

Curso de Especialização em Aprendizagem de Máquina em Inteligência Artificial

Disciplina: Aprendizagem de Máquina

AULA 03

Prof. Gustavo Gattass Ayub





Revisão

Aula Passada

- Regressão Linear Múltipla
- Regressão vs Classificação
- Regressão Logística
- Conjunto de Treinamento e Testes
- Matriz de Confusão
- Curva ROC

Exercício de Aprofundamento

Exercício Complementar - **RECOMENDADO** (não vale nota)

- [Introduction to Logistic Regression](#)
- [Building a Logistic Regression in Python, Step by Step](#)

Exercício de Aprofundamento (vale nota)

- Vamos utilizar o mesmo dataset de vinhos tintos para criar um classificador. O dataset categoriza os vinhos em 6 classes de qualidade (3-8). Para criar um classificador binário, você deve considerar que as notas (3-6) indicam “Baixa Qualidade” e (7-8) indicam “Alta Qualidade”
- Você deve particionar o dataset em dois conjuntos: treinamento e validação usando a proporção 80-20. Tome o cuidado de manter um bom balanço em termos de exemplos nos dois conjuntos.
- Você deve treinar o classificador utilizando o algoritmo da Regressão Logística. Você também deve produzir uma matriz de confusão aplicando esse classificador ao conjunto de validação.
- Dataset: arquivo .csv “Wine Dataset” (no moodle – Aula 02)



*All models are wrong,
but some are useful.*

George Box

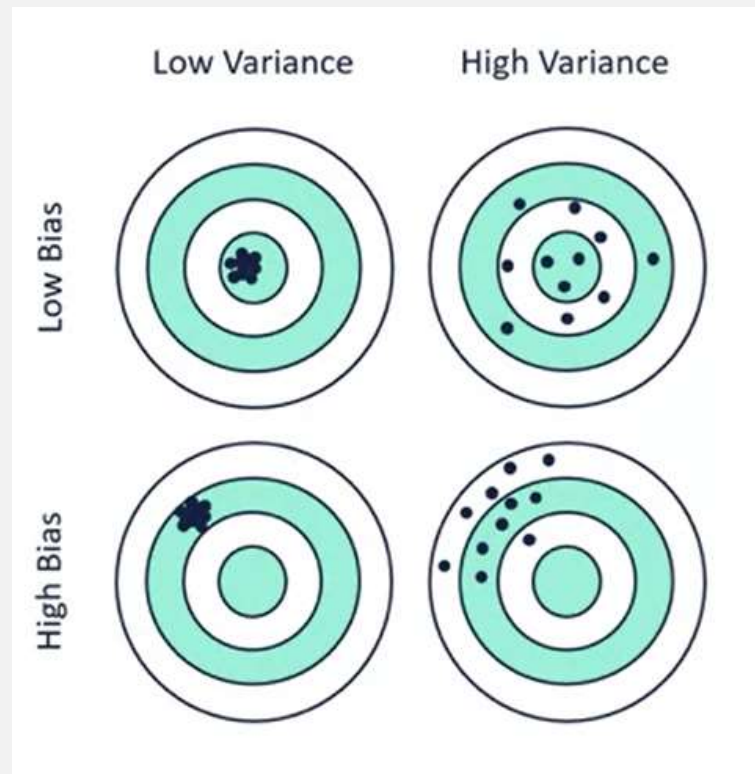
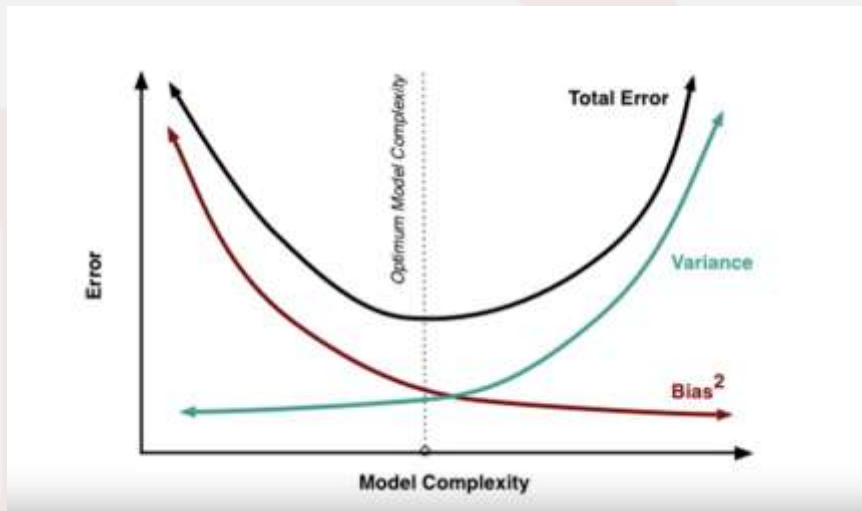
Desafios do Treinamento



