



# Aplicações de Aprendizado de Máquina & PLN

Juvenal J. Duarte

# *Aula 2: Classificação*

## *Ementa*

Tópicos de hoje:

1. Problemas comuns em classificação e suas abordagens.
2. Algoritmos de classificação e suas características.
3. Medidas de avaliação
4. Classificação de dados desbalanceados
5. Viés e variância
6. Otimização de hiper-parâmetros



# Introdução:

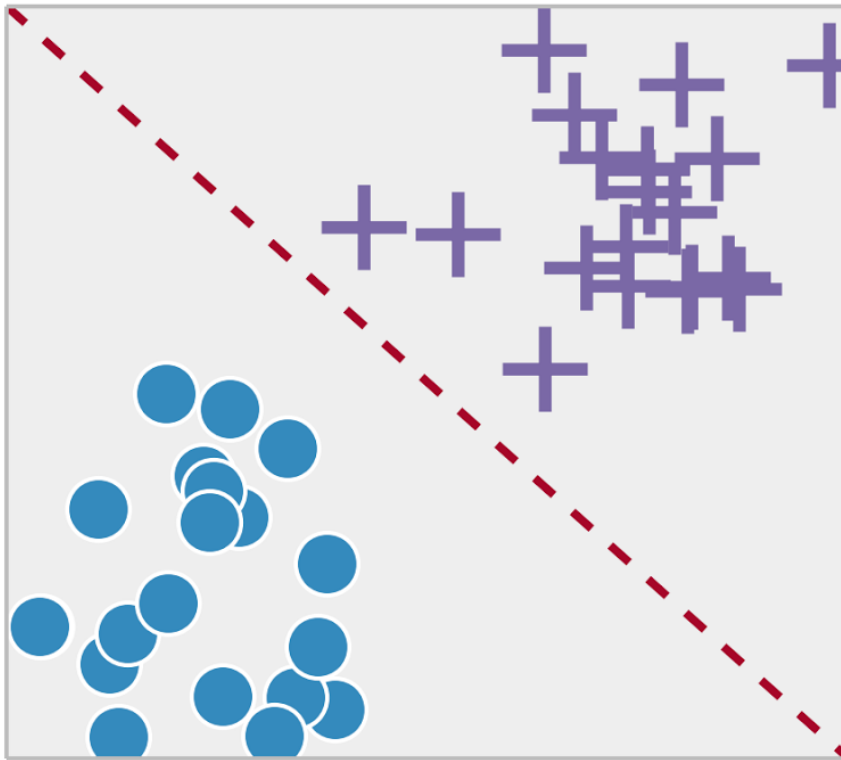
Tipos de problemas e abordagens

Juvenal J. Duarte

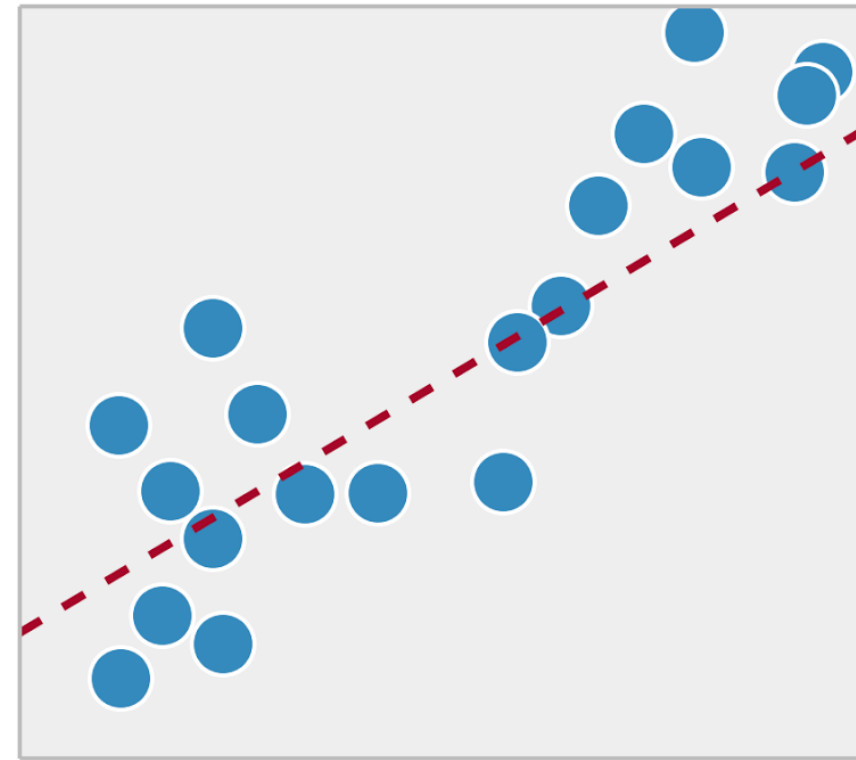
# Regressão / Classificação

*Qual a diferença?*

Classification

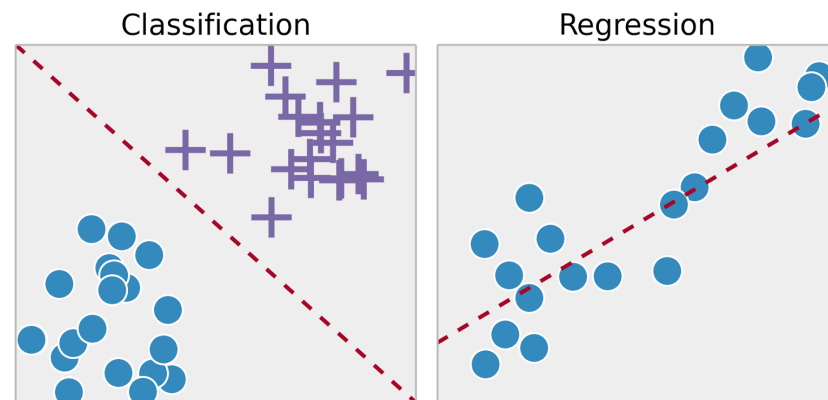


Regression



# Regressão / Classificação

Qual a diferença?



	Classificação	Regressão
<b>Atributo alvo</b>	categórico	numérico contínuo
<b>Tipo de modelo</b>	função de decisão	regressor
<b>F. Custo (mais comum)</b>	Cross Entropy	Mean Squared Error
<b>Avaliação (mais comum)</b>	F-Medida, Precisão, Revocação, ROC/AUC	MSE, MAE, RMSE, Residual Plots

# Regressão / Classificação

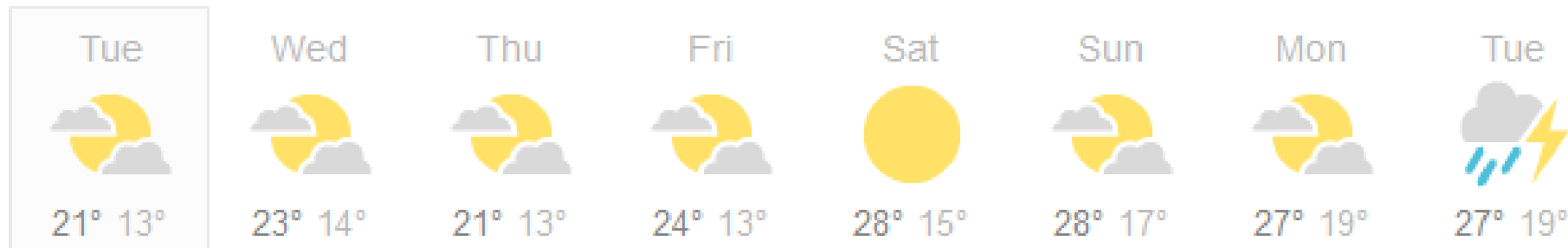
Exemplo

**Ex. 1: Fará um dia ensolarado, nublado ou chuvoso?**

Classes = {"ensolarado", "nublado", "chuvoso"}

**Ex. 2: Qual será a temperatura média?**

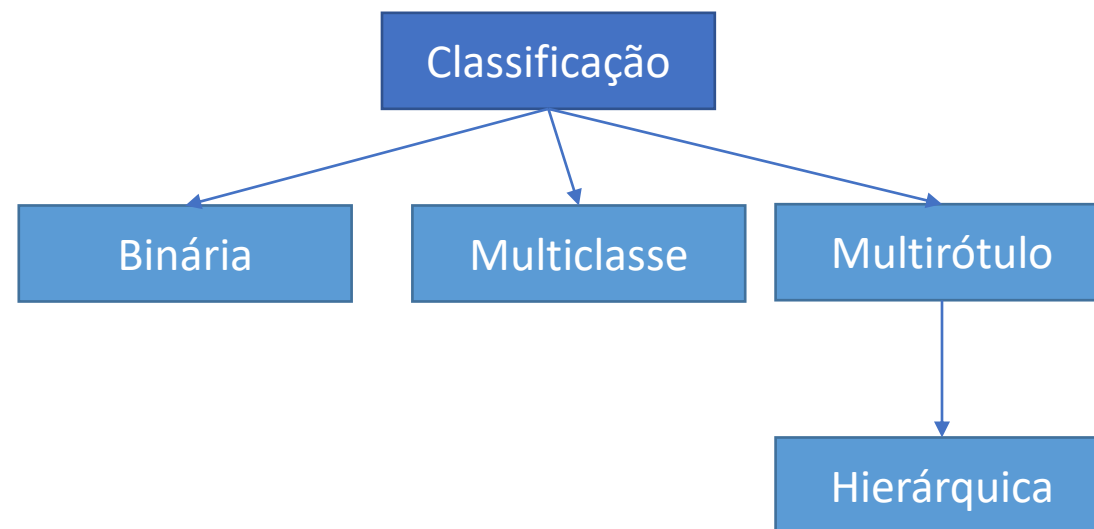
Temperatura = [-50 C, +50 C]



# Classificação

## Tipos de Problemas

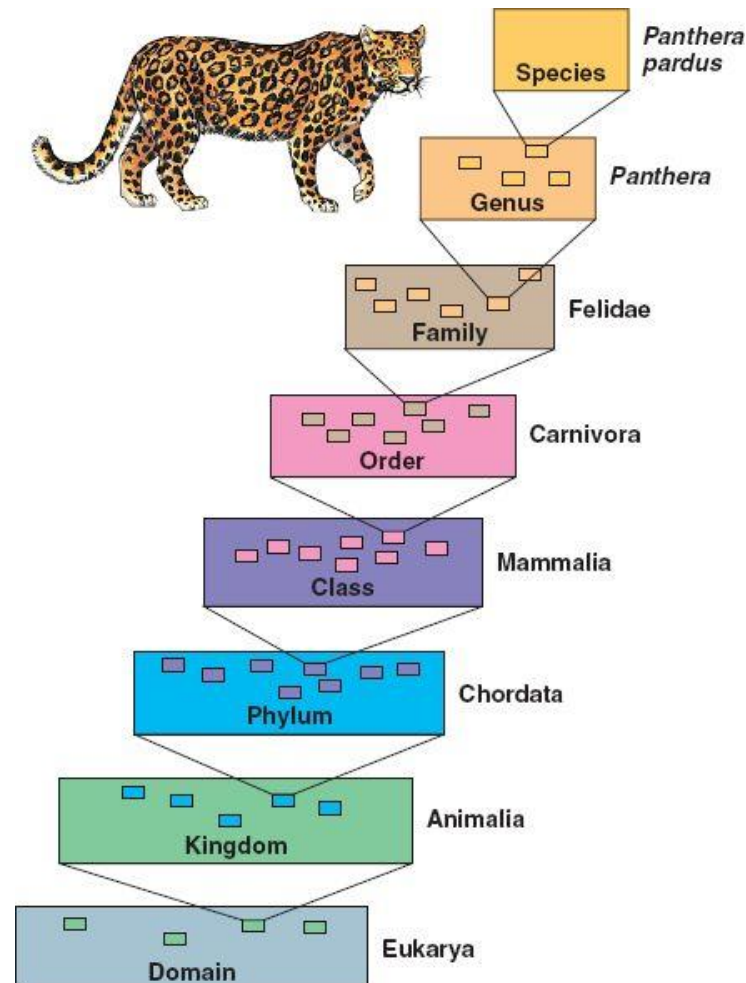
	Descrição
<b>Binária</b>	Muito comum para problemas de diagnóstico: Classe positiva Vs negativa. Ex. Diabetes, Classificação de Crédito.
<b>Multiclasse</b>	Usado quando o modelo deve decidir entre mais de duas classes. Ex. Iris Dataset, MNIST.
<b>Multirótulo</b>	Neste tipo de problema, registros podem receber mais de um rótulo. Ex. Muito usado na classificação de textos.
<b>Hierárquica</b>	Para alguns casos, a quantidade de classes possíveis é muito grande para ser tratada por um único modelo. Ex. Qual o autor de uma obra.



