), y)$ escolhida para estimar a distância entre o valor real de y e o valor previsto pelo modelo $\hat{y} = f(x)$.



Existem vários tipos de função custo. O objetivo do algoritmo é minimizar o resultado dessa função custo, minimizando assim o erro entre o modelo e o dado real.

Regressão – Função Custo

Por exemplo, para uma função linear:



LinearRegression

Regressão – Função Custo

- Erro quadrático médio (MSE);
- Erro absoluto médio (MAE);
- Distância euclidiana média;
- Distância máxima absoluta na direção y;
- Distância euclidiana máxima;
- Erro absoluto percentual médio (MAPE);
- Erro percentual médio (MPE);
- Menor erro absoluto (LAE);
- Erro percentual absoluto médio simétrico (SMAPE);

Cada função escolhida pode ter um domínio de aplicação específico que permite o algoritmo performar melhor;
Em todos os casos, estamos interessados em um problema de otimização: reduzir o valor dessa função.

Regressão – Função Custo

Soma dos erros absolutos:

$$SEA = \sum_{i=1}^{k} |y_i - \hat{y}_i|$$

Soma dos quadrados dos erros:

$$SQE = \sum_{i=1}^{k} (y_i - \hat{y}_i)^2$$

OBS: Famoso método dos mínimos quadrados, visto em cálculo numérico.

Regressão – Otimização

FIAP

- A otimização é o processo automático de minimizar os erros residuais, isto é, minimizar a função custo assumida;
- Existem vários algoritmos de otimização diferentes;
- Algoritmos de otimização são toda uma área da computação e da IA, cujos resultados geram muitas aplicações de impacto econômico e de engenharia;

- Exemplos de algoritmos de otimização para reduzir a função custo são:
  - Descida do gradiente (Gradient Descent - GD)
  - Mínimos quadrados (Ordinary least squares - OLS)
  - Método de Adams
  - Decomposição e valores singulares

Regressão – MMQ

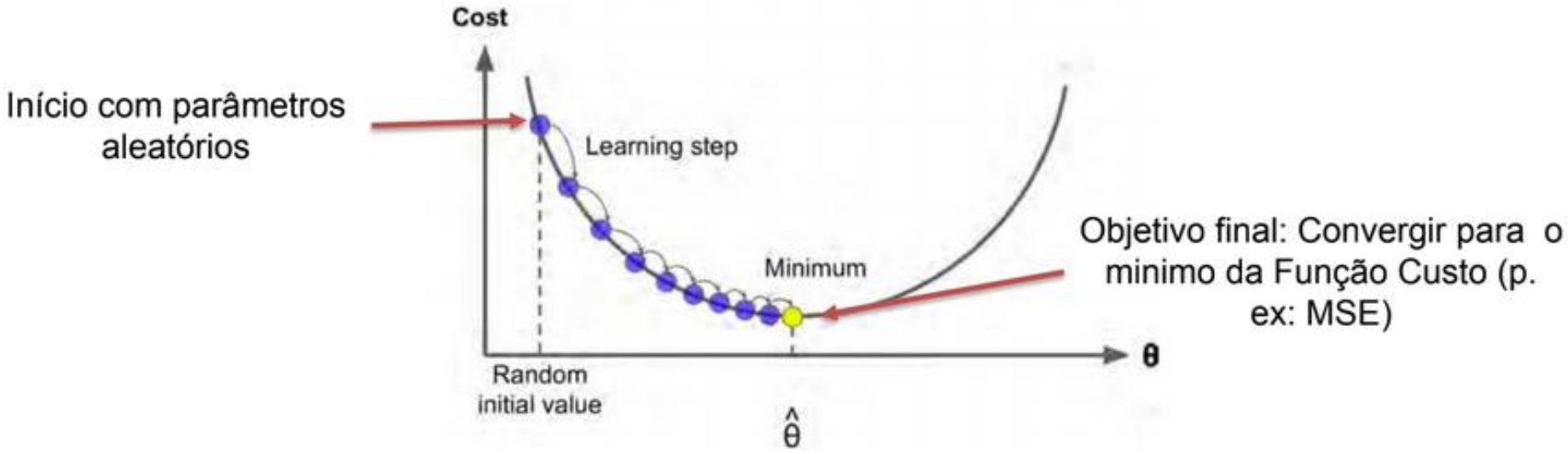
- O método dos mínimos quadrados (MMQ) é um dos métodos mais usados para realizar regressões;
- A função custo nesse caso é a soma dos quadrados dos erros;
- O objetivo é encontrar os coeficientes de um equação (método paramétrico) que minimize o valor da função custo;
- Quando a função avaliada é linear, existe um método analítico de solução (caso especial de otimização), o que torna o método muito mais rápido;
- Entretanto, nem sempre é possível inverter as matrizes da equação paramétrica associada, e o problema torna-se computacionalmente mais difícil;

$$SQRes(\boldsymbol{\theta}) = \sum_{i=1}^{n} [y_i - f(\mathbf{x}_i, \boldsymbol{\theta})]^2$$