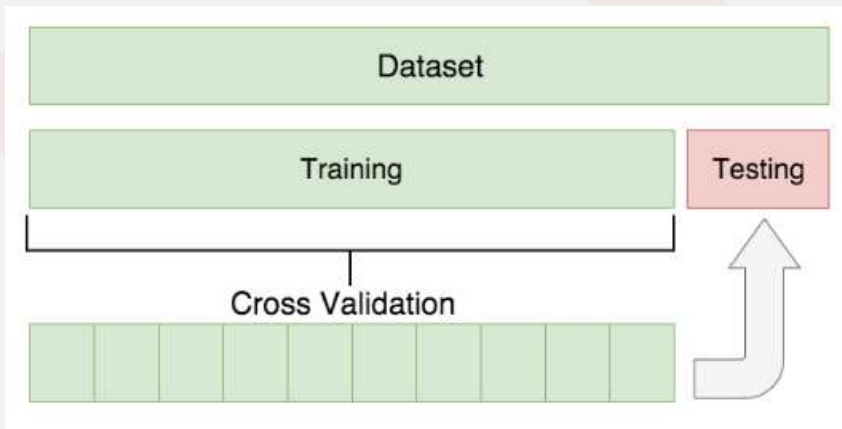
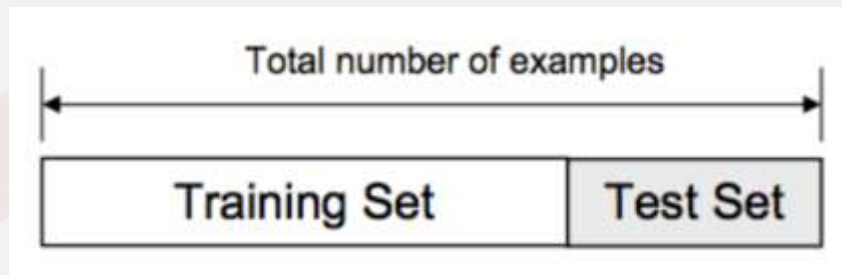
- Qualidade dos Dados
- Conjuntos de Treinamento e Validação
- Preparação dos Dados

Lidando com Desvios e Variâncias no processo

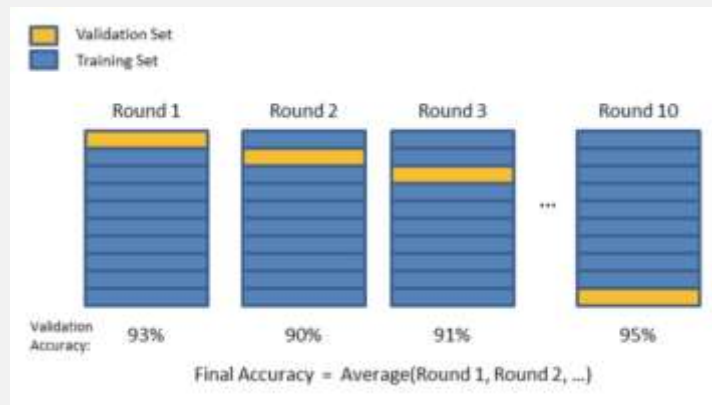
- Desvios
 - Conjunto de Treinamento
 - Modelo