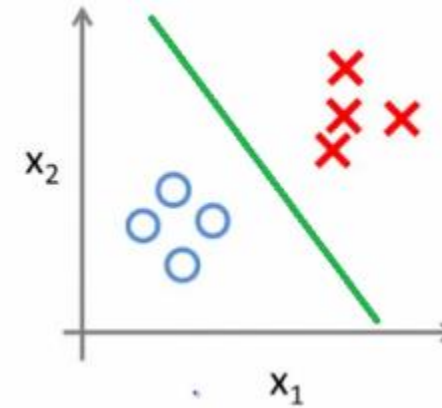
# Classificação

*Tipos de Problemas*

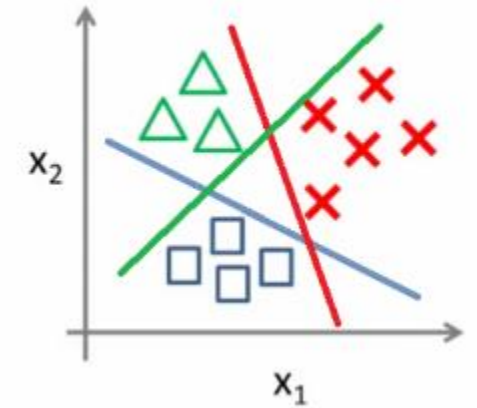
**Hierarchical classification:**









**Binary classification:**






**Multi-class classification:**



**Multi-Class**

$C = 3$	Samples
  	  
Labels (t)	$[0 \ 0 \ 1]$ $[1 \ 0 \ 0]$ $[0 \ 1 \ 0]$

**Multi-Label**

Samples
  
Labels (t)
$[1 \ 0 \ 1]$ $[0 \ 1 \ 0]$ $[1 \ 1 \ 1]$

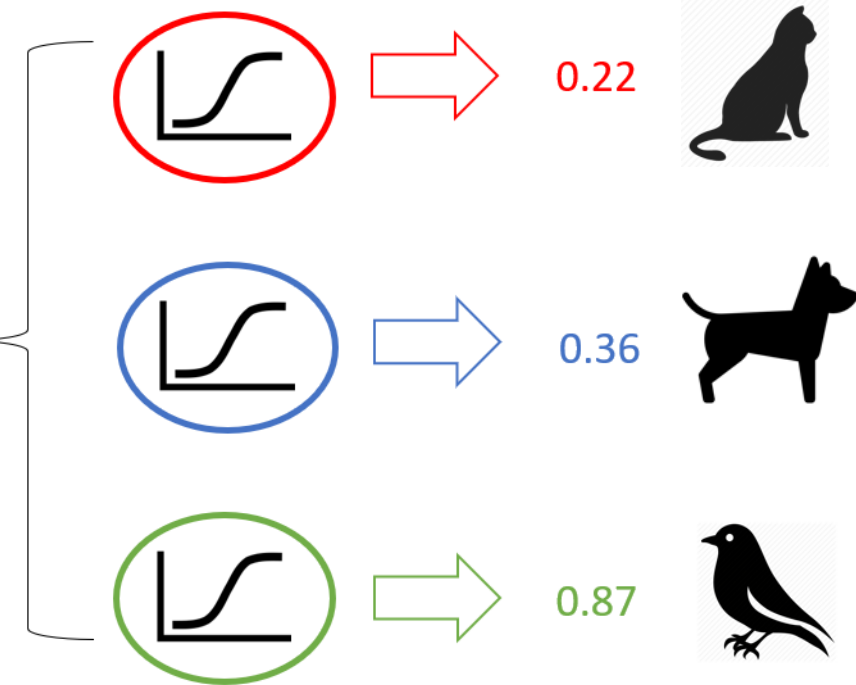


# Classificação Multiclasse

*One Vs All (One Vs Rest)*



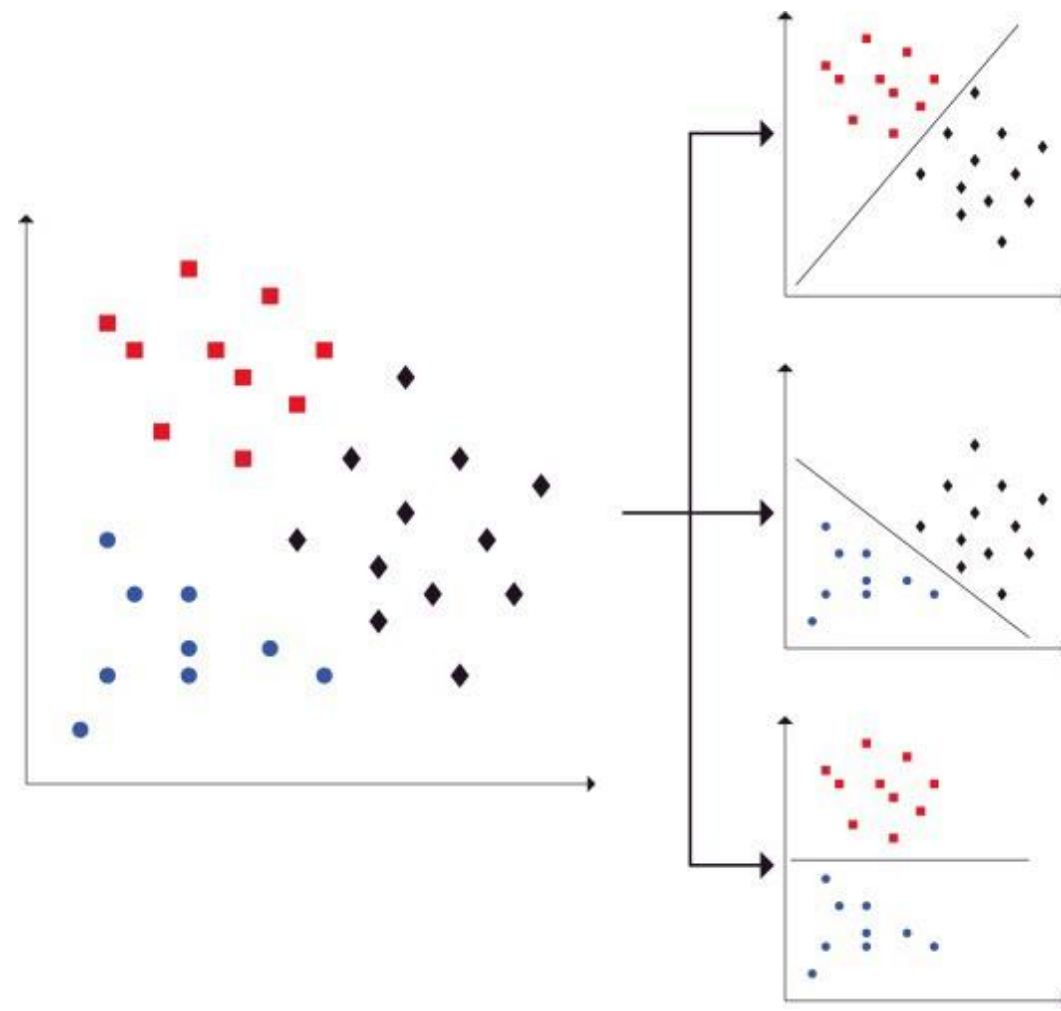
- Constroi-se um modelo para cada classe.
- O modelo aprende a diferenciar a classe de todas as demais.



# Classificação Multiclasse

*One Vs One (All Vs All)*

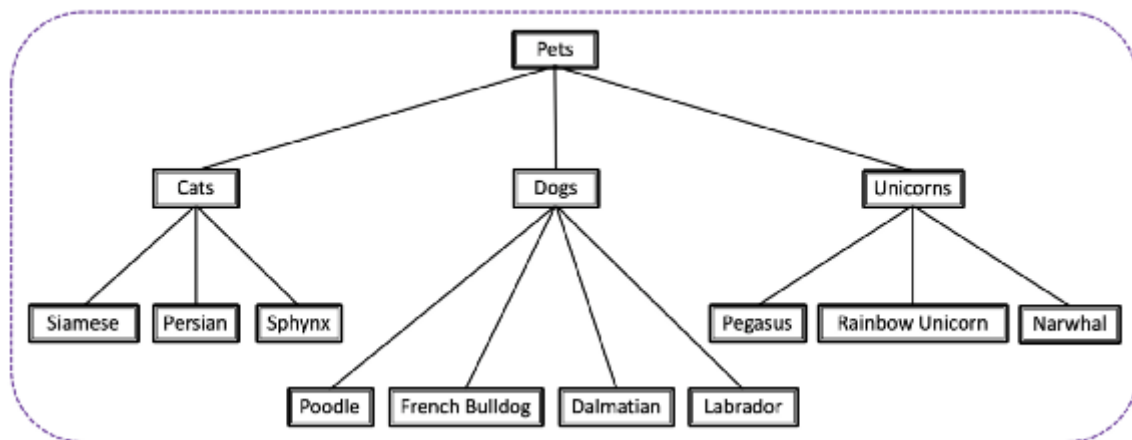
- Constroi-se um modelo para cada combinação duas a duas das classes.
- Resulta em mais modelos, porém mais simples.



# Classificação Multirótulo & Hierárquica

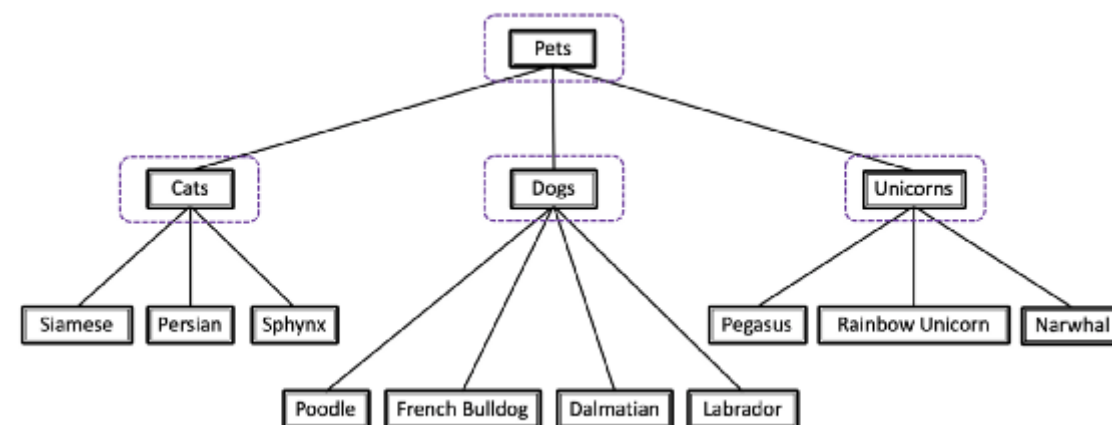
*Global Vs Local*

**Global**



- Um único classificador para distinguir entre todas as classes possíveis.
- Pode ser abordado como um problema multirótulo.

**Local (por nó pai)**



- Um classificador por nó pai.
- Classificadores encadeados baseado na classificação do nível anterior.