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X^2 + \ldots + \beta_k X^k + \varepsilon$$

Regressão – Gradiente Descendente

É um algoritmo de otimização iterativo que pode ser aplicado a diversos problemas. A ideia central é utilizar informações sobre os gradientes da função para determinar a próxima iteração do algoritmo;

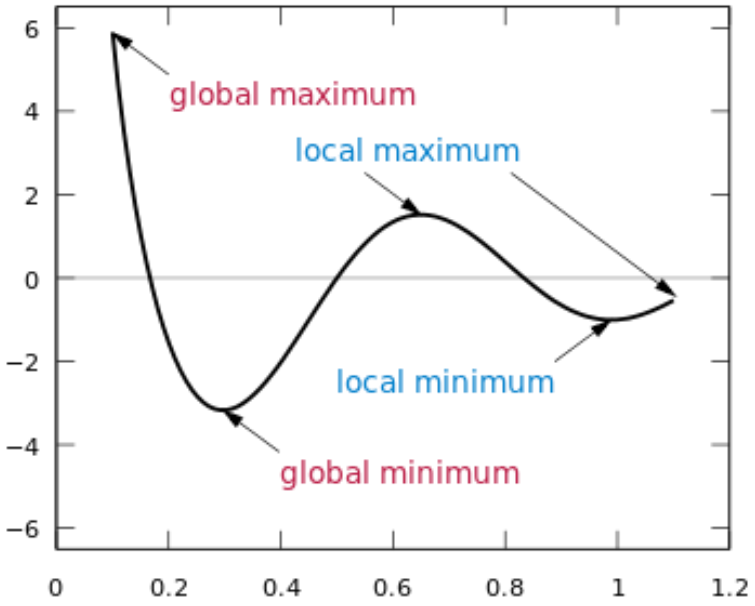


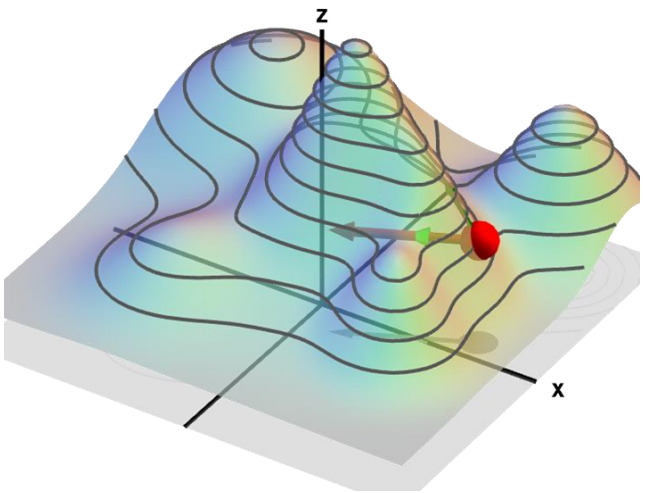
Regressão – Gradiente Descendente

Máximo ou mínimo da função $\frac{d}{d\theta} f(\theta) = 0$

Operador gradiente:

$$\nabla = \left(\frac{\partial}{\partial \theta_1}, \frac{\partial}{\partial \theta_2}, \cdots, \frac{\partial}{\partial \theta_k} \right)$$