Conjuntos de Treinamento e Validação

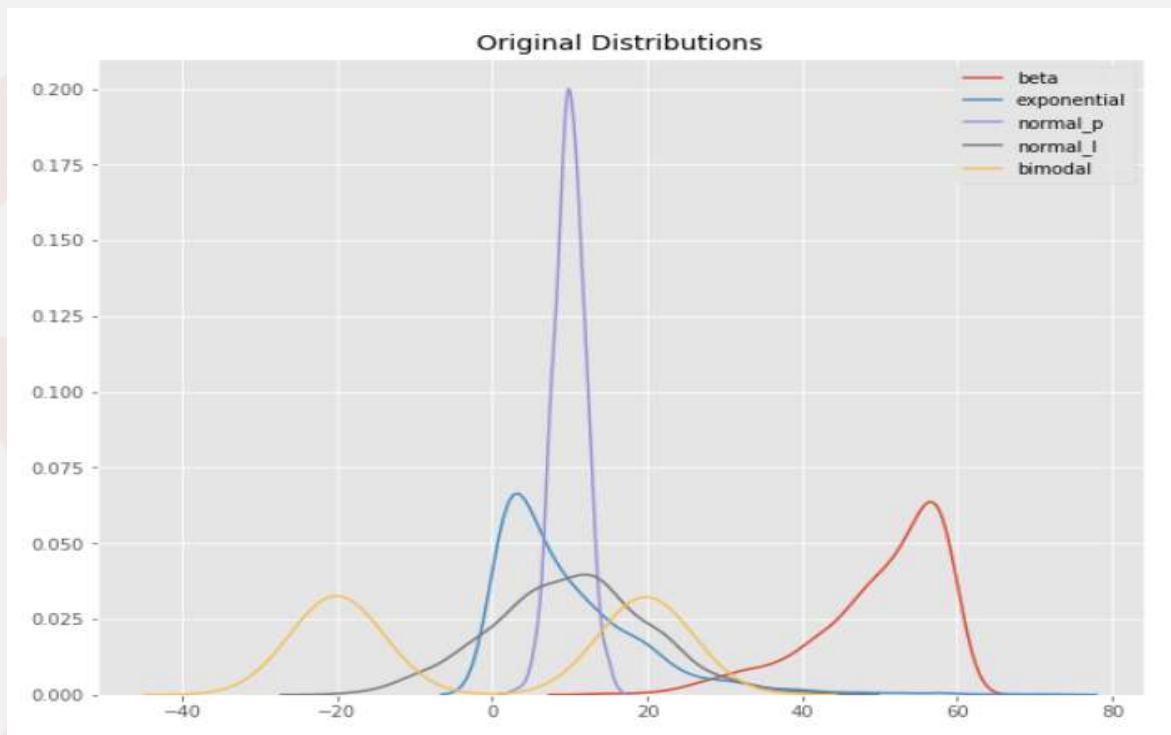


K-Folds Cross Validation

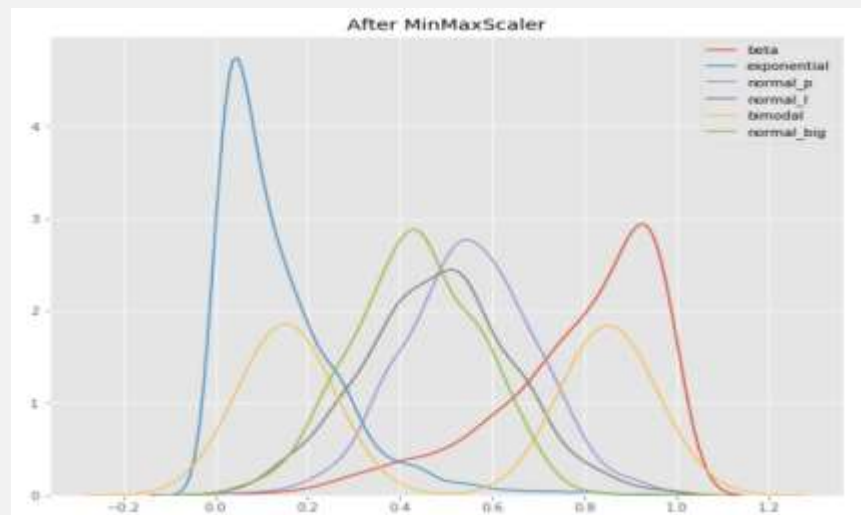
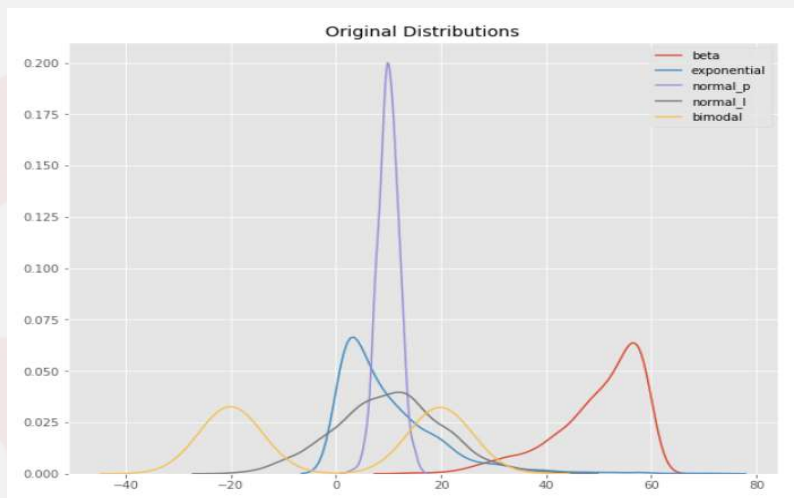


Fonte: [Train/Test Split and Cross Validation in Python](#)

Scale, Standardize

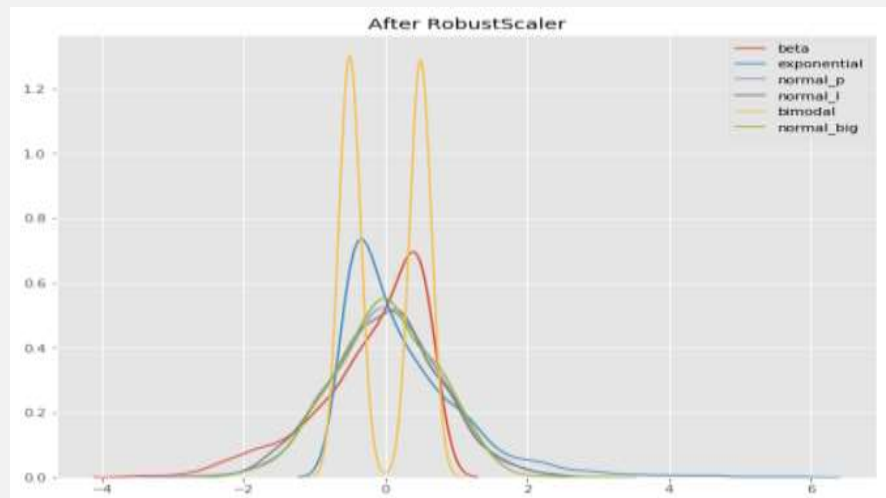
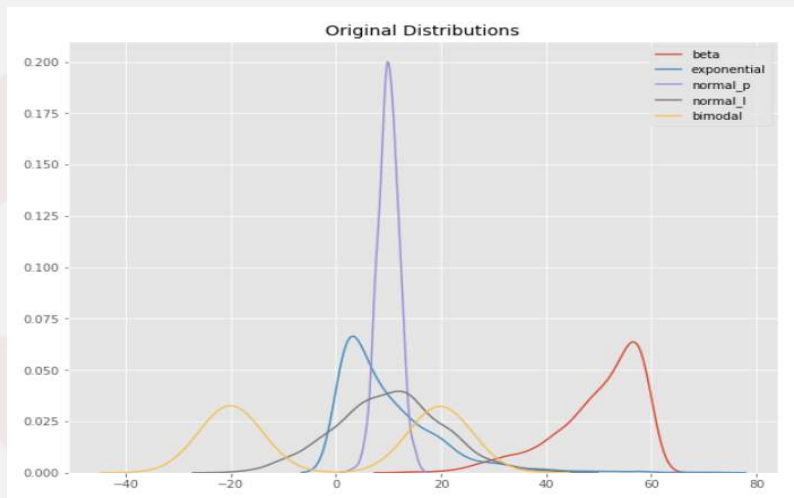