<https://towardsdatascience.com/https-medium-com-noa-weiss-the-hitchhikers-guide-to-hierarchical-classification-f8428ea1e076>



# Algoritmos:

Características, vantagens e  
limitações

Juvenal J. Duarte

# Classificação

*Técnicas mais comuns*

Baseado em Instância	Baseado em Regras	Modelos Lineares	Modelos Não Lineares	Comitês (Ensembles)
K-Nearest Neighbors	Árvores de Decisão	Regressão Logística	Redes Neurais	Bagging
	Floresta Aleatória	Naïve Bayes	Redes Bayesianas	Boosting
	Gradient Boosting	SVM-Linear	SVM-Polinomial	
	XGBoost		SVM-RBF	

# Classificação

Técnicas mais comuns

Baseado em Instância	Baseado em Regras	Modelos Lineares	Modelos Não Lineares	Comitês (Ensembles)
K-Nearest Neighbors	Árvores de Decisão	Regressão Logística	Redes Neurais	Bagging
	Floresta Aleatória	Naïve Bayes	Redes Bayesianas	Boosting
	Gradient Boosting	SVM-Linear	SVM-Polinomial	
	XGBoost		SVM-RBF	
Técnica simples com resultados tão bons quanto os melhores métodos dependendo do problema. Sofre com alta dimensionalidade (wide) e, principalmente, com muitos registros (long).	Modelos interessantes principalmente pela capacidade de interpretação. AD possui problemas crônicos com alta dimensionalidade e super ajustamento, os demais (especialmente XGBoost) possuem artifícios para estes problemas	De fácil configuração e pouco sensíveis a super ajustamento. Sofrem com problemas não lineares e atributos não independentes (colinearidade)	Conseguem representar funções altamente complexas, mas são mais susceptíveis a superajustamento. Exigem ajuste de hiperparâmetros e regularização muito mais intensos.	Combinação de modelos “fracos”. Usados para reduzir o viés sem comprometer a capacidade de generalização (variância).

# Classificação

Técnicas mais comuns

Baseado em Instância	Baseado em Regras	Modelos Lineares	Modelos Não Lineares	Comitês (Ensembles)
K-Nearest Neighbors	Árvores de Decisão	Regressão Logística	Redes Neurais	Bagging
	Floresta Aleatória	Naïve Bayes	Redes Bayesianas	Boosting
	Gradient Boosting	SVM-Linear	SVM-Polinomial	
	XGBoost		SVM-RBF	

Permite avaliar o caminho (conjunto de regras) que levaram à decisão.

Conseguem tratar problemas multiclasse/multilabel em um único modelo

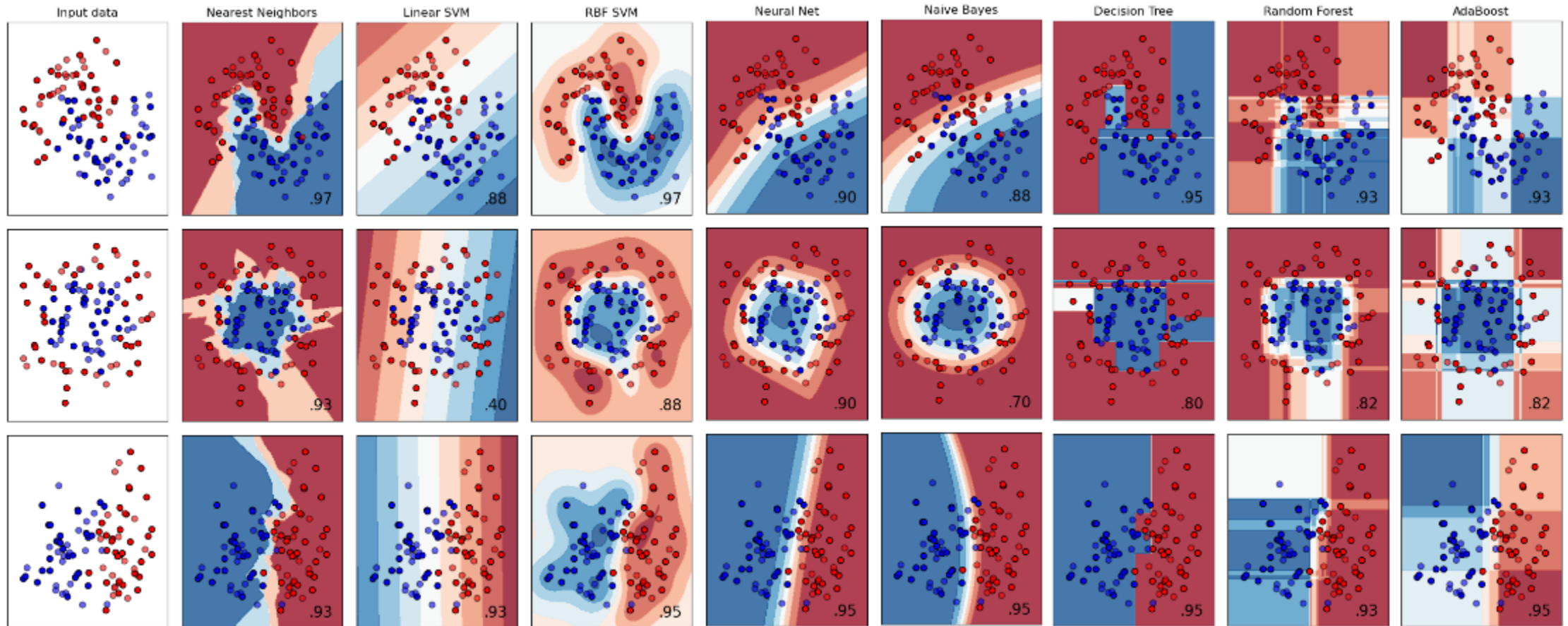
Permite a comparação com exemplos observados mais parecidos.

Saída em forma de probabilidade, permitindo a manipulação do limiar de decisão.

Decomposição mais clara do problema, pode combinar as vantagens de diferentes métodos amenizando suas desvantagens.

# Classificação

Comparativo: funções de decisão





# Classificação

## Referência

### **Regressão Logística:**

[https://scikit-learn.org/stable/modules/generated/sklearn.linear\\_model.LogisticRegression.html](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)

### **SVM:**

<https://scikit-learn.org/0.15/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC>

### **Árvore de Decisão:**

<https://scikit-learn.org/0.15/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier>

### **Floresta Aleatória:**

<https://scikit-learn.org/0.15/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier>

### **Naïve Bayes Gaussiano:**

[https://scikit-learn.org/0.15/modules/generated/sklearn.naive\\_bayes.GaussianNB.html#sklearn.naive\\_bayes.GaussianNB](https://scikit-learn.org/0.15/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB)

### **KNN:**

<https://scikit-learn.org/0.15/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier>

### **Rede Neural Artificial (MultiLayer Perceptron):**

[https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPClassifier.html](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html)



# Avaliação:

## Medidas de avaliação

Juvenal J. Duarte

# Medidas de avaliação

## Matriz de Confusão

- Compara os rótulos reais com os previstos.
- Cada quadrante indica a quantidade de registros. A soma de todos os quadrantes deve ser igual ao número total de registros  $m$ .
- Para problemas binários, as situações possíveis são:
  - *True Positive (TP): Registro cujo rótulo real é positivo e foi previsto como positivo.*
  - *True Negative (TN): Registro negativo previsto como negativo.*
  - *False Positive (FP): Registro negativo previsto como positivo.*
  - *False Negative (FN): Registro positivo previsto como negativo.*
- Para problemas multiclasse a matriz ajuda a identificar quais as classes confundidas com mais frequência.
  - As métricas TP, TN, FP e FN podem ser calculadas como One Vs Rest.

### Problema Binário

	Predicted Positives	Predicted Negatives
Positives	True Positives	False Negatives
Negatives	False Positives	True Negatives

### Problema Multiclasse

	Car	Boat	Plane	Train
Car	<div>✓ 88</div>	<div>✗ 0</div>	<div>✗ 0</div>	<div>✗ 1</div>
Boat	<div>✗ 0</div>	<div>✓ 65</div>	<div>✗ 0</div>	<div>✗ 3</div>
Plane	<div>✗ 0</div>	<div>✗ 0</div>	<div>✓ 72</div>	<div>✗ 4</div>
Train	<div>✗ 4</div>	<div>✗ 0</div>	<div>✗ 1</div>	<div>✓ 60</div>

# Medidas de avaliação

## Matriz de Confusão

- Compara os rótulos reais com os previstos.
- Cada quadrante indica a quantidade de registros. A soma de todos os quadrantes deve ser igual ao número total de registros  $m$ .
- Para problemas binários, as situações possíveis são:
  - *True Positive (TP): Registro cujo rótulo real é positivo e foi previsto como positivo.*
  - *True Negative (TN): Registro negativo previsto como negativo.*
  - *False Positive (FP): Registro negativo previsto como positivo.*
  - *False Negative (FN): Registro positivo previsto como negativo.*
- Para problemas multiclasse a matriz ajuda a identificar quais as classes confundidas com mais frequência.
  - As métricas TP, TN, FP e FN podem ser calculadas como One Vs Rest.

Acertos!

Problema Binário

	Predicted Positives	Predicted Negatives
Positives	True Positives	False Negatives
Negatives	False Positives	True Negatives

Problema Multiclasse

	Boat	Plane	Train
Boat	88 0	0	1
Plane	0	65 0	3
Train	4	0	1 60

Erros!

# Medidas de avaliação

## Matriz de Confusão - Python

### Código

```
from sklearn.metrics import confusion_matrix

y_actu = [2, 0, 2, 2, 0, 1, 1, 2, 2, 0, 1, 2]
y_pred = [0, 0, 2, 1, 0, 2, 1, 0, 2, 0, 2, 2]
confusion_matrix(y_actu, y_pred)
```

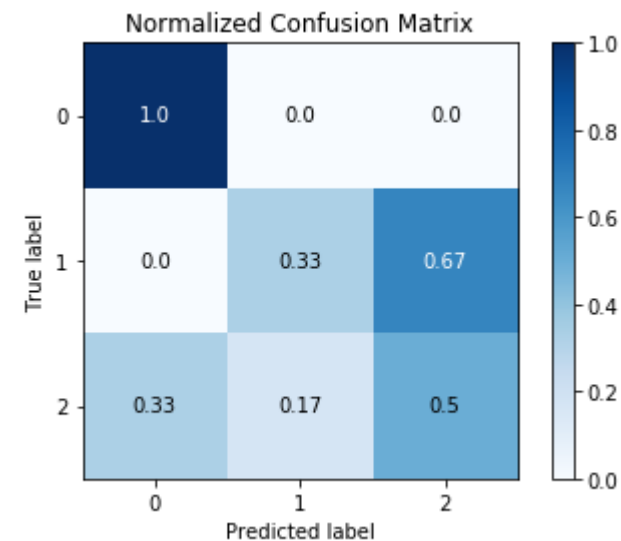
```
import scikitplot as skplt

skplt.metrics.plot_confusion_matrix(y_actu,
                                    y_pred,
                                    normalize=True)

plt.show()
```

### Saída

```
array([[3, 0, 0],
       [0, 1, 2],
       [2, 1, 3]], dtype=int64)
```



# Medidas de avaliação

## Medidas de acerto

- Revocação / Taxa de positivos verdadeiros ( $recall = \frac{TP}{TP+FN}$ ):
  - Quantos casos positivos foram previstos dentre todos os registros positivos nos dados?
- Precisão ( $precision = \frac{TP}{TP+FP}$ ):
  - Dos casos positivos previstos, quantos realmente eram positivos?
- F-Medida ( $F - Score = 2 * \frac{precision*recall}{precision+recall}$ ):
  - Média harmônica entre a precisão e revocação.
- Taxa de positivos falsos ( $recall = \frac{FP}{TN+FP}$ ):
  - Quantos casos negativos foram previstos dentre todos os registros negativos nos dados?
- Acurácia ( $accuracy = \frac{TP+TN}{m}$ ):
  - Quantos casos foram previstos corretamente dentre todos os registros?
- Taxa de erro ( $error\ rate = \frac{FP+FN}{m}$ ):
  - Quantos casos foram previstos erroneamente dentre todos os registros?

# Medidas de avaliação

## Medidas de acerto

Sensíveis a classe  
escolhida como  
positiva!

- Revocação / Taxa de positivos verdadeiros ( $recall = \frac{TP}{TP+FN}$ ):
  - Quantos casos positivos foram previstos dentre todos os registros positivos nos dados?
- Precisão ( $precision = \frac{TP}{TP+FP}$ ):
  - Dos casos positivos previstos, quantos realmente eram positivos?
- F-Medida ( $F - Score = 2 * \frac{precision*recall}{precision+recall}$ ):
  - Média harmônica entre a precisão e revocação.

