MODEL LIFECYCLE

CRIANDO MODELOS SIMPLES DE MACHINE LEARNING I

DIEGO RODRIGUES DSC

INFNET

MODEL LIFECYCLE : CRIANDO MODELOS SIMPLES DE MACHINE LEARNING I

PARTE 1 : TEORIA

- BUSINESS UNDERSTANDING
  - DATASET IRIS
  - PAIRPLOT
- DATA UNDERSTANDING
  - VARIÁVEIS ALEATÓRIAS
- DATA PREPARATION
  - STANDARD SCALER
- MODELING
  - SELEÇÃO DE ATRIBUTOS
    - BOXPLOT
  - REPRODUTIBILIDADE
    - RANDOM SEED
  - SELEÇÃO DO MODELO
    - VIZINHOS MAIS PRÓXIMOS
- EVALUATION
  - SINGLE SPLIT
    - TRAIN TEST SPLIT
  - ACURÁCIA
    - ACCURACY SCORE
  - HIPERPARÂMETRO

PARTE 2 : PRÁTICA

- NOTEBOOK
  CLASSIFICAÇÃO IRIS KNN

Produzir Ação

CICLO DE VIDA DO MODELO

Baseado em Dados

AMBIENTE PYTHON

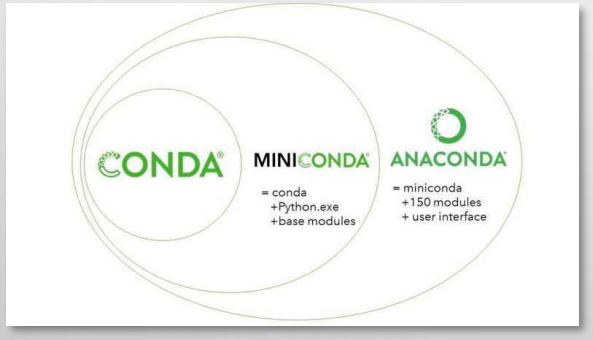
5. Visualização

6. Estimação e Inferência

7. Machine Learning

4. Variáveis Aleatórias

1. Editor de Código



2. Gestor de Ambiente

3. Ambiente Python do Projeto

3. Notebook Dinâmico

Cross Industry Standard Process for Data Mining - IBM

1) Requerimentos e Análise de Negócio

Entendimento do problema decisório, dados relacionados & revisão bibliográfica.

2) Preparação dos Dados

Entendimento das fontes de dados, dos tipos e elaboração da representação.

3) Modelagem

Análise Exploratória, Seleção de atributos e treinamento.

4) Avaliação

Seleção do melhor modelo.

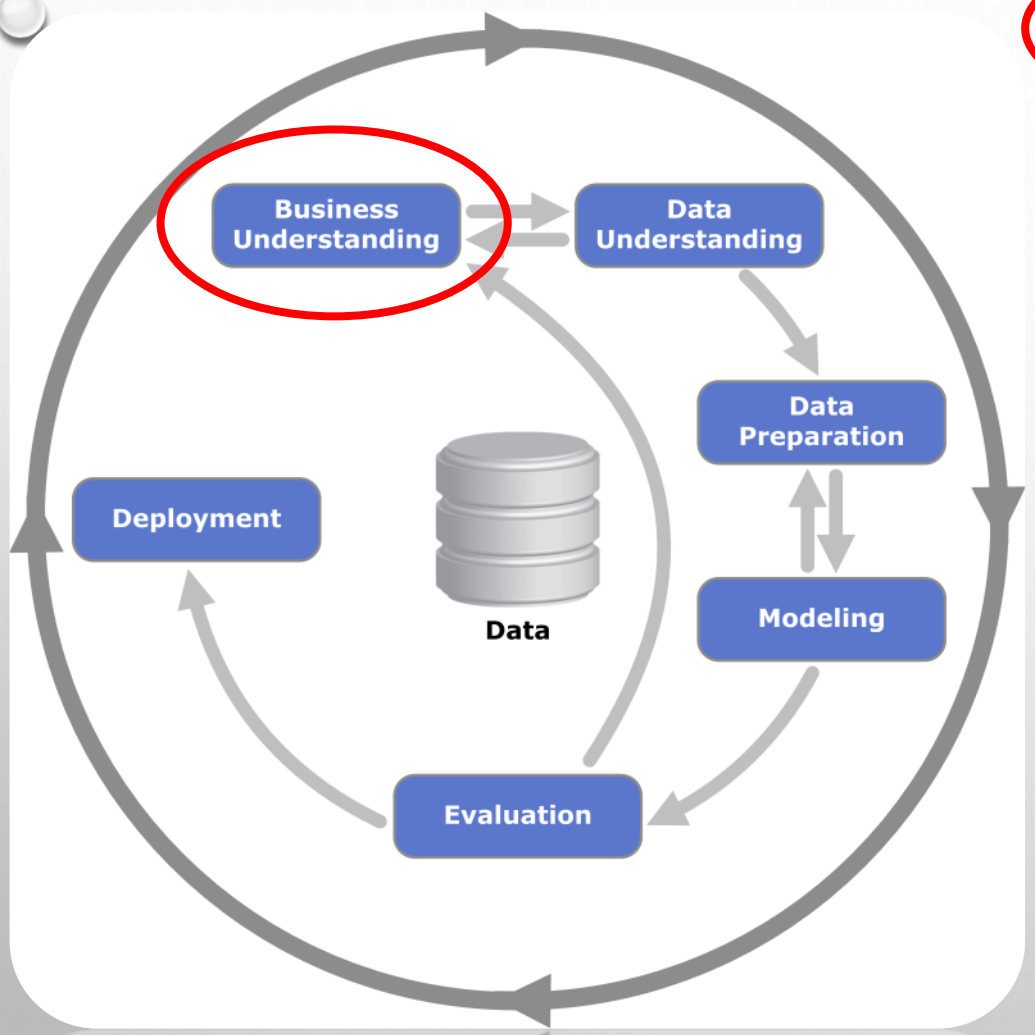
5) Liberação

Liberação do modelo no ambiente de produção.

BUSINESS UNDERSTANDING

CRISP-DM

Cross Industry Standard Process for Data Mining - IBM



1) Requerimentos e Análise de Negócio

Entendimento do problema decisório, dados relacionados & revisão bibliográfica.

2) Preparação dos Dados

Entendimento das fontes de dados, dos tipos, análise exploratória e representação.

3) Modelagem

Seleção, extração de atributos e treinamento do modelo.

4) Avaliação

Seleção do melhor modelo.

5) Liberação

Liberação do modelo no ambiente de produção.