MinMaxScaler



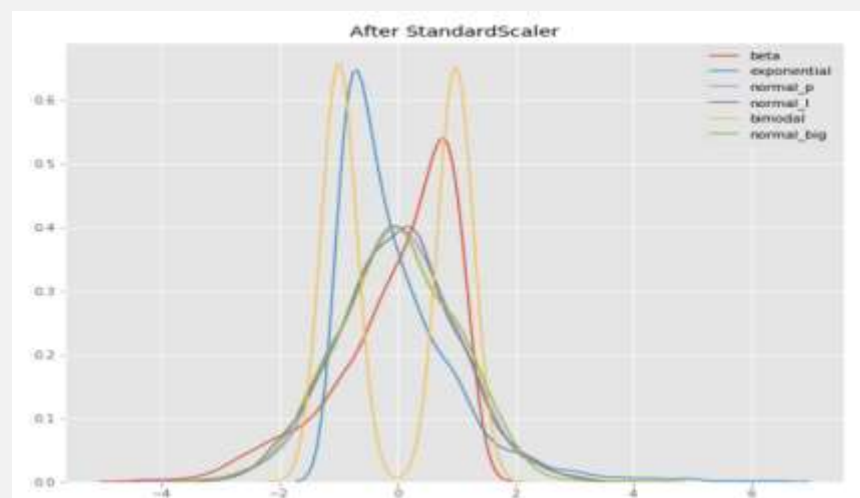
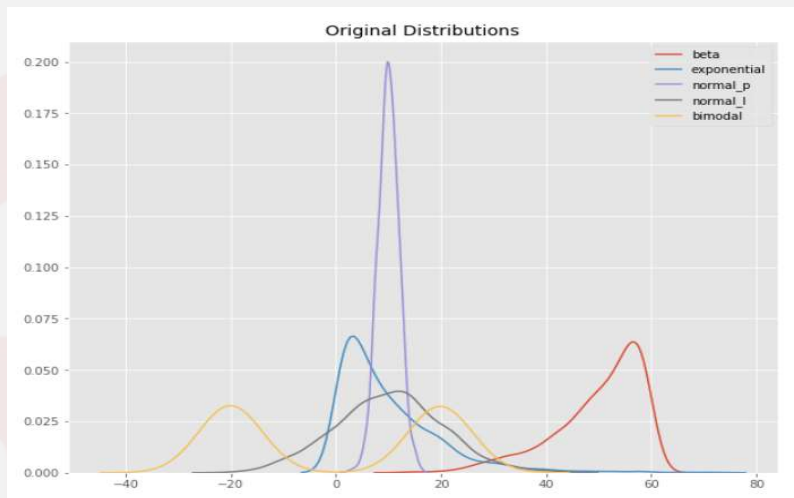
- Subtrai o valor mínimo e divide pela faixa
- O valor resultante recai no intervalo 0-1

RobustScaler



- Subtrai a média e divide pelo interquartil (75-25%)

StandardScaler



- Subtrai a média e ajusta a distribuição de modo que o desvio seja igual a 1



Outros métodos de classificação

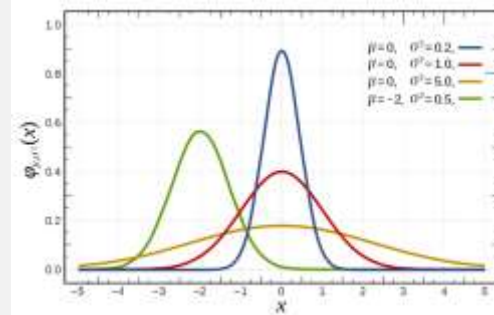
■ Classificação Bayseana

- Um dos métodos mais antigos em AM (década de 60)
- Motivação original era a classificação de texto
- Uma das principais aplicações são os sistemas de Anti-SPAM, classificadores de texto com base em frequência (ou saco de palavras). Aplicações mais recentes em medicina
- Utilizam o teorema de Bayes tornando o problema de classificação em um problema de decisão.

■ Classificação Bayseana (Cont.)

- Teorema de Bayes (ou Naive Bayes)
- Distribuição Gaussiana
- Algoritmo Gaussian Naive Bayes
 - Probabilidade das Classes (Labels)
 - Probabilidades Condicionais com base nas features

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$



Gaussian Naïve Bayes

- Classificador probabilístico baseado no Teorema de Bayes
- Por premissa as características (features) precisam ser independentes
- Na ocorrência de variáveis (características contínuas) é comum aplicar o Gaussian Naïve Bayes. Aplicações: classificação de pessoas segundo características e classificação de documentos (ex. Anti-Spam).

a vector $\mathbf{x} = (x_1, \dots, x_n)$ representing some n features (independent variables),

$$p(C_k | \mathbf{x}) = \frac{p(C_k) p(\mathbf{x} | C_k)}{p(\mathbf{x})}$$

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

$$\begin{aligned} p(C_k, x_1, \dots, x_n) &= p(x_1, \dots, x_n, C_k) \\ &= p(x_1 | x_2, \dots, x_n, C_k) p(x_2, \dots, x_n, C_k) \\ &= p(x_1 | x_2, \dots, x_n, C_k) p(x_2 | x_3, \dots, x_n, C_k) p(x_3, \dots, x_n, C_k) \\ &= \dots \\ &= p(x_1 | x_2, \dots, x_n, C_k) p(x_2 | x_3, \dots, x_n, C_k) \dots p(x_{n-1} | x_n, C_k) p(x_n | C_k) p(C_k) \end{aligned}$$