- Taxa de positivos falsos ( $recall = \frac{FP}{TN+FP}$ ):
  - Quantos casos negativos foram previstos dentre todos os registros negativos nos dados?
- Acurácia ( $accuracy = \frac{TP+TN}{m}$ ):
  - Quantos casos foram previstos corretamente dentre todos os registros?
- Taxa de erro ( $error\ rate = \frac{FP+FN}{m}$ ):
  - Quantos casos foram previstos erroneamente dentre todos os registros?

Não sensíveis a  
classe positiva!

# Medidas de avaliação

## Medidas de acerto

### Código

```
from sklearn.metrics import classification_report

y_actu = [2, 0, 2, 2, 0, 1, 1, 2, 2, 0, 1, 2]
y_pred = [0, 0, 2, 1, 0, 2, 1, 0, 2, 0, 2, 2]

target_names = ['class 0', 'class 1', 'class 2']
print(classification_report(y_actu, y_pred,
                           target_names=target_names))
```

### Saída

	precision	recall	f1-score	support
class 0	0.60	1.00	0.75	3
class 1	0.50	0.33	0.40	3
class 2	0.60	0.50	0.55	6
accuracy			0.58	12
macro avg	0.57	0.61	0.57	12
weighted avg	0.57	0.58	0.56	12



# Medidas de avaliação

## Limiar de decisão

- Alguns algoritmos fornecem a probabilidade de um registro pertencer a uma classe.
- Quando este é o caso, a classe é decidida por limiar de decisão, um *threshold*.
- A manipulação deste limiar permite tornar o modelo mais favorável a uma das classes, ajustando assim a taxa de acerto.

Exemplo	Classe Verdadeira	Probabilidade Prevista	Limiar de decisão = 0.6	Limiar de decisão = 0.7	Limiar de decisão = 0.8
1	0	0.98	1	1	1
2	1	0.67	1	0	0
3	1	0.58	0	0	0
4	0	0.78	1	1	0
5	1	0.85	1	1	1
6	0	0.86	1	1	1
7	0	0.79	1	1	0
8	0	0.89	1	1	1
9	1	0.82	1	1	1
10	0	0.86	1	1	1

<https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/>

# Medidas de avaliação

## Curva ROC (Receiver Operating Characteristic)

- Usado para fazer uma comparação gráfica entre a taxa de acertos contraposta à taxa de erro, conforme varia-se o threshold de separação das classes.

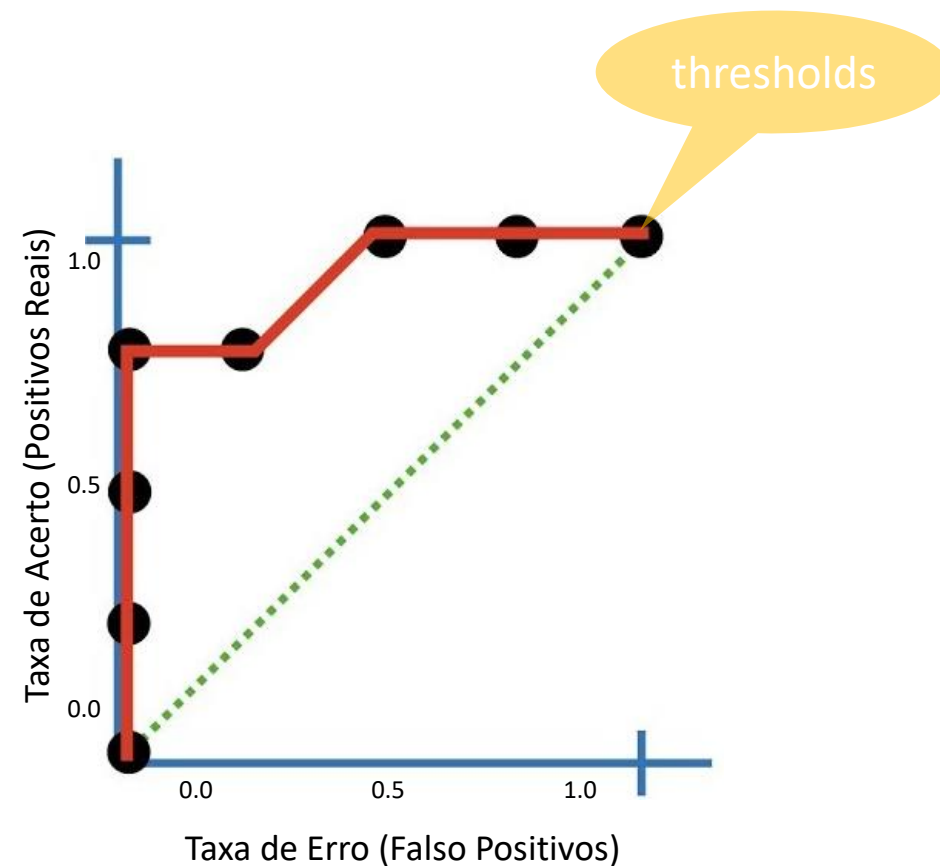
- **Construção:**

1. Escolha os thresholds de decisão, ex.: (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)
2. Baseado nos thresholds, compute as previsões do modelo.
3. Compare as previsões com os valores reais e calcule para cada threshold:

$$\text{taxa de acerto} = \frac{TP}{TP+FN}$$

$$\text{taxa de erro} = \frac{FP}{TN+FP}$$

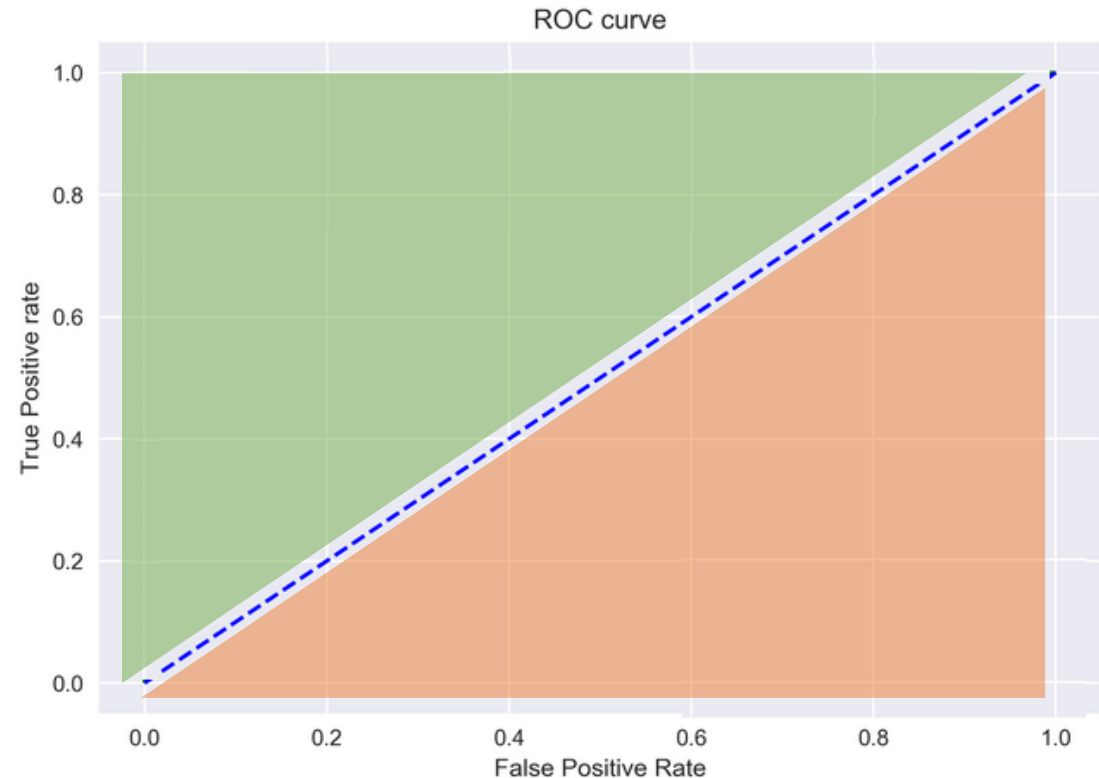
4. Plote os pares (taxa de erro, taxa de acerto) e conecte os pontos.



# Medidas de avaliação

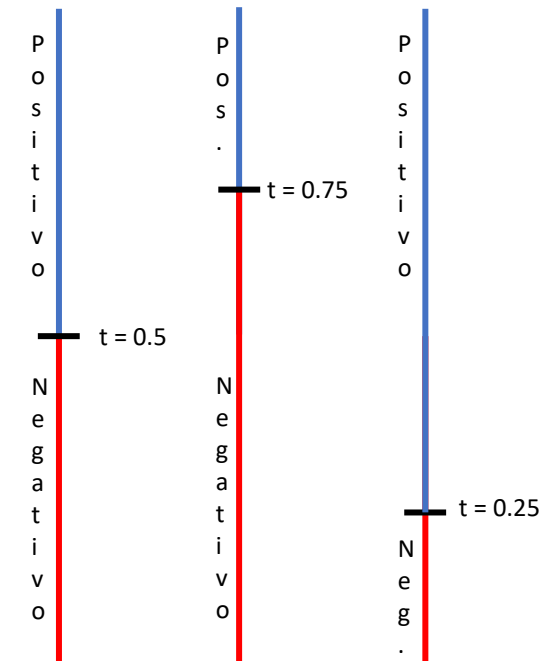
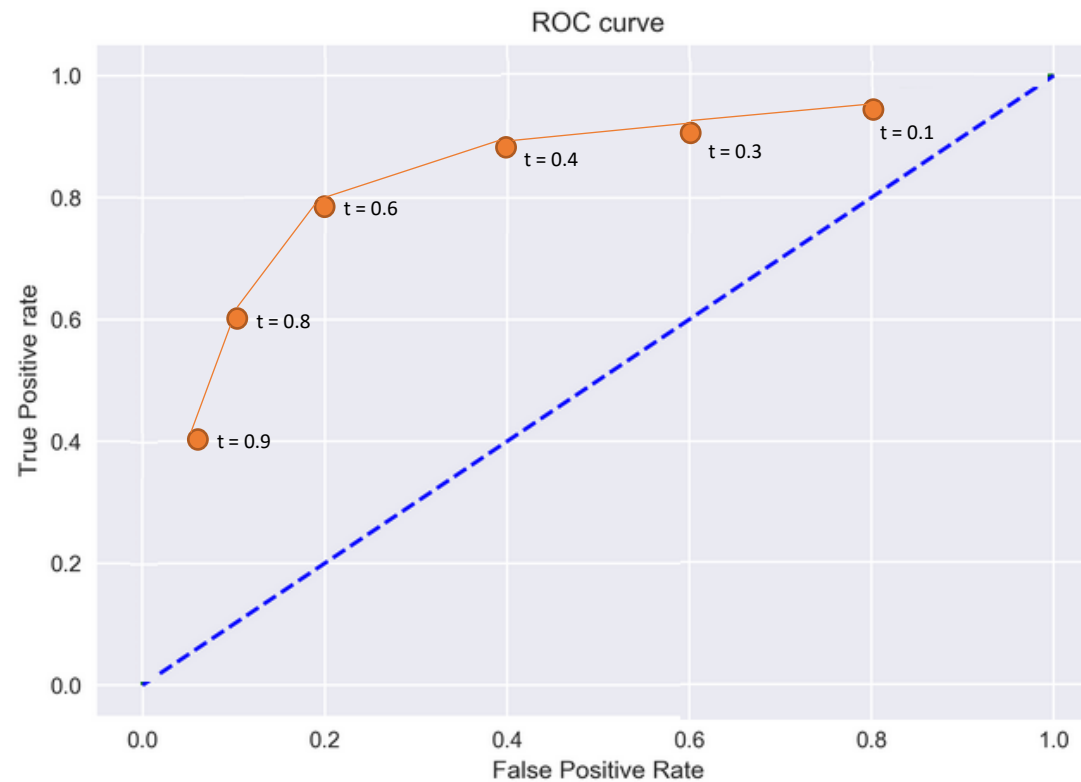
## Curva ROC: Interpretação

- A linha azul indica o comportamento de um classificador onde a quantidade de erros é sempre igual aos acertos, independente do threshold escolhido.
- A região verde é onde o classificador mantém a quantidade de acertos maior que a de erros (BOM!).
- A região laranja é onde os erros são mais frequentes que acertos (RUIM!).