PROBLEMA DE NEGÓCIO

Características das flores

Largura & comprimento da pétala

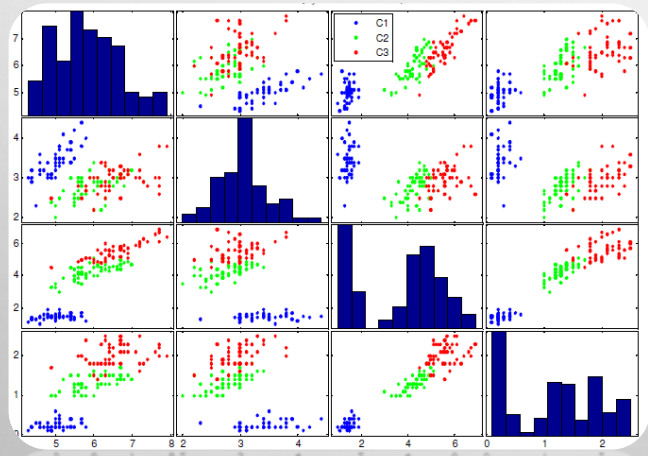
Largura & comprimento da sépala

Iris Setosa

Iris Versicolor

Iris Virginica

REPRESENTAÇÃO



http://archive.ics.uci.edu/ml/datasets/Iris

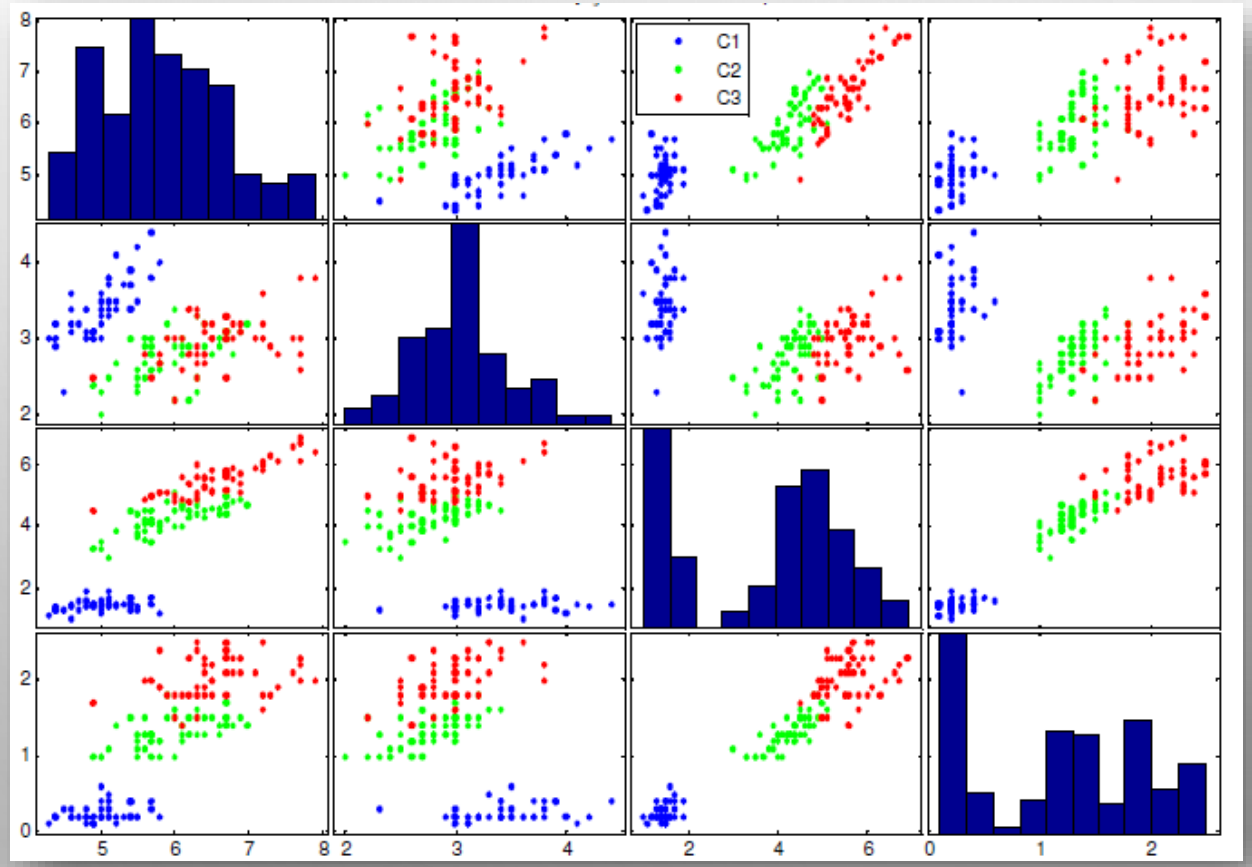
Características das flores

Largura & comprimento da pétala

Largura & comprimento da sépala

Espaço de atributos com **4 dimensões!**

QUAIS ATRIBUTOS UTILIZAR?



Para separar a Iris Setosa (azul)?

Para separar Iris Virginica (Vermelha)?

Para encontrar corretamente 3 grupos de flores?

PAIRPLOT

seaborn.pairplot

```
seaborn.pairplot(data, *, hue=None, hue_order=None, palette=None, vars=None, x_vars=None,
y_vars=None, kind='scatter', diag_kind='auto', markers=None, height=2.5, aspect=1,
corner=False, dropna=False, plot_kws=None, diag_kws=None, grid_kws=None, size=None)
```

Plot pairwise relationships in a dataset.

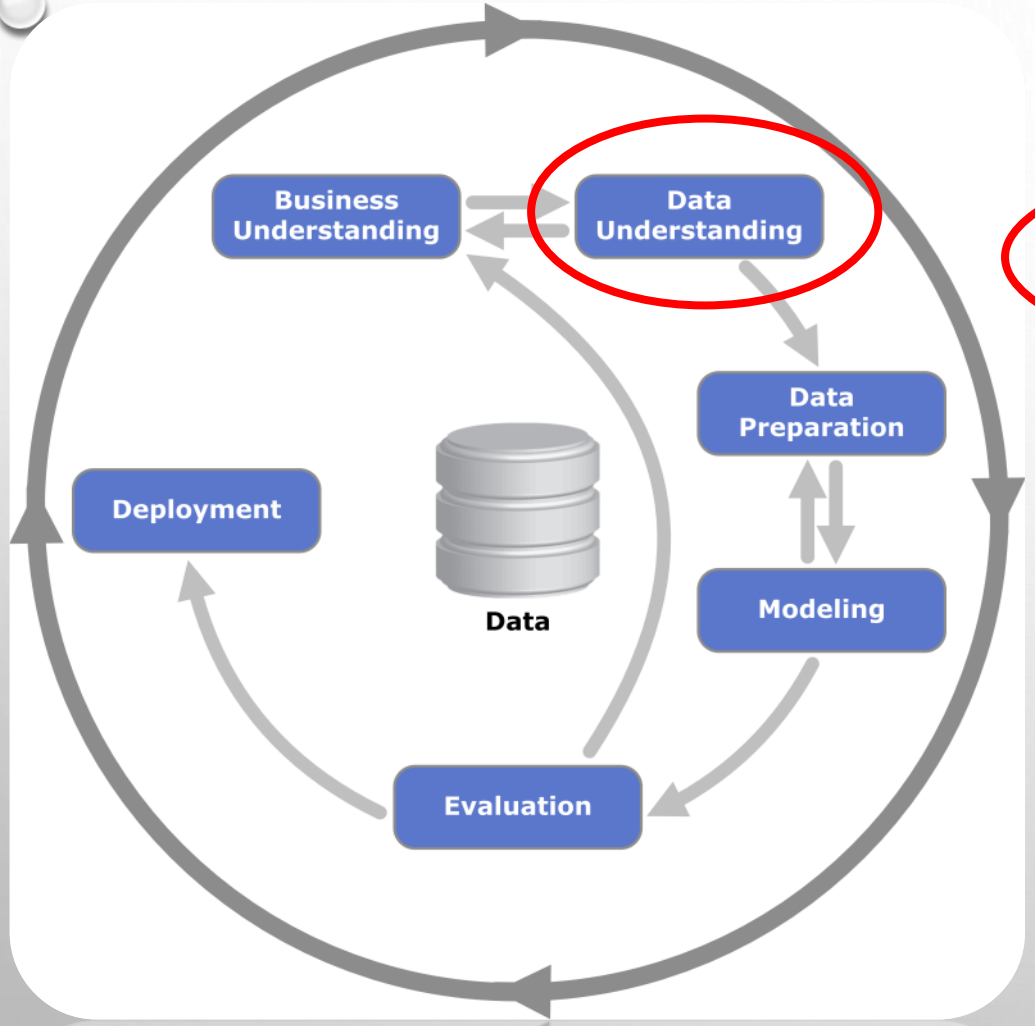
By default, this function will create a grid of Axes such that each numeric variable in `data` will by shared across the y-axes across a single row and the x-axes across a single column. The diagonal plots are treated differently: a univariate distribution plot is drawn to show the marginal distribution of the data in each column.