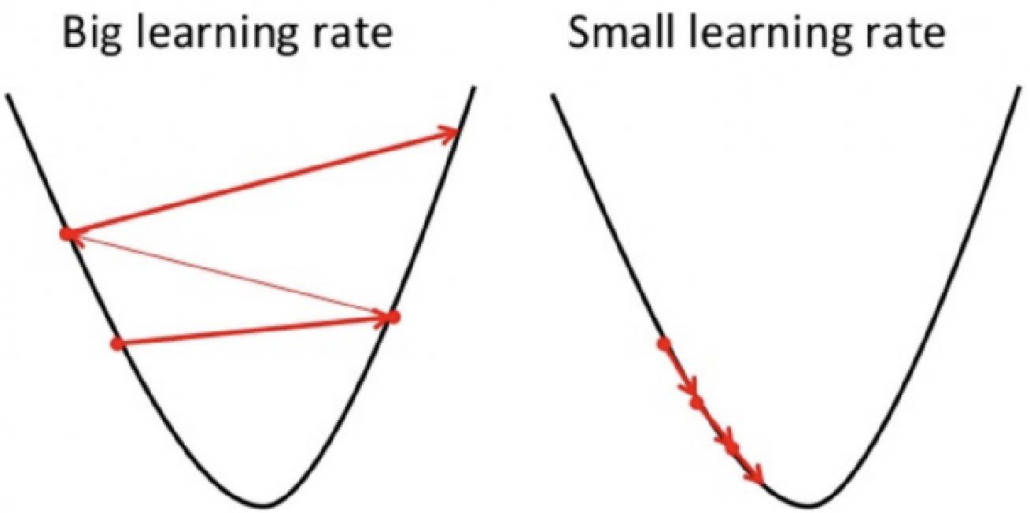




Regressão – Gradiente Descendente

O tamanho do incremento usado para varrer o espaço de busca é o learning rate. O learning rate é um exemplo de hiperparâmetro do algoritmo de Aprendizado de Máquina;

- Se ele for muito pequeno: o algoritmo deverá iterar muitas vezes para convergir (algoritmo lento);
- Se ele for muito grande: o algoritmo pode não convergir (rápido mas não funciona);



Regressão – Gradiente Descendente

FIAP

S.NO.	Gradient Descent	Normal Equation
1.	In gradient descenet , we need to choose learning rate.	In normal equation , no need to choose learning rate.
2.	It is an iterative algorithm.	It is analytical approach.
3.	Gradient descent works well with large number of features.	Normal equation works well with small number of features.
4.	Feature scaling can be used.	No need for feature scaling.
5.	No need to handle non-invertibility case.	If (X^TX) is non-invertible , regularization can be used to handle this.
6.	Algorithm complexity is O(kn^2). n is the number of features.	Algorithm complexity is O(n^3). n is the number of features.

https://www.geeksforgeeks.org/difference-between-gradient-descent-and-normal-equation/

Regressão – Ridge e Lasso

Regularização L1 e L2

- Na regularização de L1, tentamos minimizar a função custo adicionando um termo de penalidade à soma dos valores absolutos dos coeficientes.
- Na regularização de L2, tentamos minimizar a função custo adicionando um termo de penalidade à soma dos quadrados dos coeficientes.

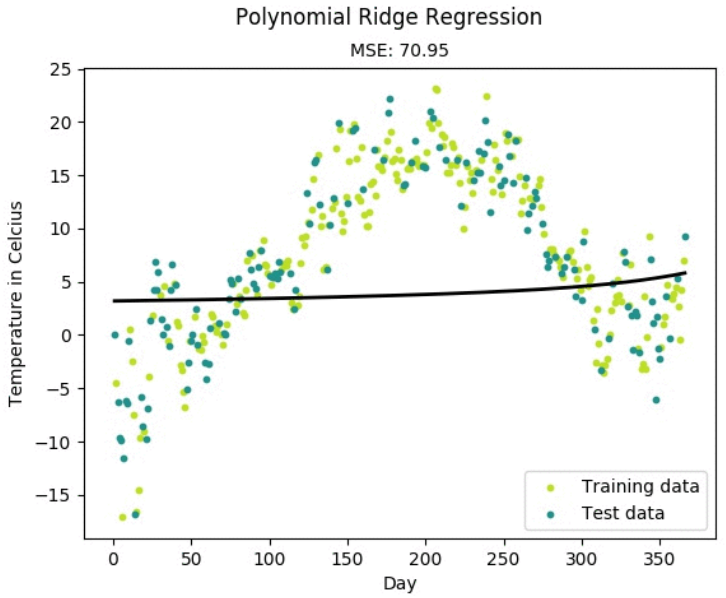
Regressão – Regressão Ridge

Na função custo da regressão linear, tentamos minimizar a soma dos quadrados dos erros.