■ Propriedades

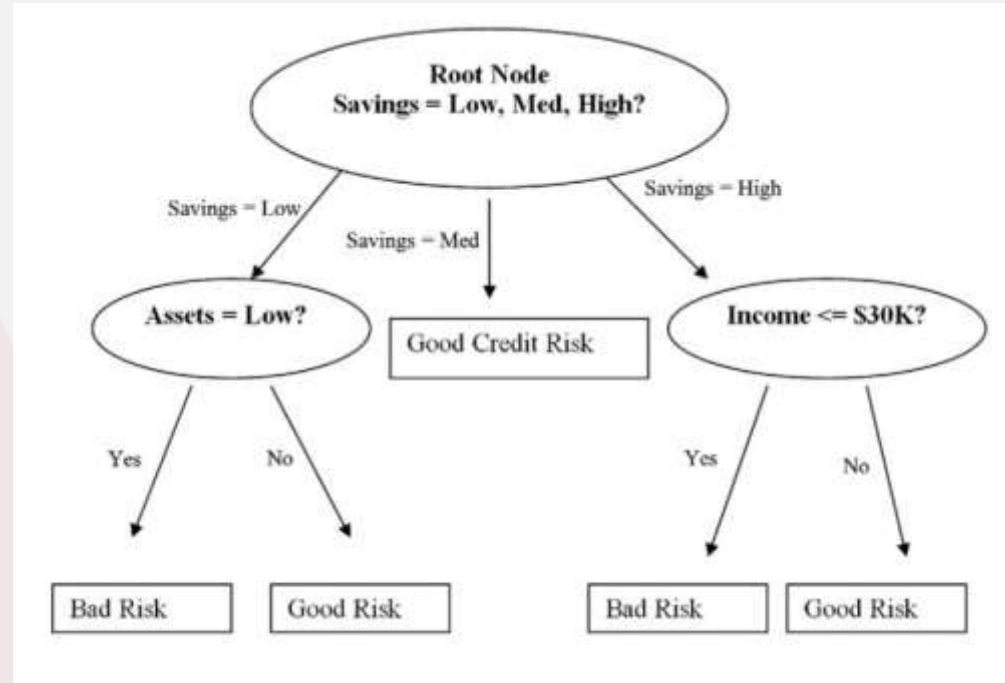
- Pode realizar classificação binária ou multi-classe
- Labels podem ser binários, categóricos ou nominais
- Se as features forem numéricas o algoritmo vai trabalhar melhor se a distribuição for Normal ou próxima da Normal. Importante remover outliers e normalizar os dados



Árvores de Decisão

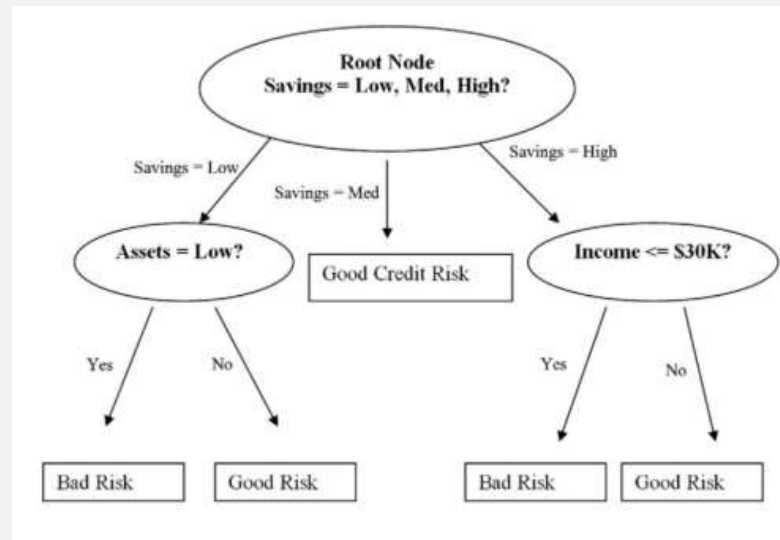
■ Árvores de Decisão

- **Root Node** (ou Raiz)
- Splitting (ou divisão)
- Decision Node
- **Leaf** / Terminal Node (ou Folha)
- **Prunning** (ou poda)
- Branch / SubTree (ou Ramo)
- **Parent** and **Child** Nodes



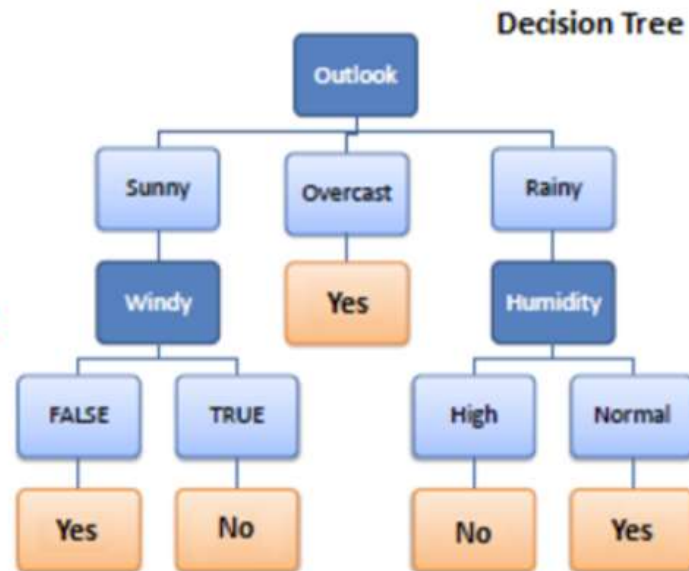
Aprendizagem baseada em Decision Trees

- Aprendizado Supervisionado
- Pode ser utilizado tanto para classificação como regressão
- O uso mais comum (veremos no curso) é como classificador
- O treinamento consiste em promover particionamento sucessivo dos dados
- O processo de treinamento constrói uma árvore (de forma indutiva) para atuar como classificador