It is also possible to show a subset of variables or plot different variables on the rows and columns.

DATA UNDERSTANDING

CRISP-DM

Cross Industry Standard Process for Data Mining - IBM



1) Requerimentos e Análise de Negócio

Entendimento do problema decisório, dados relacionados & revisão bibliográfica.

2) Preparação dos Dados

Entendimento das fontes de dados, dos tipos, análise exploratória e representação.

3) Modelagem

Seleção, extração de atributos e treinamento do modelo.

4) Avaliação

Seleção do melhor modelo.

5) Liberação

Liberação do modelo no ambiente de produção.

QUAIS SÃO OS TIPOS MAIS COMUNS DE ATRIBUTOS?

Nominal ou Categórica

- Conjunto de diferentes valores não ordenados.
- Exemplo: Sexo, cor, palavras, tipo de coisas.

string / bool

Ordinal

- Conjunto ordenado, mas a diferença entre os valores não tem significado.
- Exemplo: scores quantitativos como “excelente”, “bom”, “regular”, “ruim”.

string / int

Intervalo

- Conjunto ordenado, a diferença tem significado mas não as proporções.
- Exemplo: Datas.

datetime

Ratio

- Conjunto ordenado onde diferenças & proporções tem significado.
- Exemplo: Idade, peso, altura, dinheiro, massa, etc.

int / float

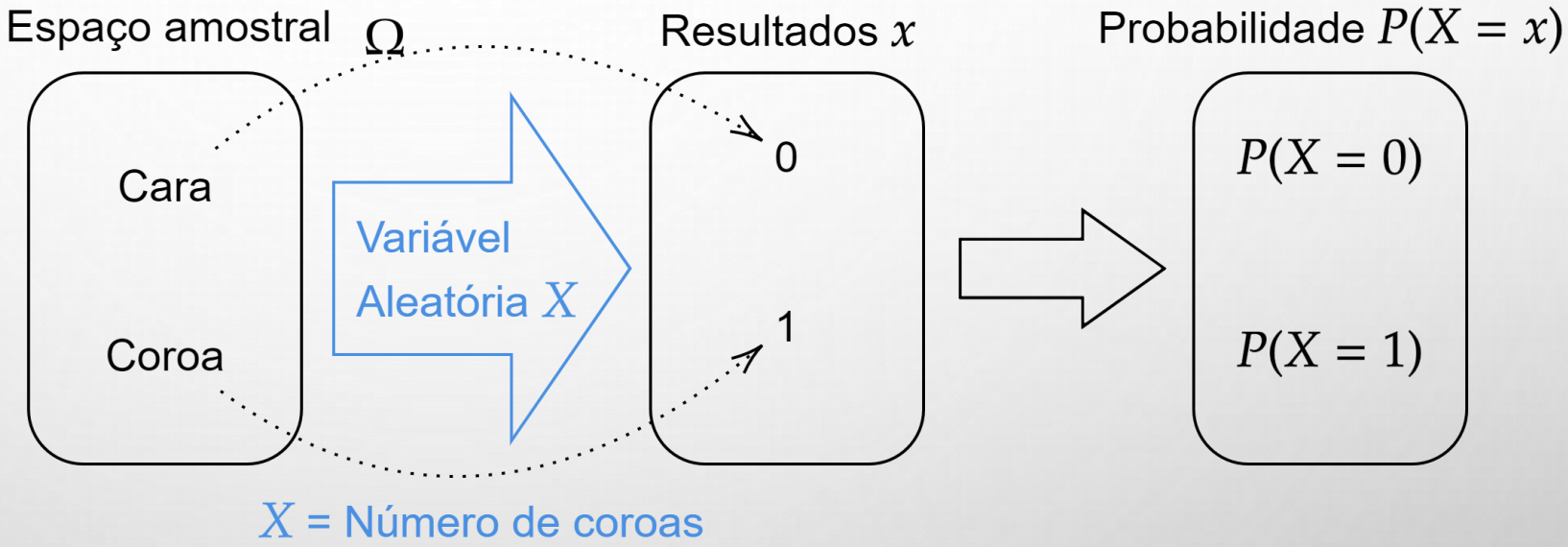
Texto

- Sequência de palavras de tamanho finito.
- Exemplo: “Ontem eu fui passear”.

string

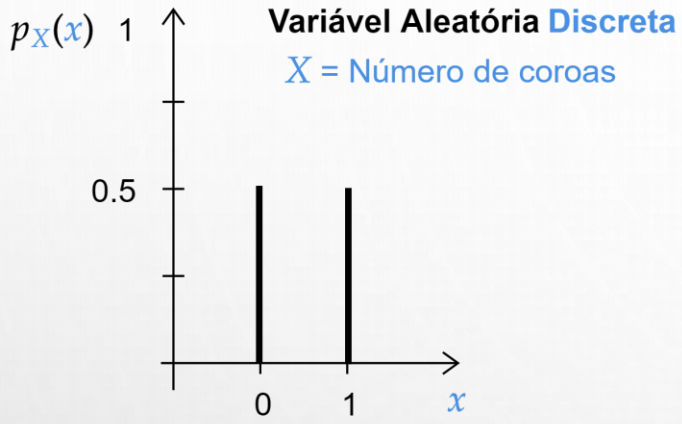
VARIÁVEIS ALEATÓRIAS

VARIÁVEL ALEATÓRIA

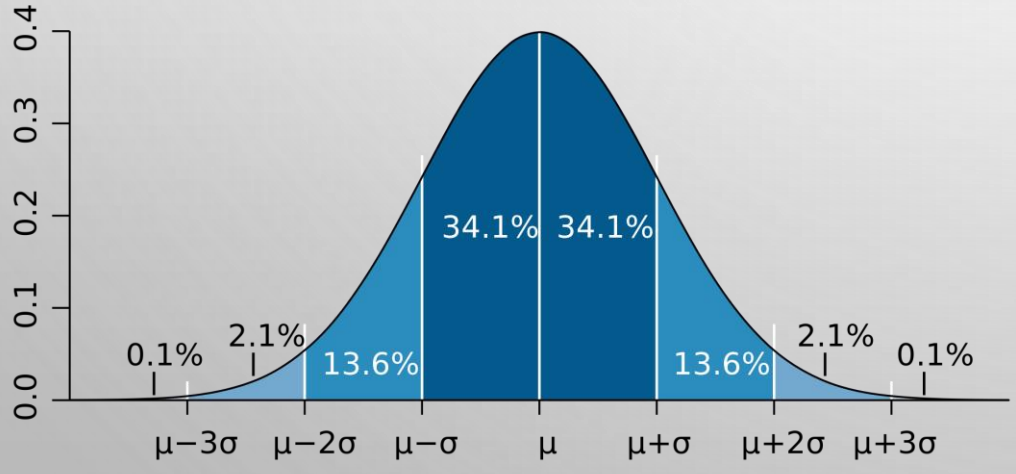


https://www.inf.ufsc.br/~andre.zibetti/probabilidade/variaveis_aleatorias.html[1]