Na **Regressão Ridge**, adicionamos uma restrição na soma dos quadrados dos coeficientes de regressão por meio de um parâmetro de regularização que chamamos de **L2**. Assim a função custo fica como:

$$Min(\sum \varepsilon^2 + \lambda\beta^2) = Min \sum (Y - (\beta_1 + \beta_2 X_2 + \beta_3 + \ldots + \beta_k X_k))^2 + \lambda \sum \beta^2$$

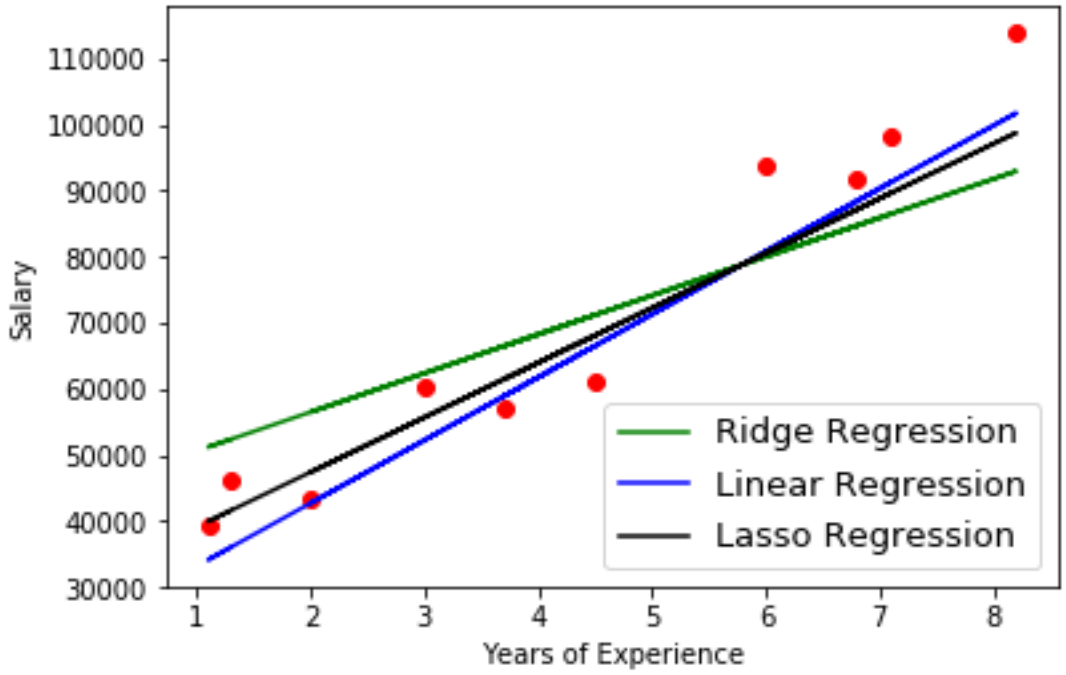
λ é um número não negativo chamado de parâmetro de regularização;



Regressão – Regressão Lasso

Lasso significa **Least Absolute Shrinkage and Selection Operator** . Faz uso da técnica de regularização L1, assim como a regressão Ridge na função objetivo. Assim, a função objetivo na regressão LASSO torna-se:

$$Min(\sum \varepsilon^2 + \lambda \sum |\beta|) = Min \sum (Y - (\beta_1 + \beta_2 X_2 + \beta_3 + \ldots + \beta_k X_k))^2 + \lambda \sum |\beta|$$