Um exemplo de treinamento

Predictors				Target
Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



■ Algoritmos

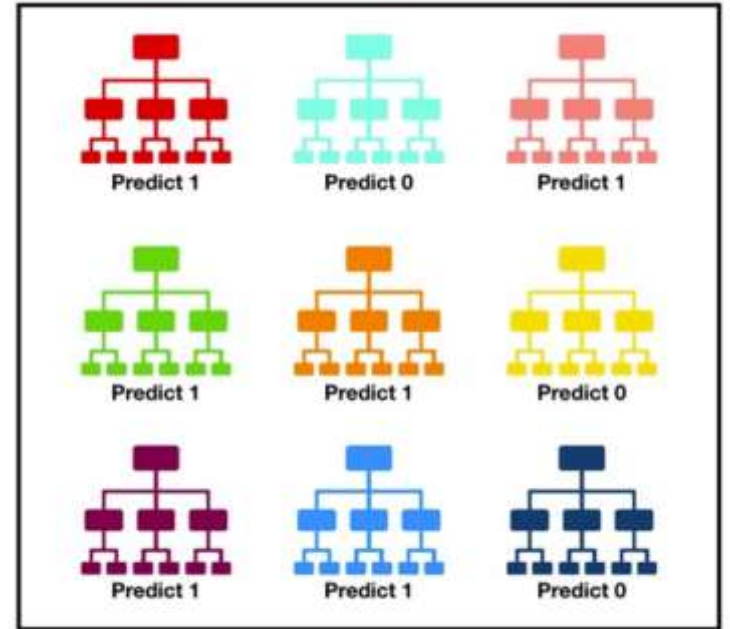
- Buscam maximizar a eficiência da árvore medindo a “pureza” de suas partições:
- Ganho de informação (algoritmo ID3 – entropia)
- Taxa/Razão de ganho (algoritmo C4.5)
- Gini index



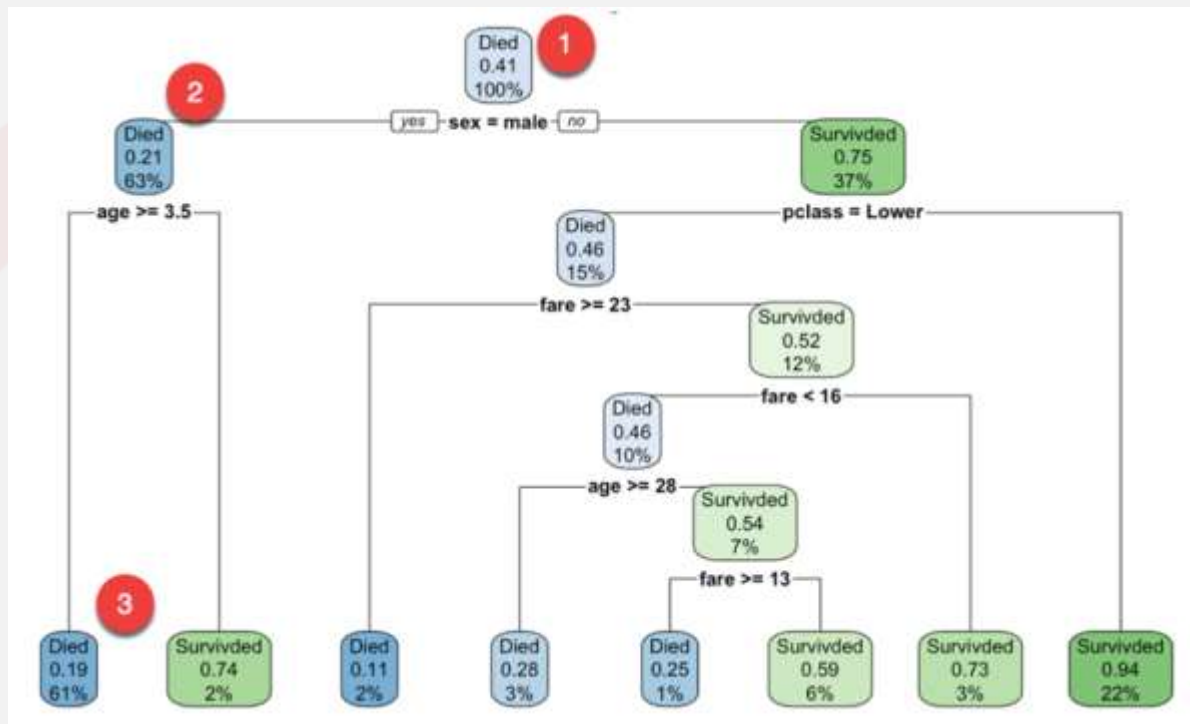
Random Forests

Random Forests

- Método baseado em Ensembles
- Parte do princípio que árvores independentes operando em um colegiado produzem melhores resultados.