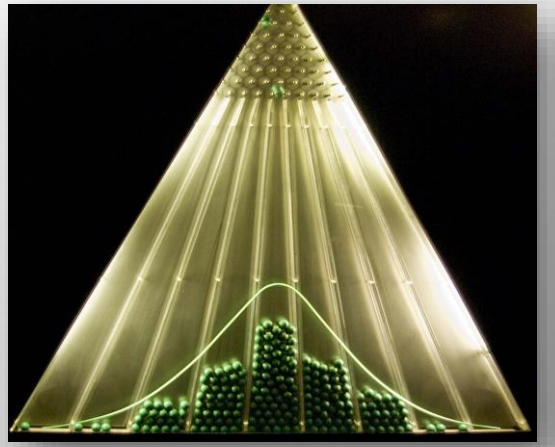
DISTRIBUIÇÃO DE PROBABILIDADE



Zibetti [1]

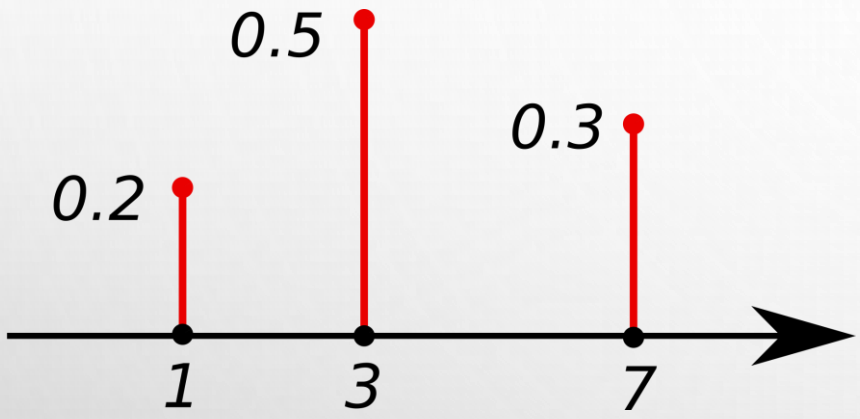


Wikipedia [2]



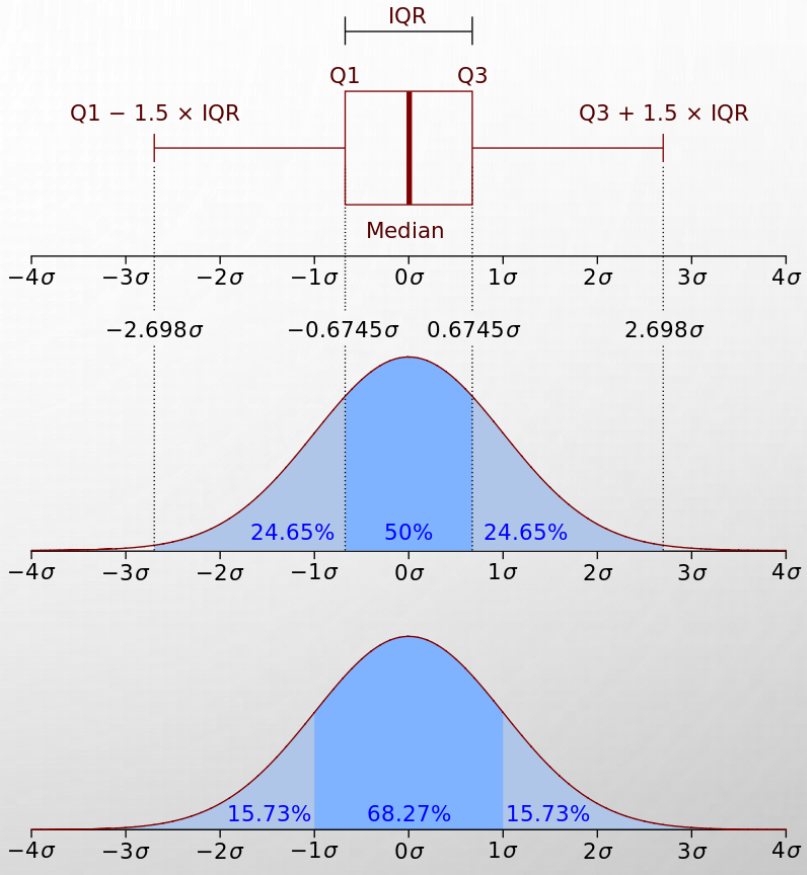
Distribuição teórica, paramétrica, e a realização experimental, não paramétrica.

FUNÇÃO DENSIDADE DE PROBABILIDADE



Função Massa de Probabilidade (Discreta)

[2]

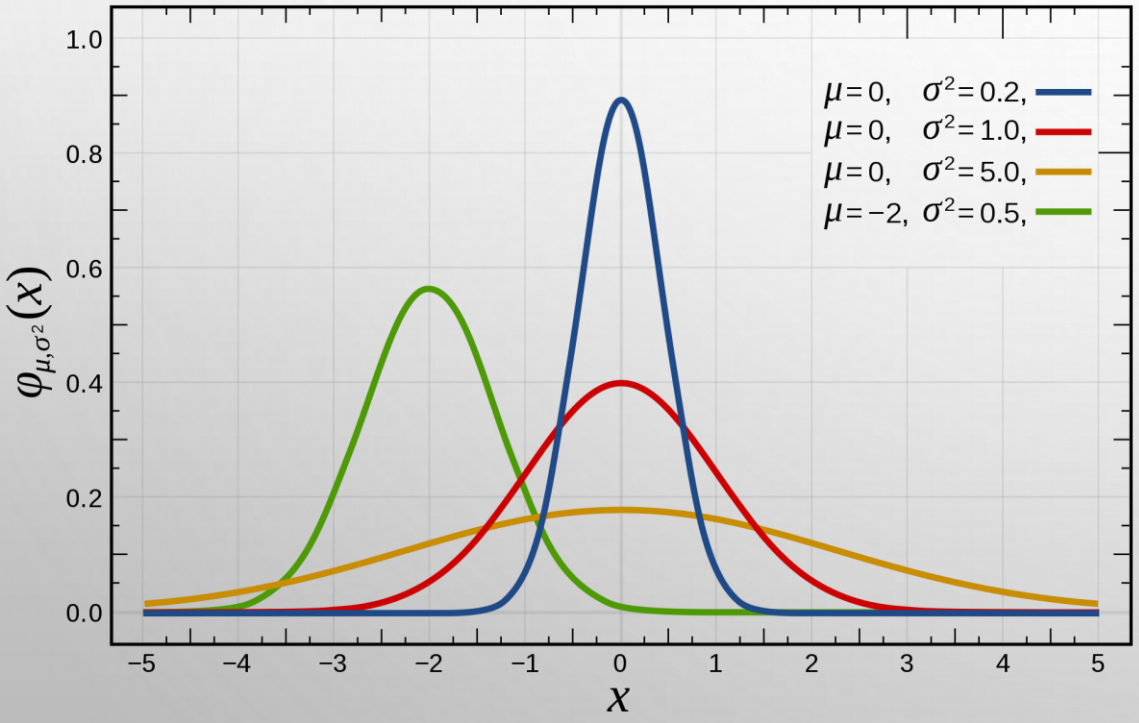


Função Densidade de Probabilidade (Contínua)