# Medidas de avaliação

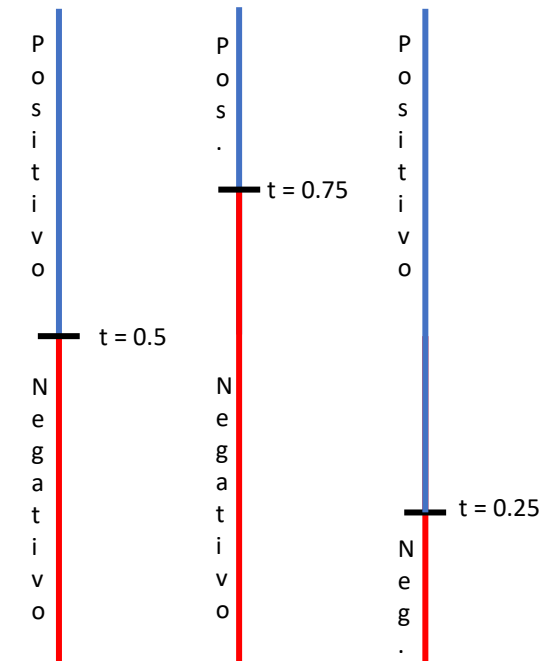
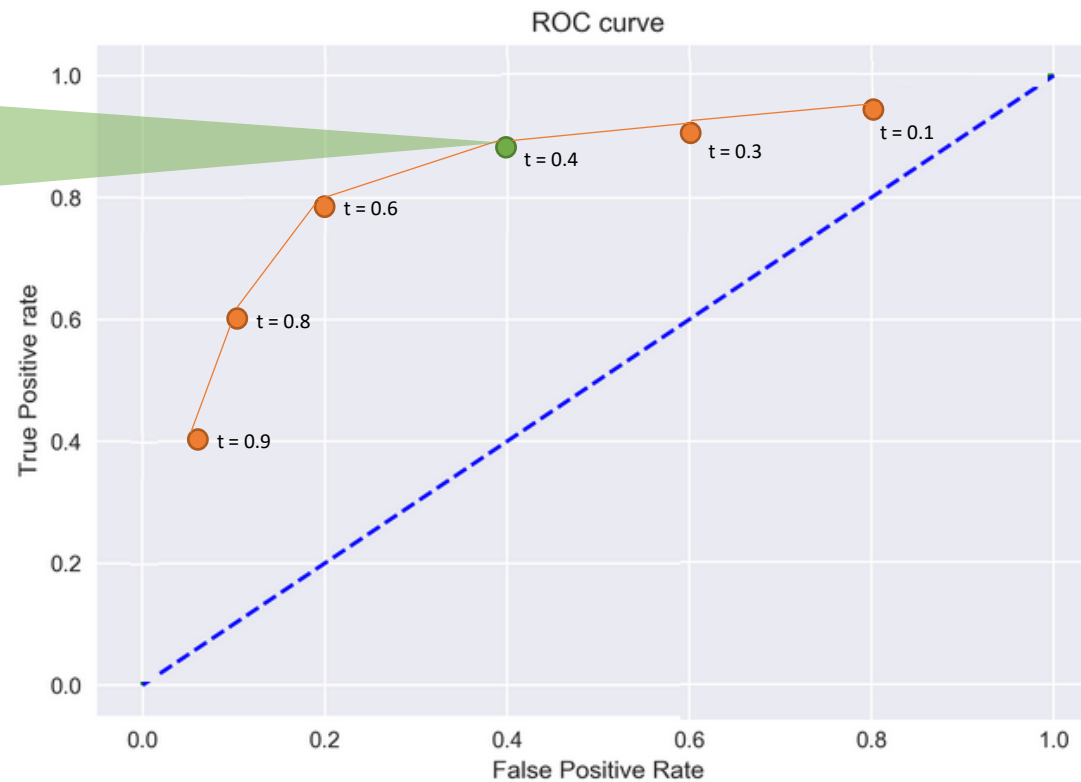
*Curva ROC: Qual o melhor classificador?*



# Medidas de avaliação

## Curva ROC: Qual o melhor classificador?

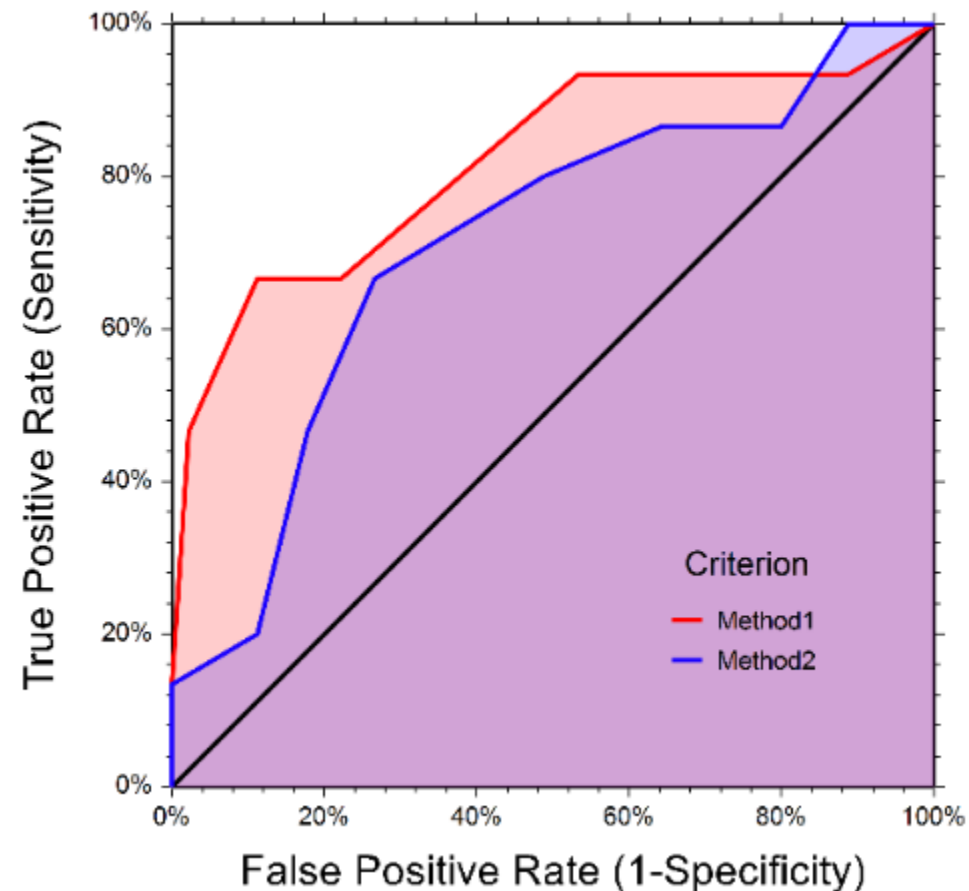
Na prática depende o quão crítico é seu problema, mas este é um bom candidato: *mantém uma boa taxa de acerto com erro notoriamente mais baixo que os pontos à direita.*



# Medidas de avaliação

## Curva ROC, superfície AUC (ROC-AUC)

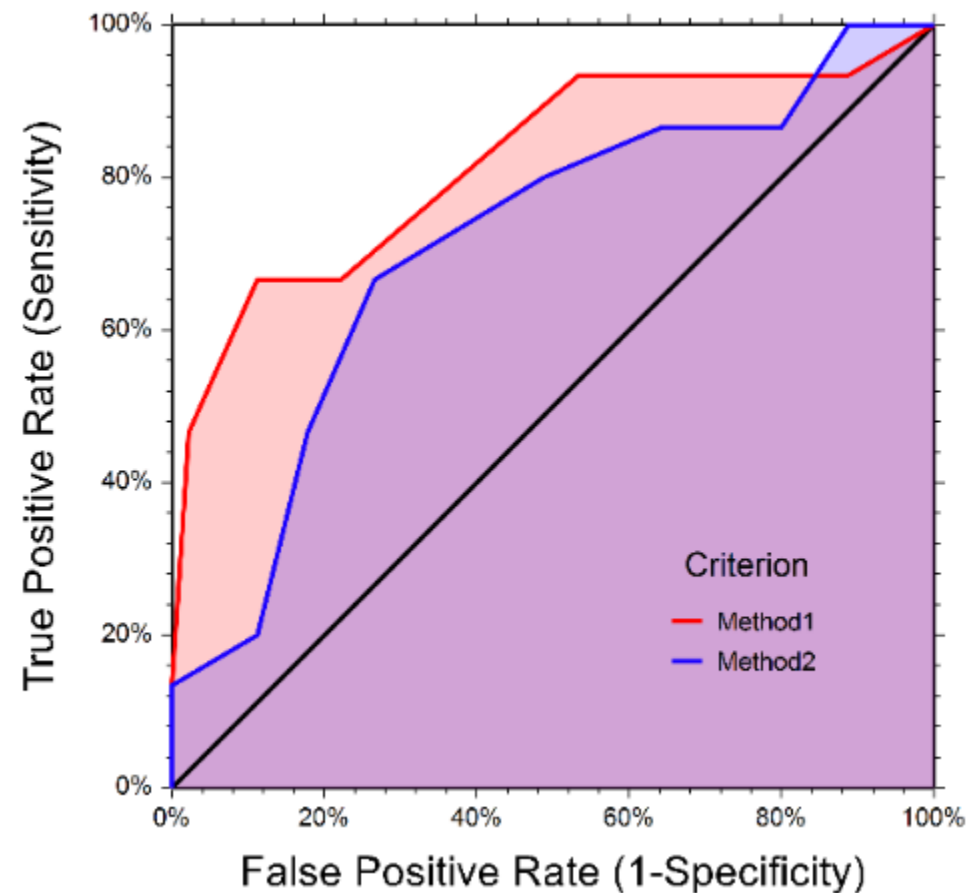
- A curva **ROC** fornece uma maneira prática de avaliar qual o melhor limiar de probabilidade para separar as classes (maior acerto, menor erro).
- É muito comum combinar a curva **ROC** com a superfície **AUC** (area under curve) em análises com múltiplos métodos.
- **AUC** tem valor máximo 1.0.
- Em geral a métrica **AUC** determina quanta dúvida o classificador apresenta:
  - Se o classificador acerta sempre com probabilidades próximas de 1 e 0 a métrica AUC tende a ser alta. *O impacto do threshold é baixo!*
  - Se o classificador mantém probabilidades sempre entre 0.25 e 0.75 para ambas as classes AUC tende a ser baixo. *O impacto do threshold é alto!*



# Medidas de avaliação

## Curva ROC, superfície AUC (ROC-AUC)

- Qual o melhor método segundo a métrica AUC?
- Qual o melhor classificador?



# Medidas de avaliação

## Curva ROC, superfície AUC (ROC-AUC)

- Qual o melhor método segundo a métrica AUC?