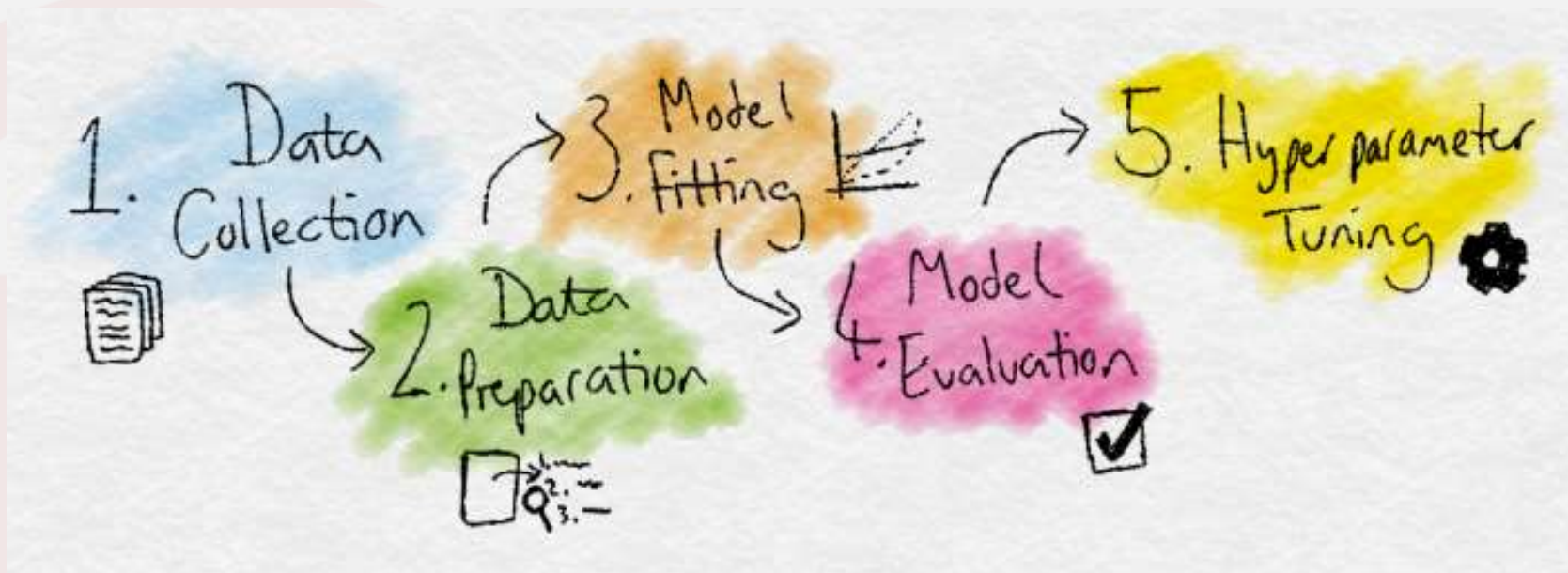
Vamos experimentar (ver Moodle)





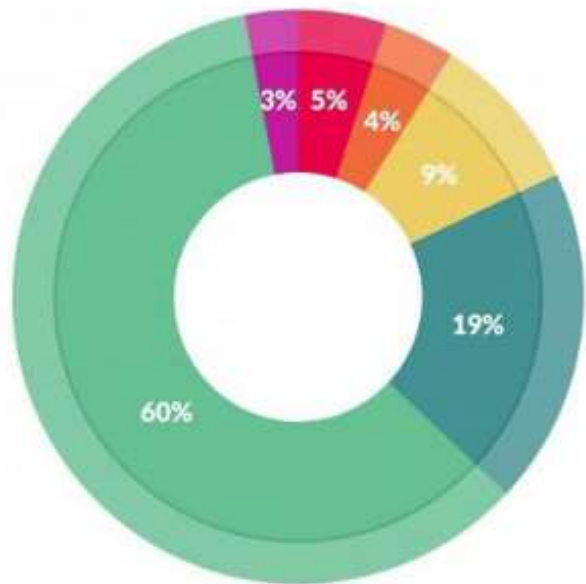
Feature Engineering

O processo de treinamento



Fonte: <https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0>

A preparação dos dados



What data scientists spend the most time doing

- Building training sets: 3%
- Cleaning and organizing data: 60%
- Collecting data sets: 19%
- Mining data for patterns: 9%
- Refining algorithms: 4%
- Other: 5%

Source: <https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/>

■ Tratamento de nulos e exclusão

1. Datasets podem conter features com valores vazios. Isso ocorre por erro humano (data entry), de sistemas e/ou coleta.
2. Remover registros onde uma dada feature está vazio
3. Remover (desconsiderar) uma feature quanto a frequência de registros vazios for alta demais (ex. >60-70%)
4. Dependendo da relevância da feature é necessário trabalhar a coleta do dado

■ Preenchimento

1. Dependendo da frequência de valores nulos (quando baixa - $<5-20\%$) é possível preencher vazios com algum valor numérico ou categórico.
2. Para valores numéricos é comum usar média ou mediana.
3. Outra possibilidade é por usar agrupamentos (clustering) para categorizar e portanto derivar o melhor valor para a feature.