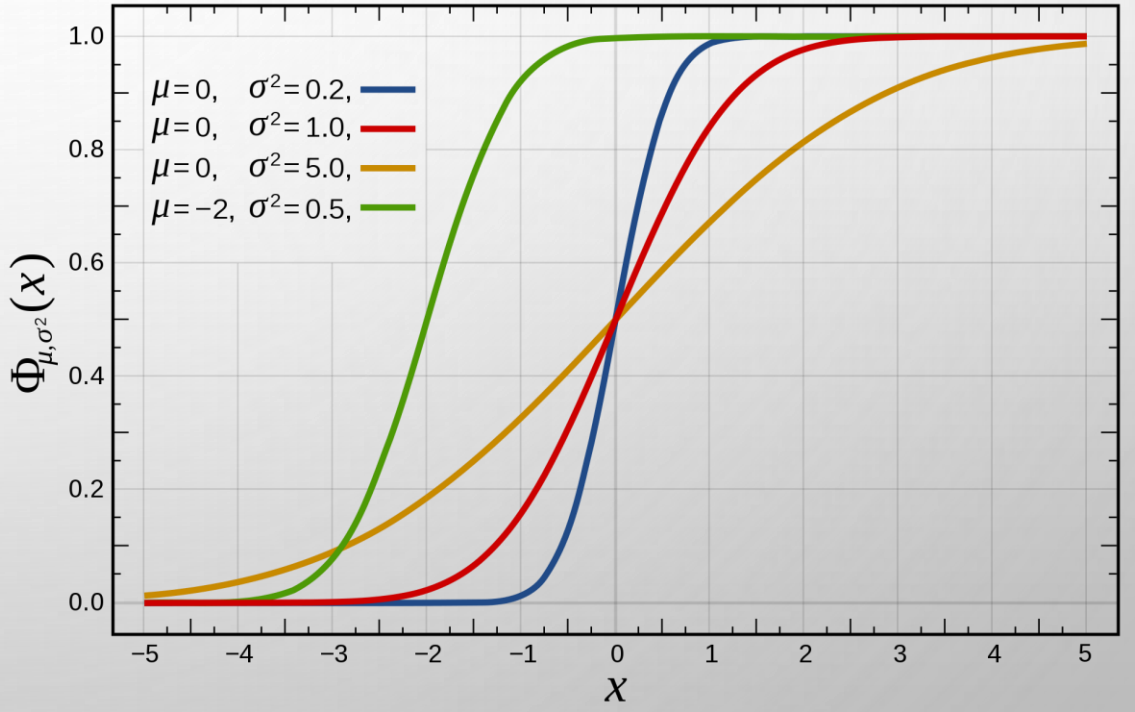
[2]

FUNÇÃO DISTRIBUIÇÃO DE PROBABILIDADE ACUMULADA

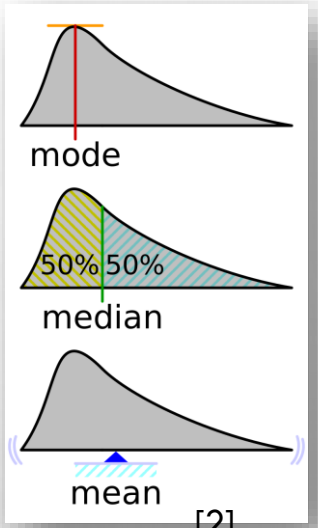
$$\mathrm{F}(x) = P(X \leq x)$$



$\int \varphi$



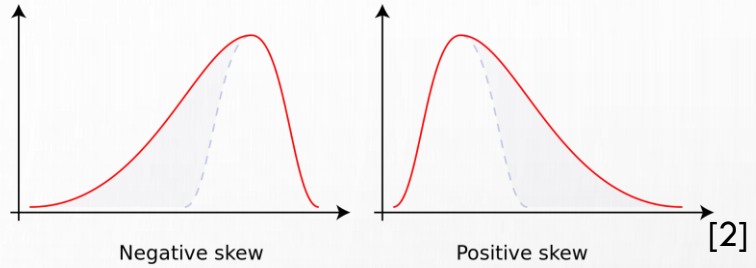
ESTATÍSTICAS DE UMA DISTRIBUIÇÃO



[2]

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

MÉDIA – O MOMENTO CENTRAL

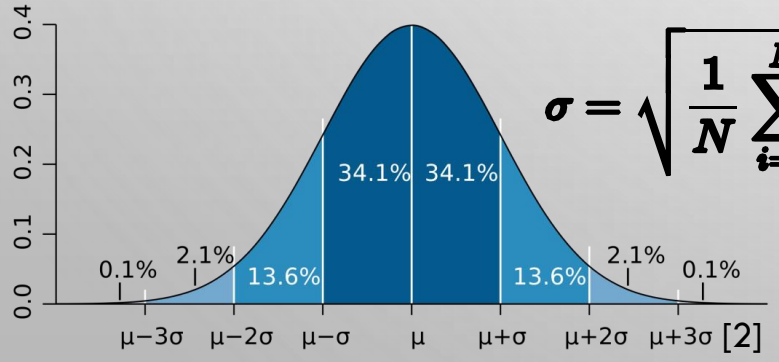


[2]

$$\tilde{\mu}_3 = \mathrm{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] = \frac{\mathrm{E}[X^3] - 3\mu\,\mathrm{E}[X^2] + 3\mu^2\,\mathrm{E}[X] - \mu^3}{\sigma^3} = \frac{\mathrm{E}[X^3] - 3\mu(\mathrm{E}[X^2] - \mu\,\mathrm{E}[X]) - \mu^3}{\sigma^3} = \frac{\mathrm{E}[X^3] - 3\mu\sigma^2 - \mu^3}{\sigma^3}.$$

ASSIMETRIA - DESEQUILIBRIO

O DESVIO PADRÃO – DISPERSÃO DOS DADOS



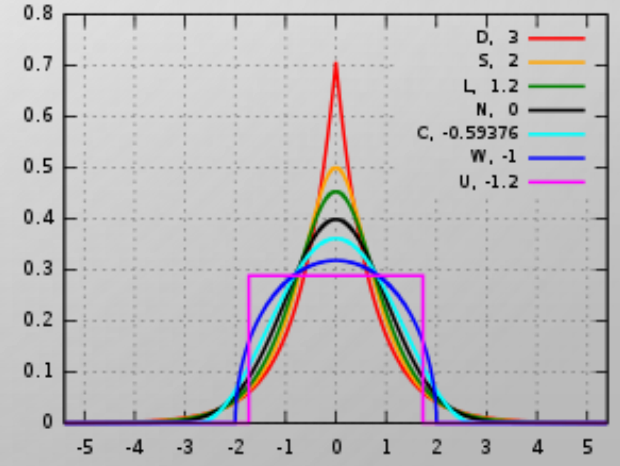
[2]