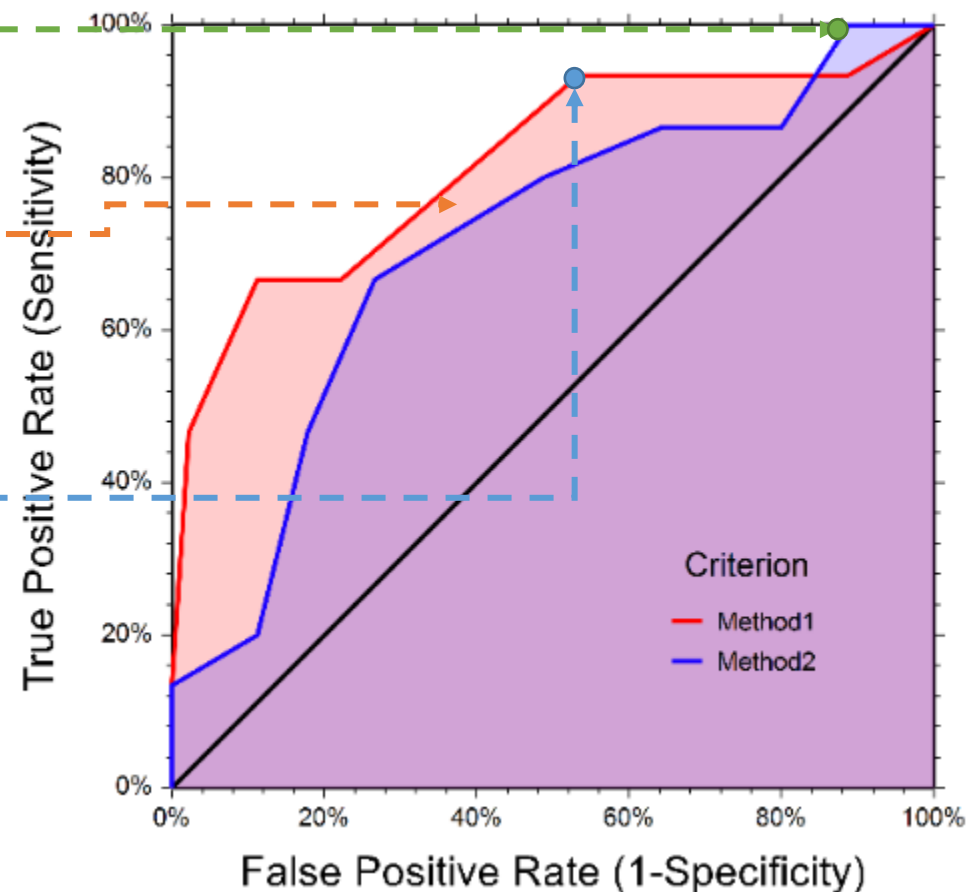
O método 1 (vermelho) apresenta a maior área

- Qual o melhor classificador?

Depende!!!

Maior acerto obtido no método azul, porém com alto erro:

Melhor balanço obtido no método Vermelho:





# Medidas de avaliação

## ROC-AUC: Python

### Código

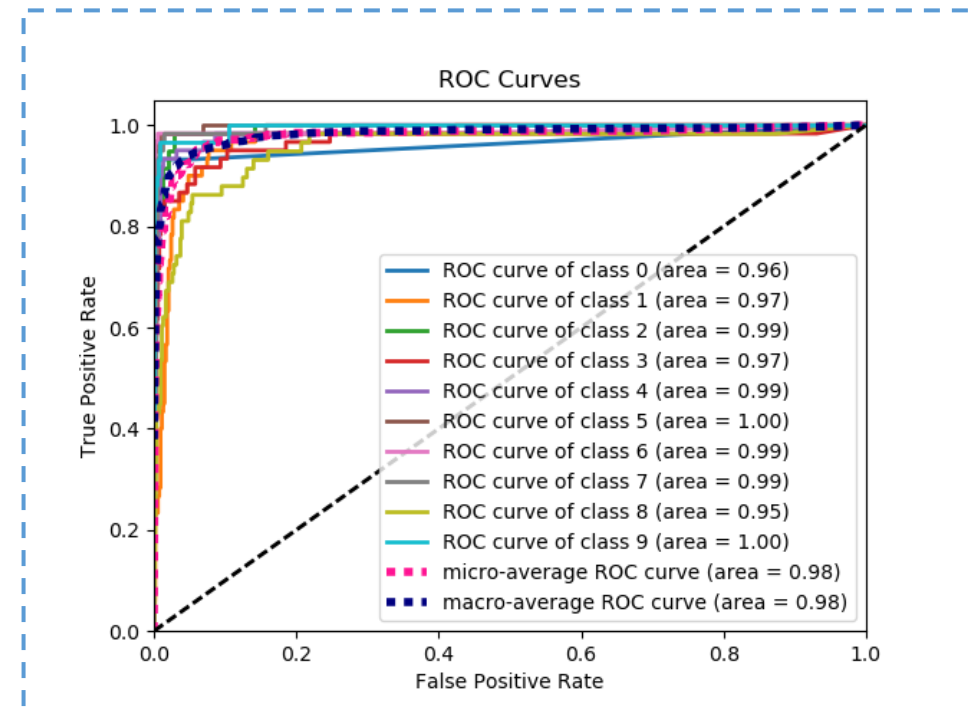
```
import scikitplot as skplt
import matplotlib.pyplot as plt

y_true = # ground truth labels
y_probab = # probabilities generated by classifier

skplt.metrics.plot_roc_curve(y_true, y_probab)

plt.show()
```

### Saída





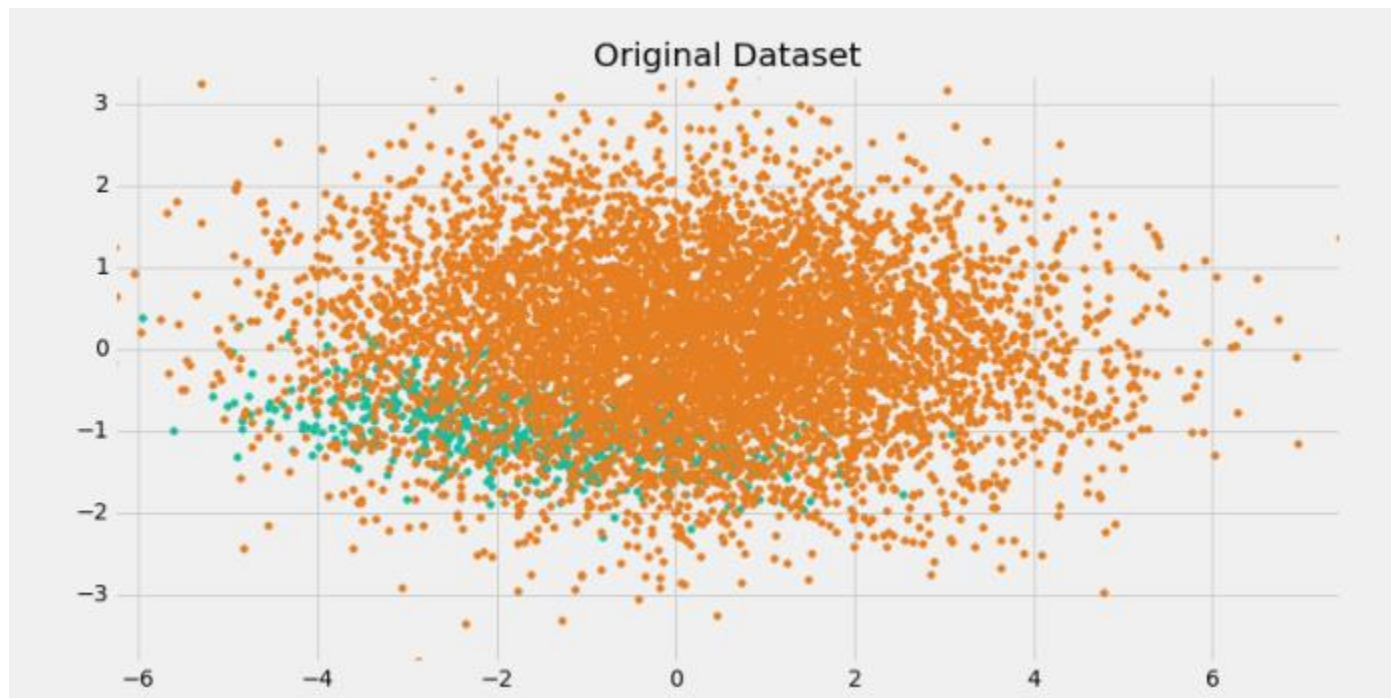
# Balanceamento:

## Tratando dados desbalanceados

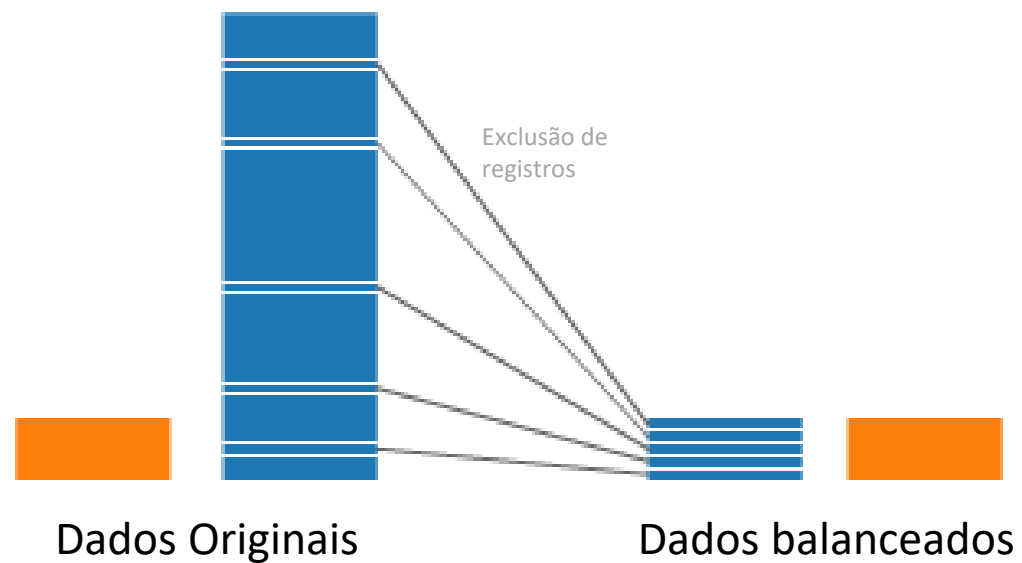
Juvenal J. Duarte

# *Dados Desbalanceados*

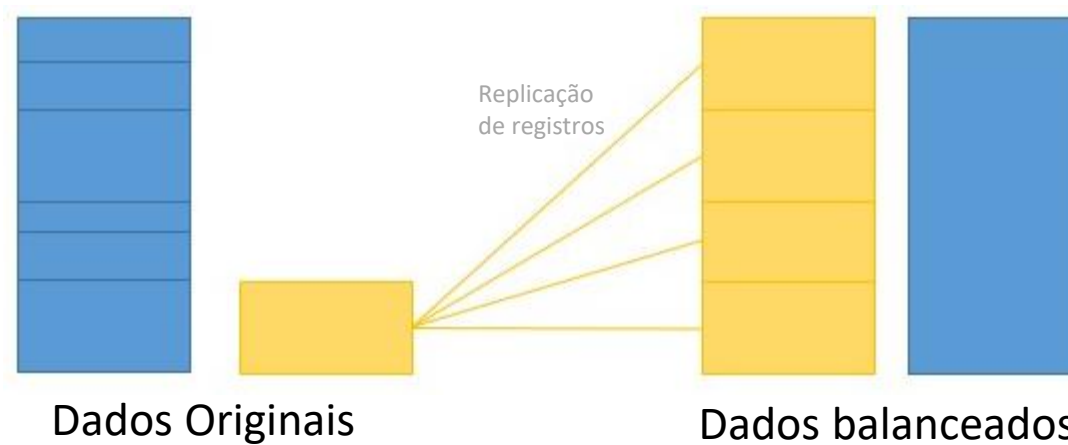
- Faz com que o algoritmo dê menor importância ou até despreze a classe minoritária durante o treinamento.