■ Outliers

1. Identifiando outliers com o desvio padrão (tipicamente a uma distância da média $> 2-4$ desvios)
2. A primeira alternativa (e mais comum) é simplesmente eliminar o outlier
3. Outra opção é substituir o valor por um valor máximo (cap)

Binning

#Numerical Binning Example

Value	Bin
0-30 ->	Low
31-70 ->	Mid
71-100 ->	High

#Categorical Binning Example

Value	Bin
Spain ->	Europe
Italy ->	Europe
Chile ->	South America
Brazil ->	South America

Fonte: <https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114>

One-Hot Encoding

User	City
1	Roma
2	Madrid
1	Madrid
3	Istanbul
2	Istanbul
1	Istanbul
1	Roma



User	Istanbul	Madrid
1	0	0
2	0	1
1	0	1
3	1	0
2	1	0
1	1	0
1	0	0

One hot encoding example on City column

Fonte: <https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114>

Feature Split

```
data.name
0  Luther N. Gonzalez
1   Charles M. Young
2     Terry Lawson
3   Kristen White
4   Thomas Logsdon

#Extracting first names
data.name.str.split(" ").map(lambda x: x[0])
0      Luther
1    Charles
2     Terry
3    Kristen
4     Thomas

#Extracting last names
data.name.str.split(" ").map(lambda x: x[-1])
0    Gonzalez
1     Young
2    Lawson
3     White
4    Logsdon
```

Fonte: <https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114>

■ Normalização

```
data = pd.DataFrame({'value':[2,45, -23, 85, 28, 2, 35, -12]})  
  
data['normalized'] = (data['value'] - data['value'].min()) /  
(data['value'].max() - data['value'].min())
```

	value	normalized
0	2	0.23
1	45	0.63
2	-23	0.00
3	85	1.00
4	28	0.47
5	2	0.23
6	35	0.54
7	-12	0.10

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Fonte: <https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114>

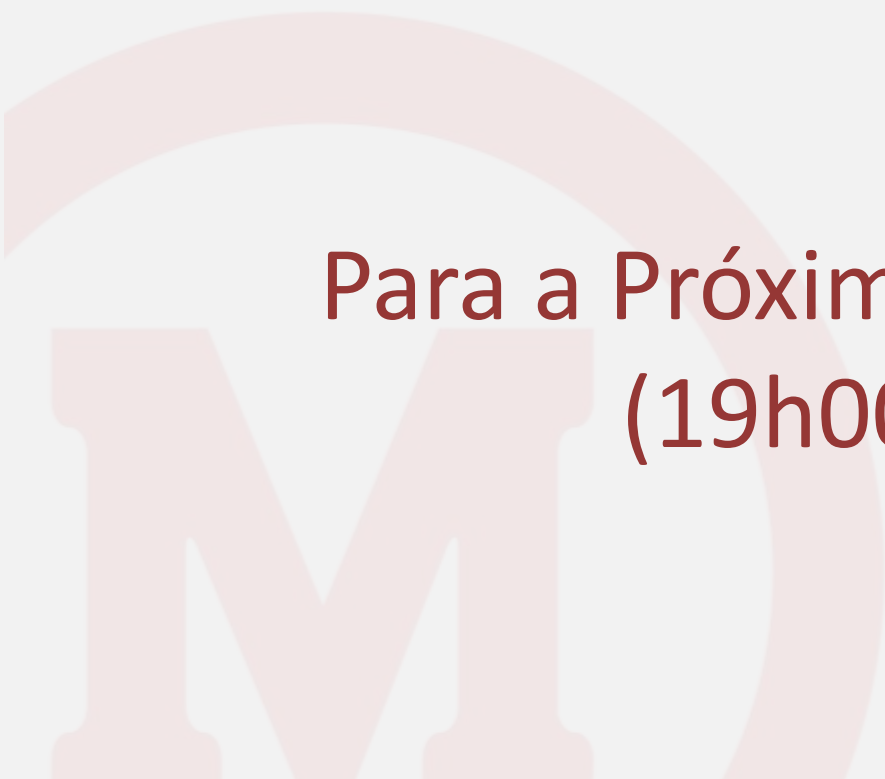
■ Padronização

```
data = pd.DataFrame({'value':[2,45, -23, 85, 28, 2, 35, -12]})  
  
data['standardized'] = (data['value'] - data['value'].mean()) /  
data['value'].std()
```

	value	standardized
0	2	-0.52
1	45	0.70
2	-23	-1.23
3	85	1.84
4	28	0.22
5	2	-0.52
6	35	0.42
7	-12	-0.92

$$z = \frac{x - \mu}{\sigma}$$

Fonte: <https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114>



Para a Próxima Aula – 23/jun
(19h00-22h30)

Exercício de Aprofundamento (Titanic)



- Vamos resolver o problema clássico do Titanic
- **Link:** <https://www.kaggle.com/c/titanic>
- Classificação Binária
- Você deve treinar os modelos usando o conjunto de treinamento e avaliar os resultados utilizando o conjunto de testes
- Você deve utilizar os conceitos de preparação de dados apresentados na aula de hoje nesse exercício. Nesse problema uma boa preparação será fundamental.
- Explore os algoritmos de classificação vistos até o momento: Regressão Logística, Gaussian Naive Bayes, Árvores de Decisão e Random Forests. Sua entrega deve apresentar um comparativo das acurácias obtidas com cada algoritmo no conjunto de testes.
- **Dicas:** Não precisa usar Cross-Validation. Utilize os datasets de treinamento e validação apenas. Lembre-se que todo tratamento de preparação realizado no conjunto de treinamento deve ser replicado no dataset de validação.

Até a próxima aula

MUITO OBRIGADO!