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

CURTOSE - HOMOGENEIDADE

$$g_2 = \frac{m_4}{m_2^2} - 3 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{\left[\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right]^2} - 3$$

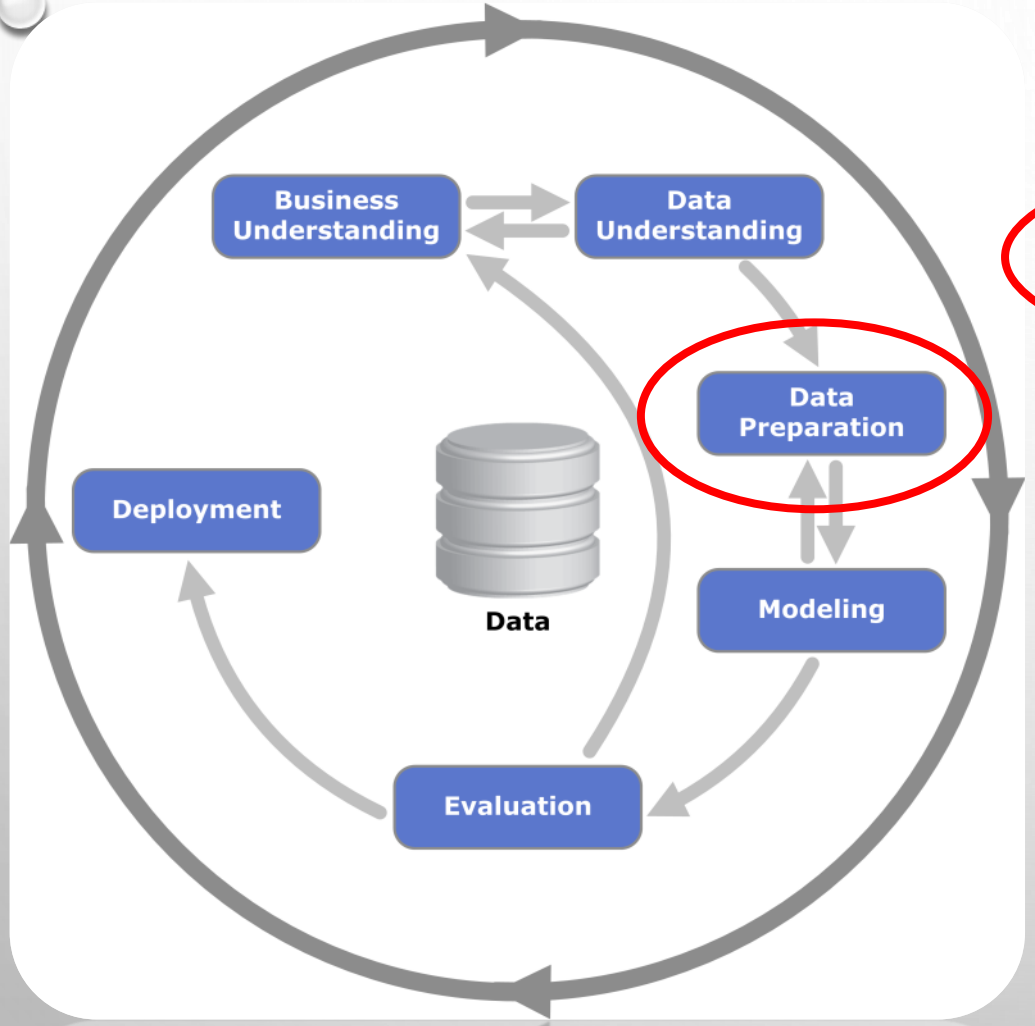
[2]



DATA PREPARATION

CRISP-DM

Cross Industry Standard Process for Data Mining - IBM



1) Requerimentos e Análise de Negócio

Entendimento do problema decisório, dados relacionados & revisão bibliográfica.

2) Preparação dos Dados

Entendimento das fontes de dados, dos tipos, análise exploratória e representação.

3) Modelagem

Seleção, extração de atributos e treinamento do modelo.

4) Avaliação

Seleção do melhor modelo.

5) Liberação

Liberação do modelo no ambiente de produção.

DATA PREPARATION

Quantificação dos Atributos

- Transformar todos os atributos em atributos numéricos.

Escalonamento

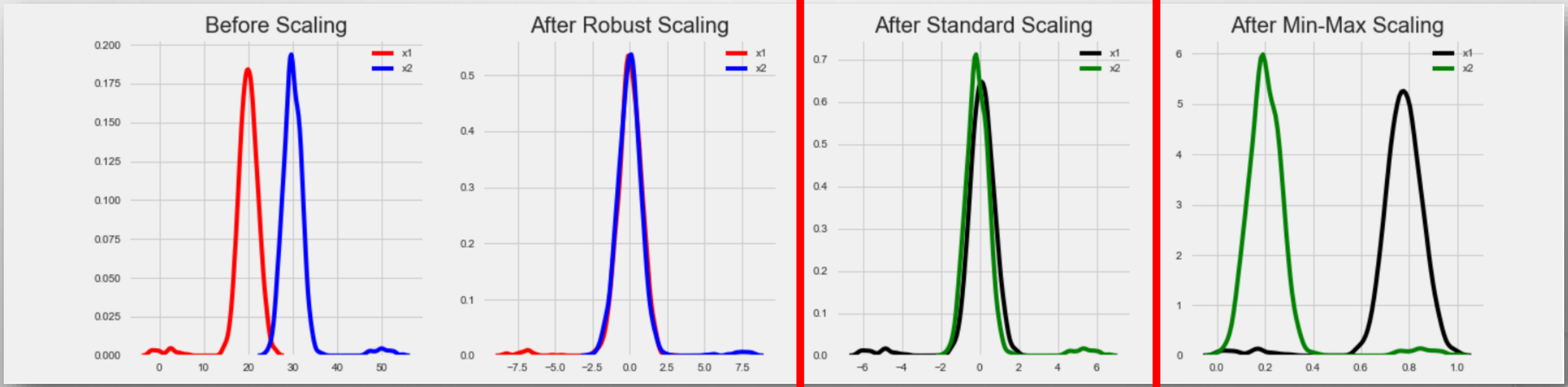
- Transformar todos os atributos para a mesma faixa dinâmica, de maneira a assegurar que todos tenham o mesmo "peso numérico" para o treinamento do modelo.

Normalização

- Garantir que os dados tenham uma distribuição de probabilidade gaussiana (Normal).

ESCALONAMENTO E NORMALIZAÇÃO

- Garantir que as variáveis possuam a mesma escala
- Mesmo efeito numérico na otimização independente da escala.



STANDARD SCALER

StandardScaler

class sklearn.preprocessing.**StandardScaler**(*, *copy*=*True*, *with_mean*=*True*, *with_std*=*True*) [source]

Standardize features by removing the mean and scaling to unit variance.

The standard score of a sample `x` is calculated as:

```
z = (x - u) / s
```

where `u` is the mean of the training samples or zero if `with_mean=False`, and `s` is the standard deviation of the training samples or one if `with_std=False`.

Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Mean and standard deviation are then stored to be used on later data using `transform`.