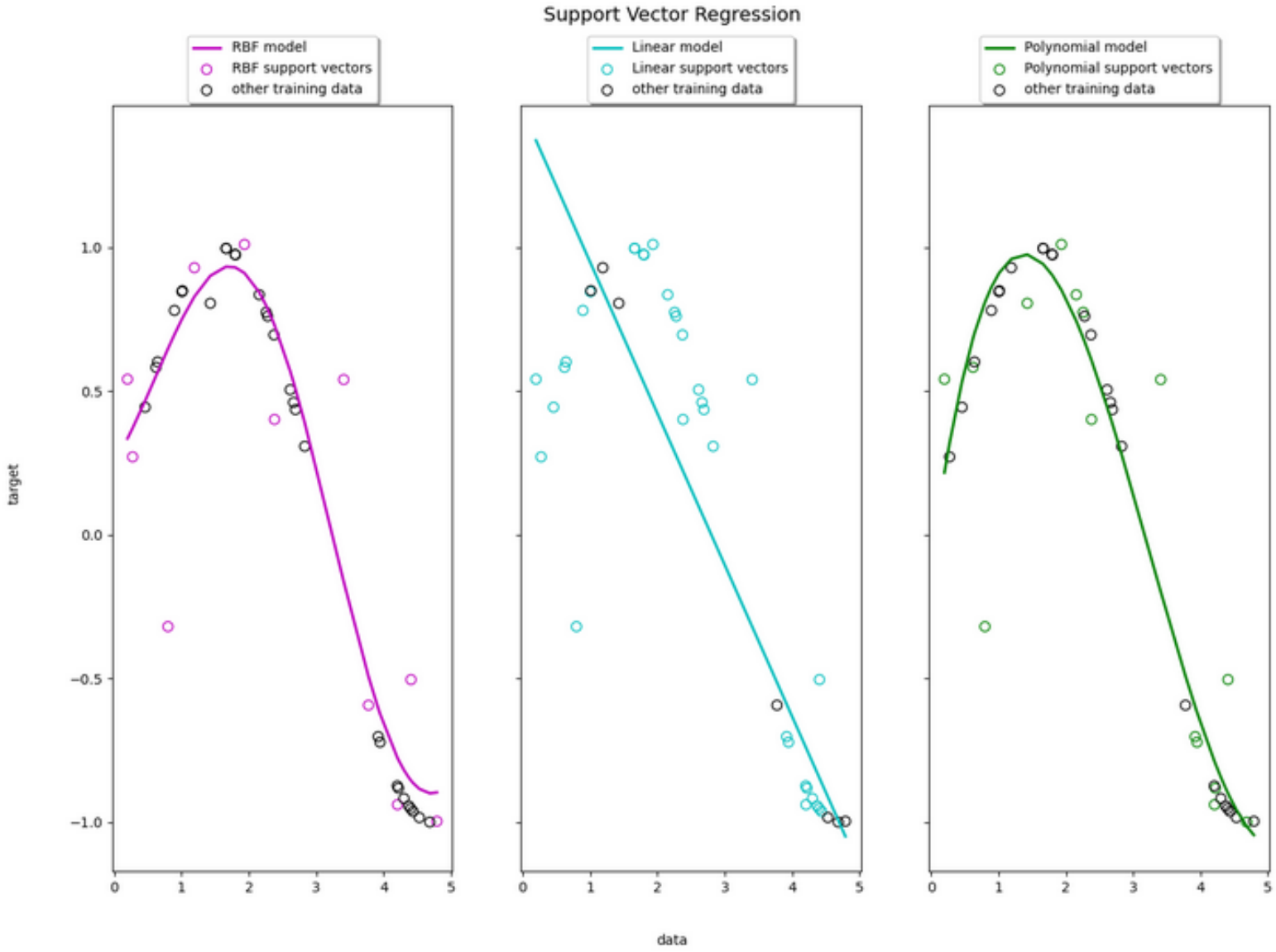
Regressão SVR

FIAP

- É possível adaptar o método de classificação SVM para criar um método de regressão (Support Vector Regression – SVR);
- No Sklearn existem 3 implementações desse método: SVR, LinearSVR e NuSVR;
- A vantagem de usar esses métodos é que devido ao truque de kernel é possível regredir funções não lineares;

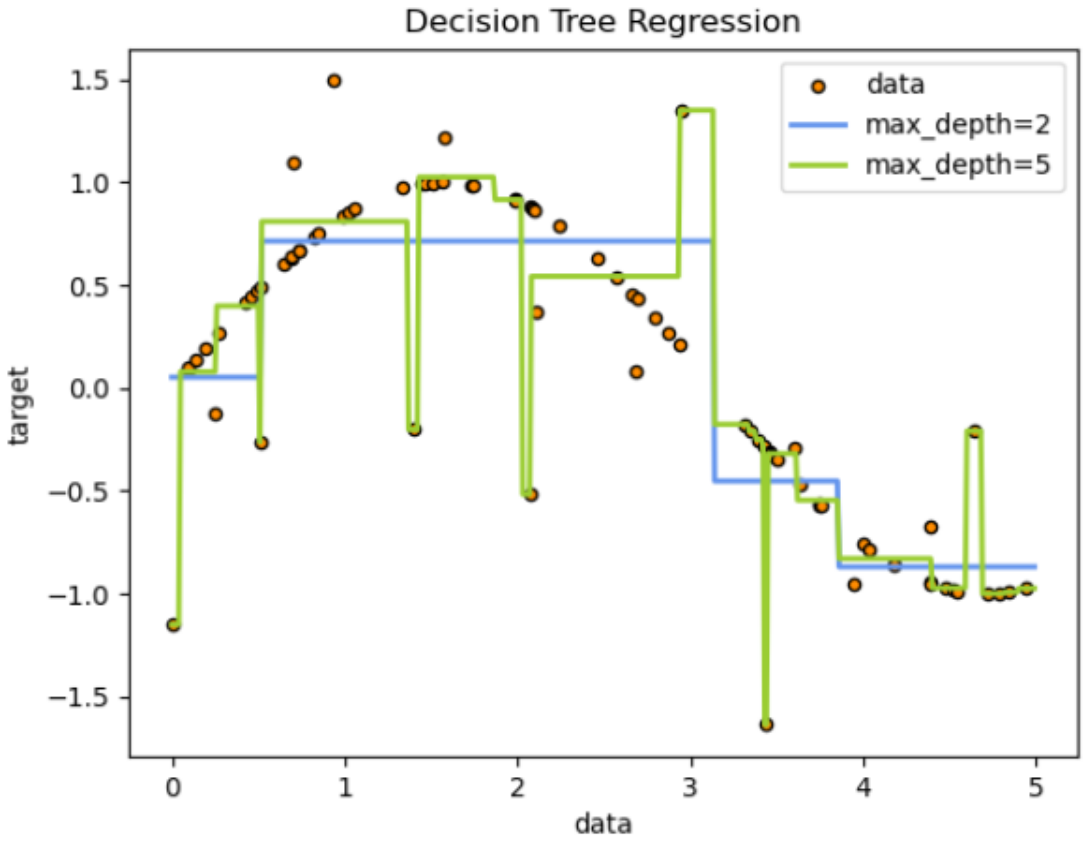
https://scikit-learn.org/stable/modules/svm.html#svm-regression