# *Sub-amostragem (Undersampling)*

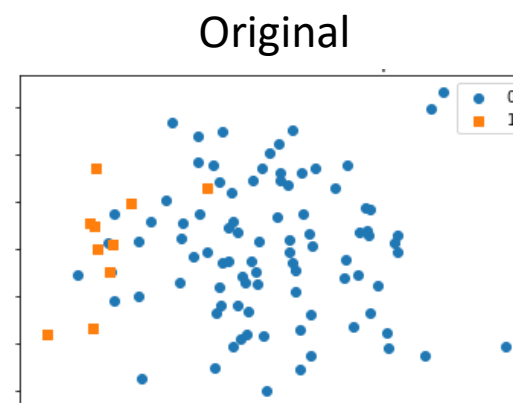


# *Sobre-amostragem (Oversampling)*

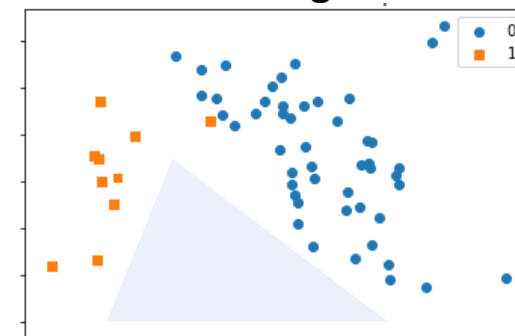


# Balanceamento de dados

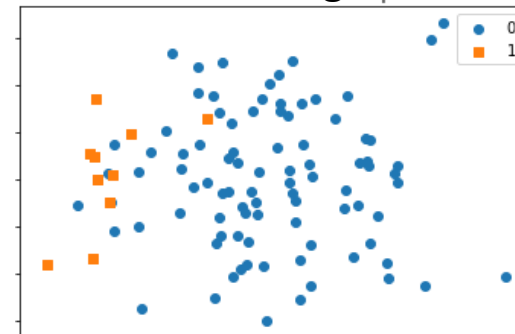
- Na prática, fazer sub ou sobre amostragem sem critérios pode trazer problemas:
  - Ao excluir dados na sub amostragem pode-se acrescentar viés na função de decisão.
  - Simplesmente replicar os pontos da classe minoritária ajuda a balancear a importância das classes no aprendizado, mas pode prejudicar a capacidade de generalização do modelo.



Sub-amostragem em 0



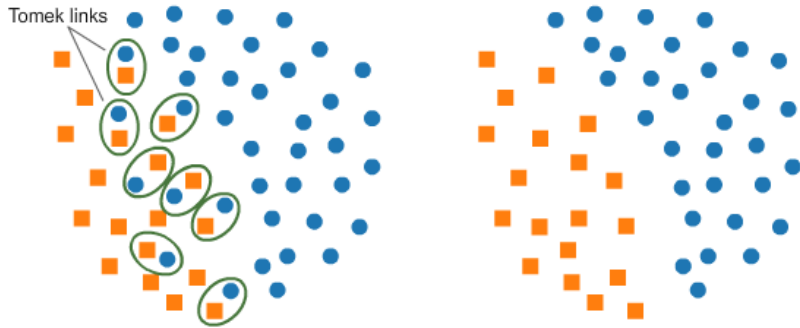
Sobre-amostragem em 1



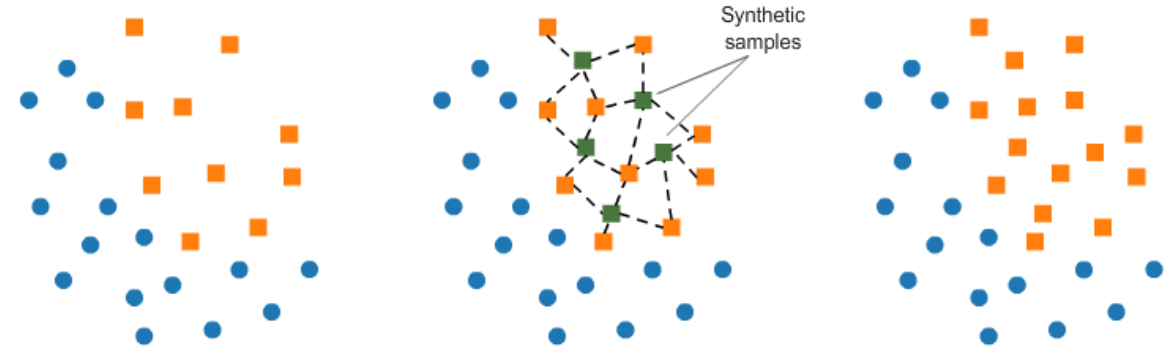
<https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets>

# Balanceamento de dados

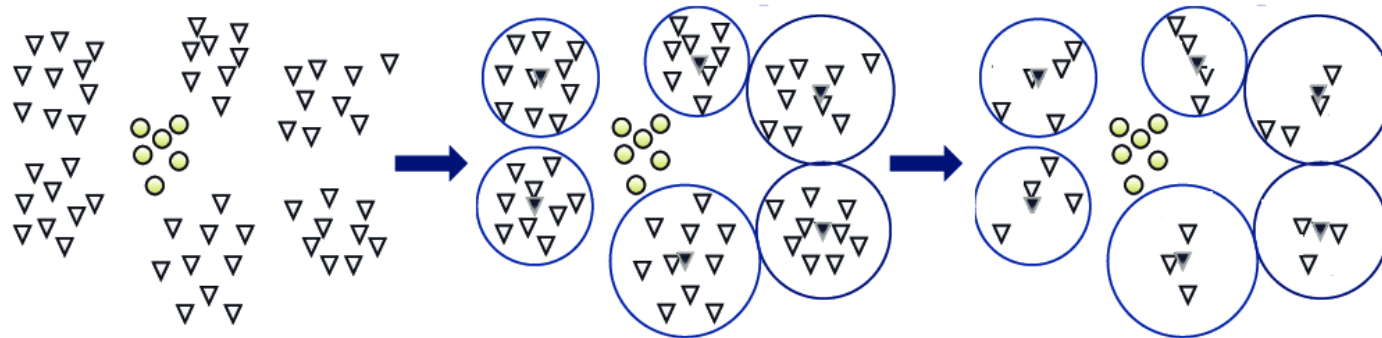
## Outras Técnicas



Under Sampling: Tomek links



Over Sampling: SMOTE (Synth. Minority Over. Technique)



Under Sampling: Cluster Centroids

<https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets>

# Balanceamento de dados

## Outras Técnicas

- Alguns classificadores conseguem tratar o desbalanceamento de classes diretamente no processo de aprendizagem. Para isto, registros da classe minoritária previstos incorretamente são punidos mais severamente.
- Métodos como LogisticRegression e DecisionTreeClassifier implementam essa funcionalidade através do parametro `class_weight="balanced"`.

**`class_weight` : dict or 'balanced', default=None**

Weights associated with classes in the form `{class_label: weight}`. If not given, all classes are supposed to have weight one.

The "balanced" mode uses the values of `y` to automatically adjust weights inversely proportional to class frequencies in the input data as `n_samples / (n_classes * np.bincount(y))`.

Note that these weights will be multiplied with `sample_weight` (passed through the fit method) if `sample_weight` is specified.

*New in version 0.17: `class_weight='balanced'`*



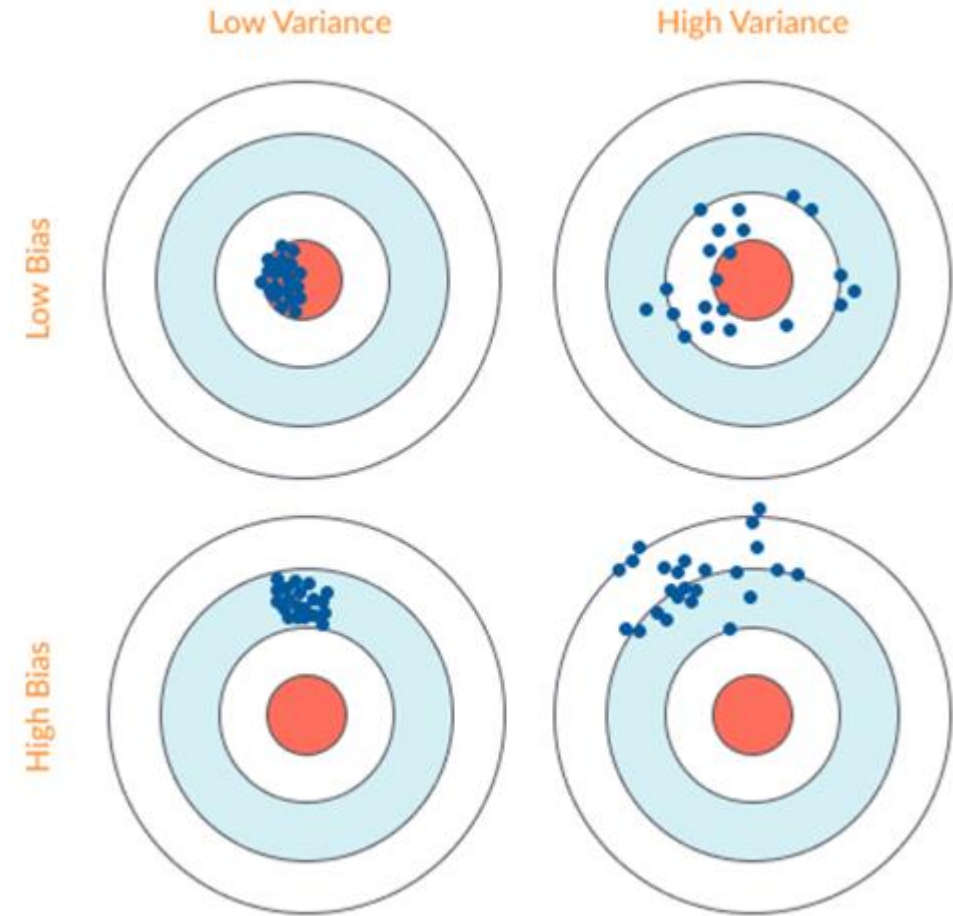
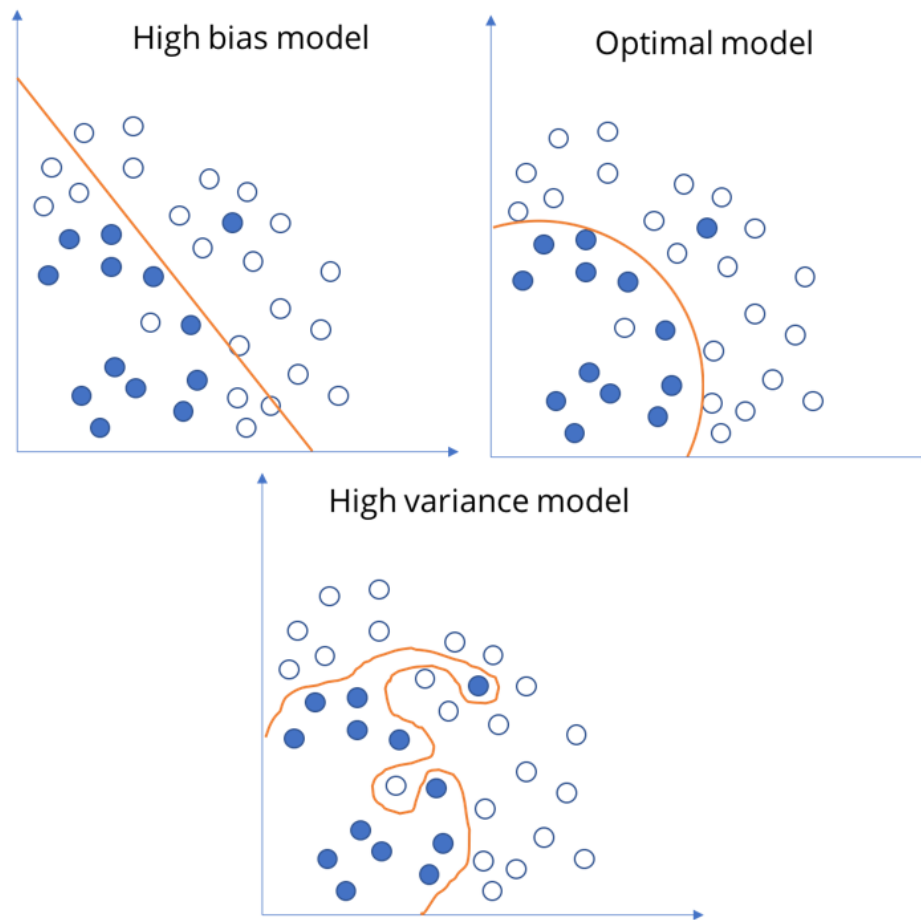


# Bias/Variance Trade-Off:

Juvenal J. Duarte

# Bias – Variance Trade-Off

O que é?