Standardization of a dataset is a common requirement for many machine learning estimators: they might behave badly if the individual features do not more or less look like standard normally distributed data (e.g. Gaussian with 0 mean and unit variance).

For instance many elements used in the objective function of a learning algorithm (such as the RBF kernel of Support Vector Machines or the L1 and L2 regularizers of linear models) assume that all features are centered around 0 and have variance in the same order. If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected.

`StandardScaler` is sensitive to outliers, and the features may scale differently from each other in the presence of outliers. For an example visualization, refer to Compare StandardScaler with other scalers.

MODELING

CRISP-DM

Cross Industry Standard Process for Data Mining - IBM

1) Requerimentos e Análise de Negócio

Entendimento do problema decisório, dados relacionados & revisão bibliográfica.

2) Preparação dos Dados

Entendimento das fontes de dados, dos tipos, análise exploratória e representação.

3) Modelagem

Seleção, extração de atributos e treinamento do modelo.

4) Avaliação

Seleção do melhor modelo.

5) Liberação

Liberação do modelo no ambiente de produção.

MODELING

Seleção de Atributos

- Quantificar e ordenar os atributos por importância para o problema.
- Eliminar atributos irrelevantes.

Extração de Atributos

- Transformar os atributos do espaço original para um espaço que favoreça a modelagem.

Treinamento

- Encontrar os hiper-parâmetros e parâmetros do modelo, para os dados disponíveis, avaliando a figura de mérito selecionada.

SELEÇÃO DE ATRIBUTOS

TÉCNICAS DE SELEÇÃO DE ATRIBUTOS

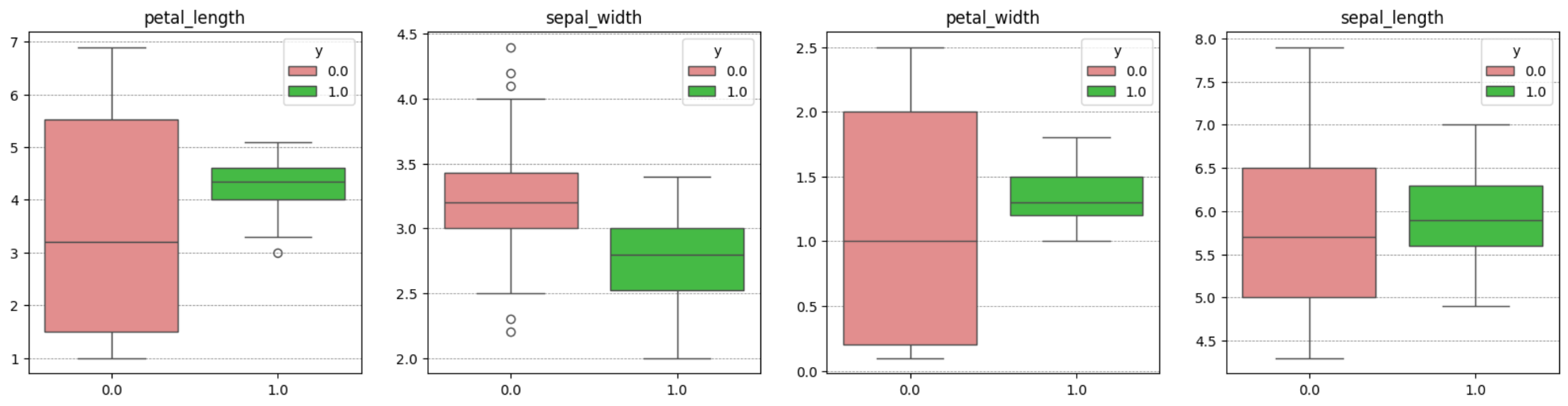
Filtragem – mede a relação entre atributos ou atributos e classes, utilizando estatísticas, sem depender do modelo.

- **Coeficiente de Correlação de Pearson** – Estatística que mede a relação linear entre duas variáveis aleatórias.
- **Teste T de diferença de médias** – Informa se a média de um determinado atributo muda de acordo com uma categoria binária.
- **ANOVA** – O mesmo que o teste T, mas serve para múltiplas categoria.
- **Informação Mútua** – Estatística que mede relação não-linear entre duas variáveis aleatórias.

Wrapper – mede a relação entre atributos e classes, utilizando um modelo treinado.

- **Gini –** Estatística que representa a importância de um atributo na divisão de base de dados por uma árvore de decisão.
- **Relevância** – Estatística que representa a variação causada na saída do modelo quando um atributo é substituído por sua média.

COMPARAÇÃO DE MEDIANAS



BOXPLOT

seaborn.boxplot

```
seaborn.boxplot(data=None, *, x=None, y=None, hue=None, order=None, hue_order=None,
orient=None, color=None, palette=None, saturation=0.75, fill=True, dodge='auto', width=0.8,
gap=0, whis=1.5, linecolor='auto', linewidth=None, fliersize=None, hue_norm=None,
native_scale=False, log_scale=None, formatter=None, legend='auto', ax=None, **kwargs)
```

Draw a box plot to show distributions with respect to categories.

A box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable. The box shows the quartiles of the dataset while the whiskers extend to show the rest of the distribution, except for points that are determined to be “outliers” using a method that is a function of the inter-quartile range.