Regressão SVR



Árvores de Decisão

- Também é possível adaptar o método de Árvore de Decisão para realizar a tarefa de regressão;
- As árvores de decisão são métodos não paramétricos;



Árvores de Decisão

FIAP

Vantagens:

- Modelo caixa-branca: fácil de entender e de interpretar (as árvores podem ser visualizadas);
- Requer pouca preparação dos dados;
- O tempo computacional é logarítmico com o número de exemplos de treinamento $O(\log n)$;
- Consegue resolver problemas de múltiplos outputs;
- É possível validar com analises estatísticas;

Desvantagens:

- Overffiting: modelos tem pouca capacidade de generalização (preciso usar métodos de poda e limites para a profundidade da árvore para eliminar isso);
- São instáveis a ruído (dados com alta variância e outliers);
- Não são bons extrapoladores pois aproximam as funções por retas constantes;
- Dataset precisa estar balanciado para não gerar bias;
- Funções XOR, paridade e problemas de multiplexação podem ser difíceis de serem resolvidos;

MÉTRICAS DE DESEMPENHO DE REGRESSÃO

Desempenho de Regressão

FIAP

- Na regressão a resolução do problema envolve predizer um valor numérico $\hat{y}$ a partir de dados de treinamento $(x_1, x_2, x_3, \ldots, x_n, y)$. Para realizar isso, o algoritmo pode encontrar um modelo matemático expresso por $f(x_1, x_2, \ldots, x_n)$. Lembrando que, durante a estimação da função f o algoritmo usa as informações do y de treinamento.
- Com os dados de teste devemos realizar uma análise de desempenho do algoritmo de regressão treinado;

- Algumas métricas possíveis:
  - Erro médio quadrático;
  - Erro absoluto médio;
  - Coeficiente de determinação R^2;
  - Coeficiente de correlação de Pearson;

Regressão – Função Custo

Erro absoluto médio (Mean Absolute Error – MAE):

$$MAE = \frac{1}{k}\sum_{i=1}^{k}|y_i - \hat{y}_i|$$

Erro quadrático médio (Mean Squared Error – MSE):

$$MSE = \frac{1}{k}\sum_{i=1}^{k}(y_i - \hat{y}_i)^2$$

OBS: Famoso método dos mínimos quadrados, visto em cálculo numérico.

Coeficiente de Determinação R²

- Baseado nos quadrados dos resíduos (erros);
- R² = 0.76 significa que o modelo de regressão linear explica 76% da variância de y a partir de $(x_1, x_2, x_3, \dots, x_n)$;