# *Bias – Variance Trade-Off*

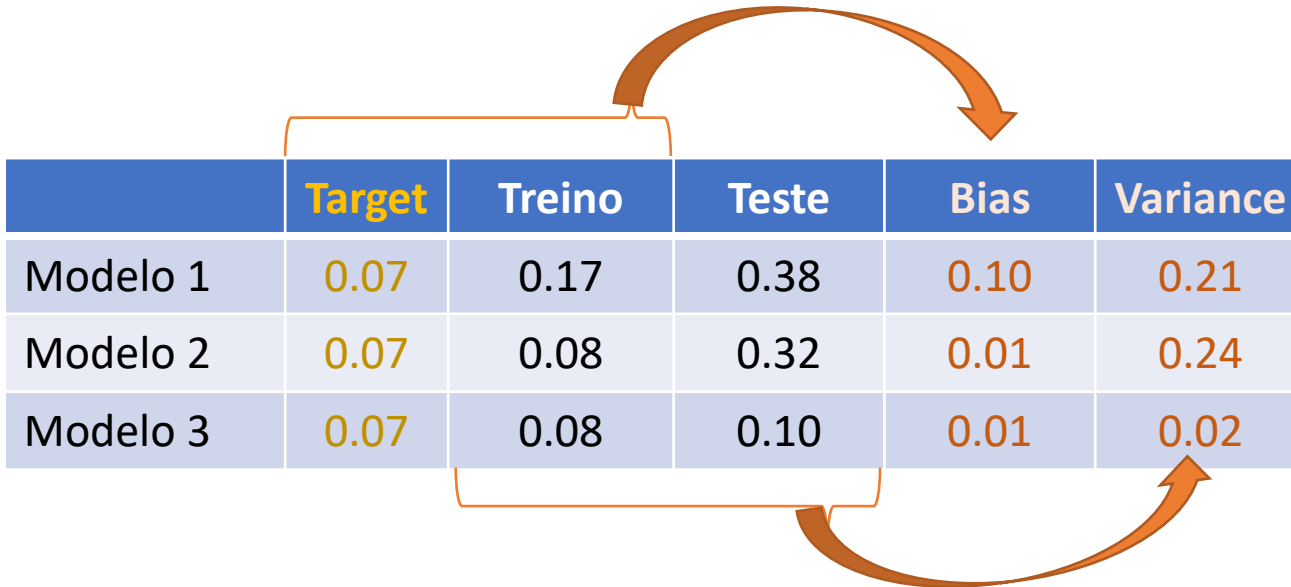
*Como avaliar?*

*Recall Error*

	<b>Target</b>	<b>Treino</b>	<b>Teste</b>
Modelo 1	0.07	0.17	0.38
Modelo 2	0.07	0.08	0.32
Modelo 3	0.07	0.08	0.10

# Bias – Variance Trade-Off

*Como avaliar?*

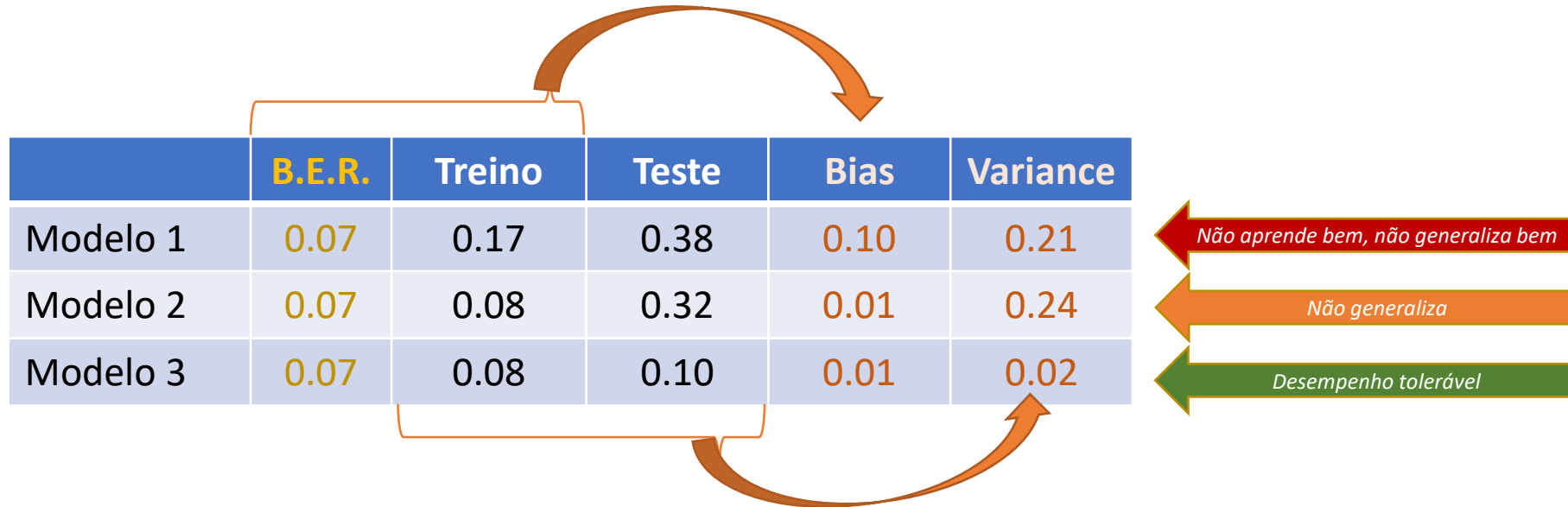


	Target	Treino	Teste	Bias	Variance
Modelo 1	0.07	0.17	0.38	0.10	0.21
Modelo 2	0.07	0.08	0.32	0.01	0.24
Modelo 3	0.07	0.08	0.10	0.01	0.02

Qual o  
melhor  
modelo?

# Bias – Variance Trade-Off

Como avaliar?



The diagram illustrates the Bias-Variance Trade-Off for three models. A table shows the Bias and Variance values for each model. Arrows indicate the relationship between Bias and Variance for each model: Model 1 has high Bias and high Variance, Model 2 has low Bias and high Variance, and Model 3 has low Bias and low Variance. The arrows show that as Bias decreases, Variance increases, and vice versa.

	B.E.R.	Treino	Teste	Bias	Variance
Modelo 1	0.07	0.17	0.38	0.10	0.21
Modelo 2	0.07	0.08	0.32	0.01	0.24
Modelo 3	0.07	0.08	0.10	0.01	0.02

Modelo 1: Não aprende bem, não generaliza bem

Modelo 2: Não generaliza

Modelo 3: Desempenho tolerável



# Hiper-parâmetros:

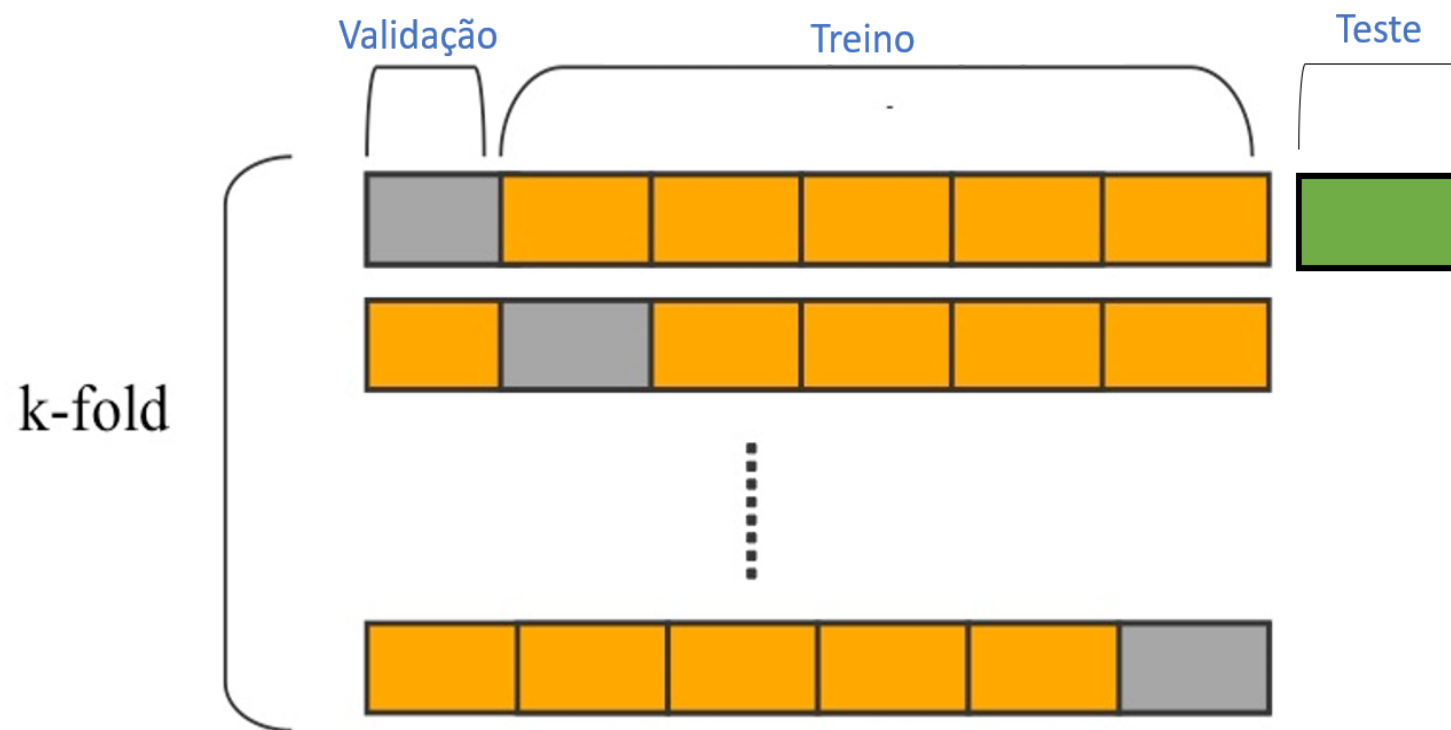
Validação cruzada + Busca em grid

Juvenal J. Duarte

# Validação Cruzada:

## K-Fold

- Validação cruzada: método K-Fold.
  - Subdivide o dataset em K porções.
  - Executa K iterações de treinamento e validação de modelos, sendo que a cada iteração uma partição é escolhida para validação e todas as demais são usadas no treinamento.
  - Os melhores hiper-parâmetros são escolhidos baseado no melhor resultado médio.



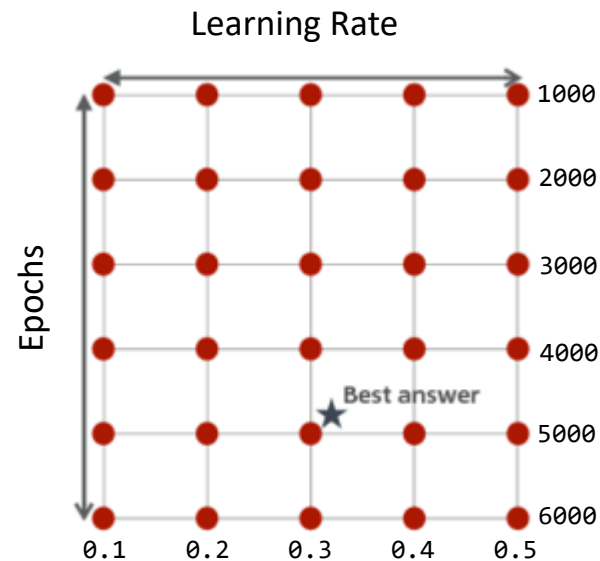
# Grid Search

- Busca exaustiva pelos melhores parâmetros para o modelo. Deve ser associada a uma estratégia de validação: hold-out, k-fold etc.
- É necessário estabelecer os valores de cada parâmetro a serem testados. Exemplo no SKLearn:

```
GridSearchCV(cv=None,
             estimator=MLPClassifier(),
             param_grid={'learning_rate': [0.1, 0.2, 0.3, 0.4, 0.5],
                        'max_iter': [1000, 2000, 3000, 4000, 5000, 6000]})
```

- São testadas todas as combinações de valores para os hiperparâmetros (produto vetorial), o que leva a um custo computacional elevado.

[https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.GridSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)



Para cada combinação de parâmetros execute a validação...