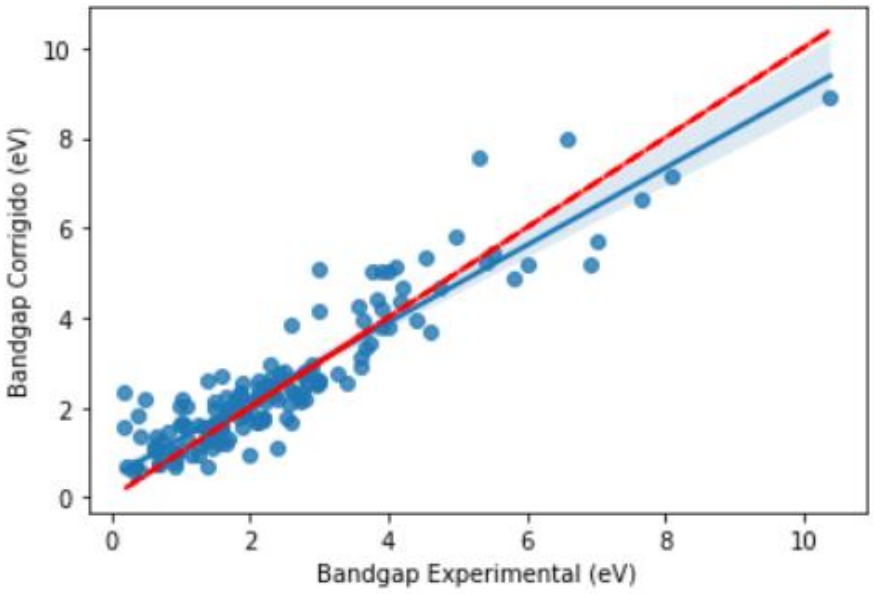
Soma Total dos Quadrados $$SQ_{tot} = \sum_{i=1}^{k} (y_i - \bar{y})^2$$

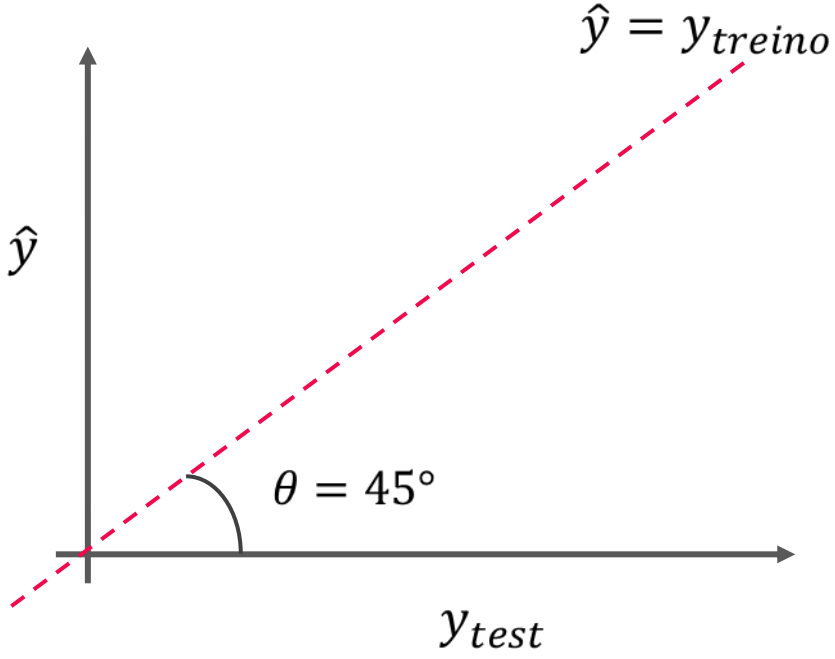
Soma dos Quadrados dos Resíduos $$SQ_{res} = \sum_{i=1}^{k} (y_i - \hat{y}_i)^2$$

Coeficiente de determinação $$R^2 = 1 - \frac{SQ_{res}}{SQ_{tot}}$$

Quanto mais próximo de 1, melhor!

Gráfico $\hat{y}$ vs y_{test}





Coeficiente de Pearson ρ

FIAP

- Baseado na correlação entre dados. Pressupõem uma dependência linear, por isso devemos aplicá-lo usando os dados previstos e reais, isto é, $(y, \hat{y})$

Coeficiente de Pearson

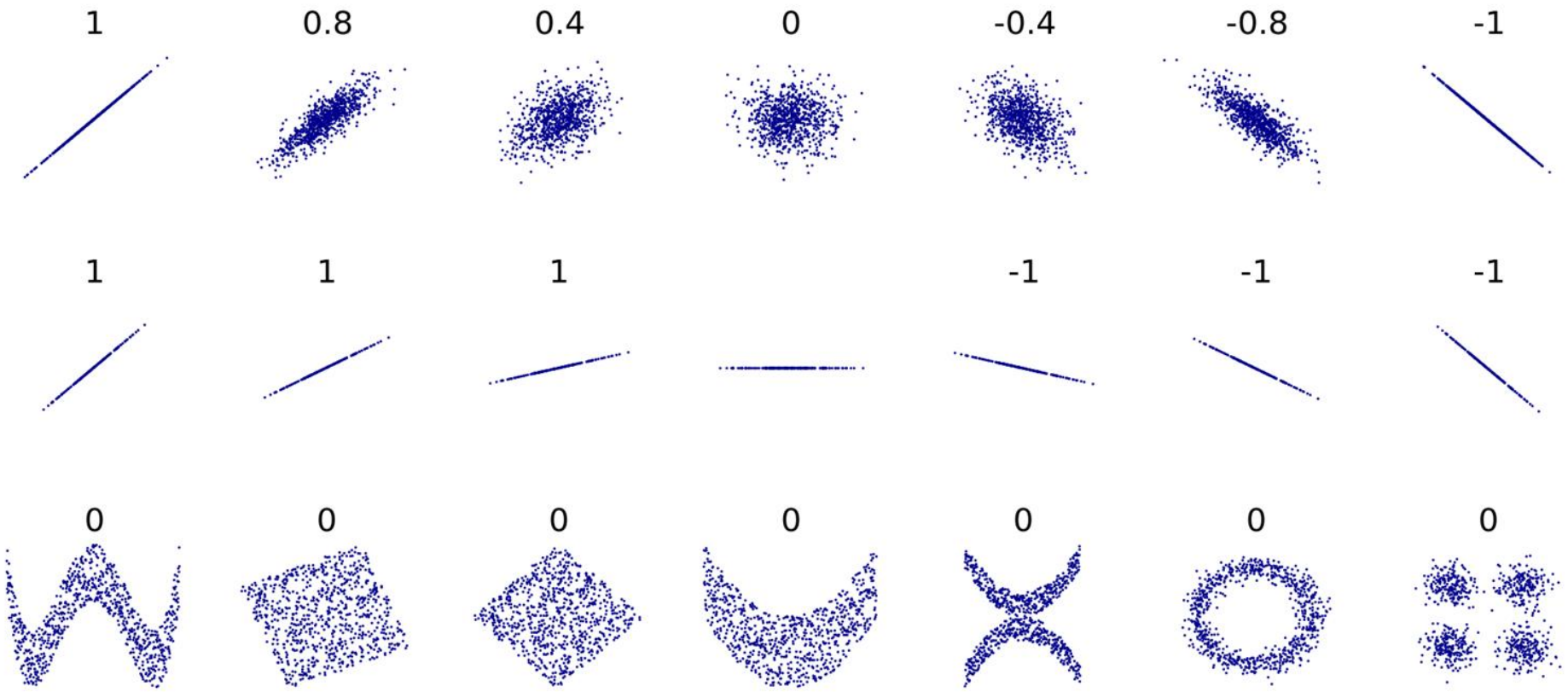
$$\rho = \frac{cov(X,Y)}{\sqrt{var(X)var(Y)}}$$

$$\operatorname{cov}(X,Y) = \frac{1}{n}\left[\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)\right]$$

$$\operatorname{Var}(X) = \sum_{i=1}^{n} p_i \cdot (x_i - \mu)^2,$$

Quanto mais próximo de 1, melhor!

Coeficiente de Pearson ρ



Coeficiente de Pearson

$$\rho = \frac{cov(X,Y)}{\sqrt{var(X)var(Y)}}$$

Na prática:

FIAP

- Em Python já existem bibliotecas prontas para se calcular essas métricas;
- O scikit-learn traz módulos internos com funções prontas para isso:

https://scikit-learn.org/stable/modules/model_evaluation.html

```
1 mean_absolute_error(y_test, y_pred)
```

```
1 mean_squared_error(y_test, y_pred)
```

Regression	
'explained_variance'	metrics.explained_variance_score
'max_error'	metrics.max_error
'neg_mean_absolute_error'	metrics.mean_absolute_error
'neg_mean_squared_error'	metrics.mean_squared_error
'neg_root_mean_squared_error'	metrics.mean_squared_error
'neg_mean_squared_log_error'	metrics.mean_squared_log_error
'neg_median_absolute_error'	metrics.median_absolute_error
'r2'	metrics.r2_score
'neg_mean_poisson_deviance'	metrics.mean_poisson_deviance
'neg_mean_gamma_deviance'	metrics.mean_gamma_deviance
'neg_mean_absolute_percentage_error'	metrics.mean_absolute_percentage_error

FIAP GRADUAÇÃO