SELEÇÃO DO MODELO

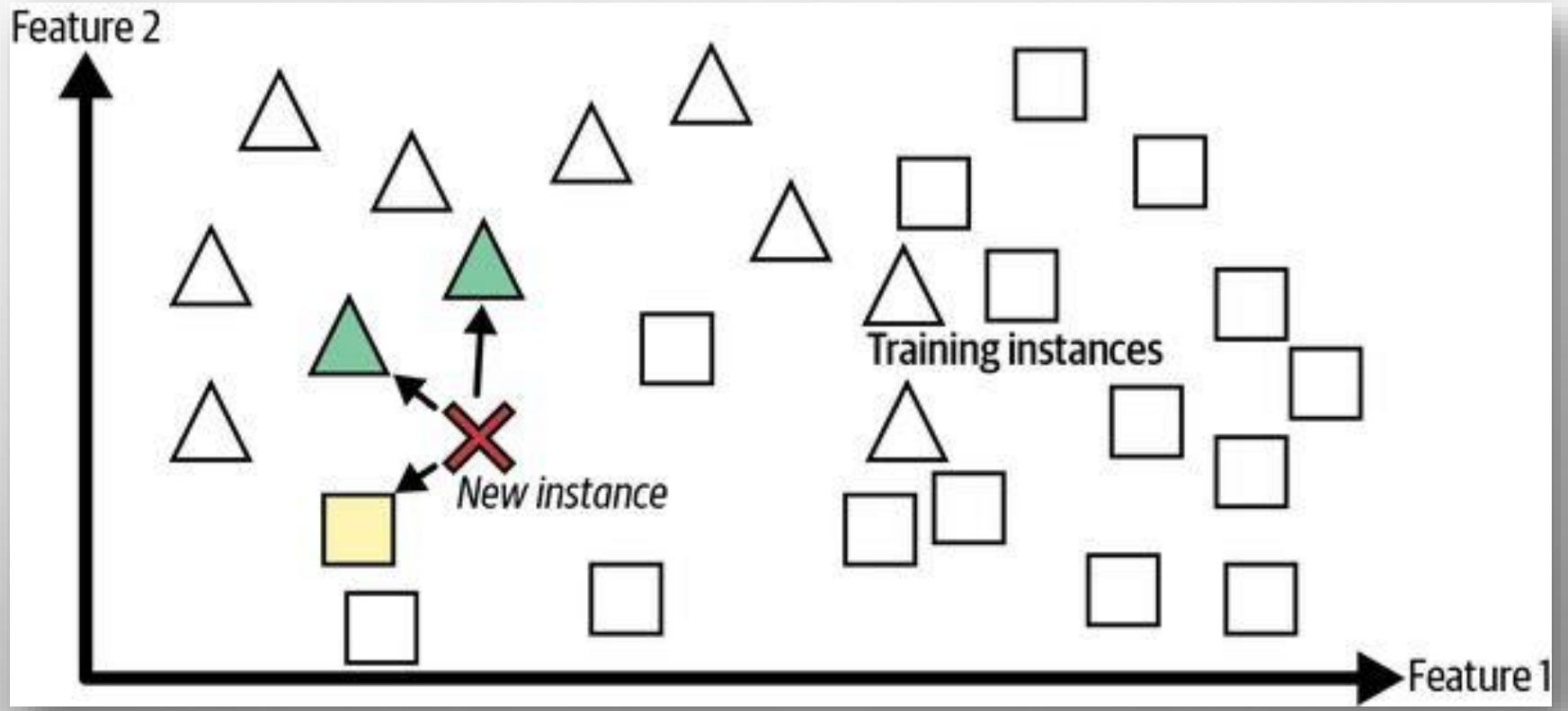
REPRODUTIBILIDADE

numpy.random.seed

```
random.seed(seed=None)
```

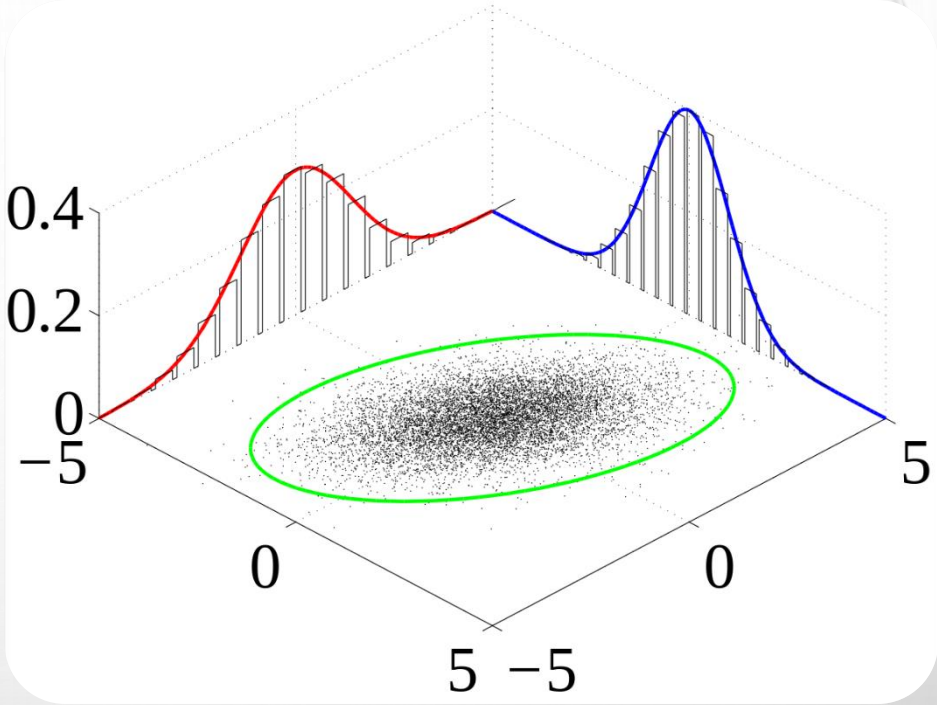
Reseed the singleton RandomState instance.

VIZINHOS MAIS PRÓXIMOS



ALGORITMOS BASEADOS EM DENSIDADE

Algoritmos que dependem da **função densidade de probabilidade** dos dados, ou aproximações locais, para determinar a classe de observações fora da amostra de treino.



1) Classificador Bayesiano
2) Classificador Bayesiano "Naïve"
3) **K-Vizinhos mais próximos**

Algoritmos baseados em densidade dependem da **DENSIDADE (!!)**. Consequentemente, se beneficiam de um **conjunto grande de observações e de baixa esparsidade do espaço de atributos**. O Classificador Bayesiano é considerado o classificador "ótimo", mas é raramente utilizado, dada a dificuldade de estimar a função densidade de probabilidade dos dados. É normalmente utilizado como benchmark para comparação teórica entre os algoritmos de classificação.

VIZINHOS MAIS PRÓXIMOS

KNeighborsClassifier

```
class sklearn.neighbors.KNeighborsClassifier(n_neighbors=5, *, weights='uniform',
algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None,
n_jobs=None)
```

[source]

Classifier implementing the k-nearest neighbors vote.

Read more in the User Guide.

Parameters:

n_neighbors : *int, default=5*

Number of neighbors to use by default for `kneighbors` queries.

weights : *{'uniform', 'distance'}, callable or None, default='uniform'*

Weight function used in prediction. Possible values:

- 'uniform' : uniform weights. All points in each neighborhood are weighted equally.
- 'distance' : weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.
- [callable] : a user-defined function which accepts an array of distances, and returns an array of the same shape containing the weights.

Refer to the example entitled Nearest Neighbors Classification showing the impact of the `weights` parameter on the decision boundary.

EVALUATION

ESTIMANDO O ERRO DE GENERALIZAÇÃO

SINGLE SPLIT (GRUPO DE CONTROLE)

- Amostra é dividida entre treino e teste, mantendo um percentual das observações como grupo de teste externo ao treinamento.

LEAVE ONE OUT

- Uma única observação é deixada de fora a cada treinamento. N treinamentos são realizados para calcular a estatística de erro.

K FOLDS

- Amostra é dividida em K conjuntos. K treinamentos são realizados, mantendo um conjunto como fora-da-amostra.

BOOTSTRAPPING

- O algoritmo itera, amostrando aleatoriamente M observações, para a quantidade Q desejada de treinamentos.

TRAIN TEST SPLIT

train_test_split

sklearn.model_selection.train_test_split(**arrays*, *test_size=None*, *train_size=None*, *random_state=None*, *shuffle=True*, *stratify=None*) [source]

Split arrays or matrices into random train and test subsets.

Quick utility that wraps input validation, `next(ShuffleSplit().split(X, y))`, and application to input data into a single call for splitting (and optionally subsampling) data into a one-liner.

Read more in the User Guide.

Parameters:

***arrays** : *sequence of indexables with same length / shape[0]*

Allowed inputs are lists, numpy arrays, scipy-sparse matrices or pandas dataframes.

test_size : *float or int, default=None*

If float, should be between 0.0 and 1.0 and represent the proportion of the dataset to include in the test split. If int, represents the absolute number of test samples. If None, the value is set to the complement of the train size. If `train_size` is also None, it will be set to 0.25.

train_size : *float or int, default=None*

If float, should be between 0.0 and 1.0 and represent the proportion of the dataset to include in the train split. If int, represents the absolute number of train samples. If None, the value is automatically set to the complement of the test size.

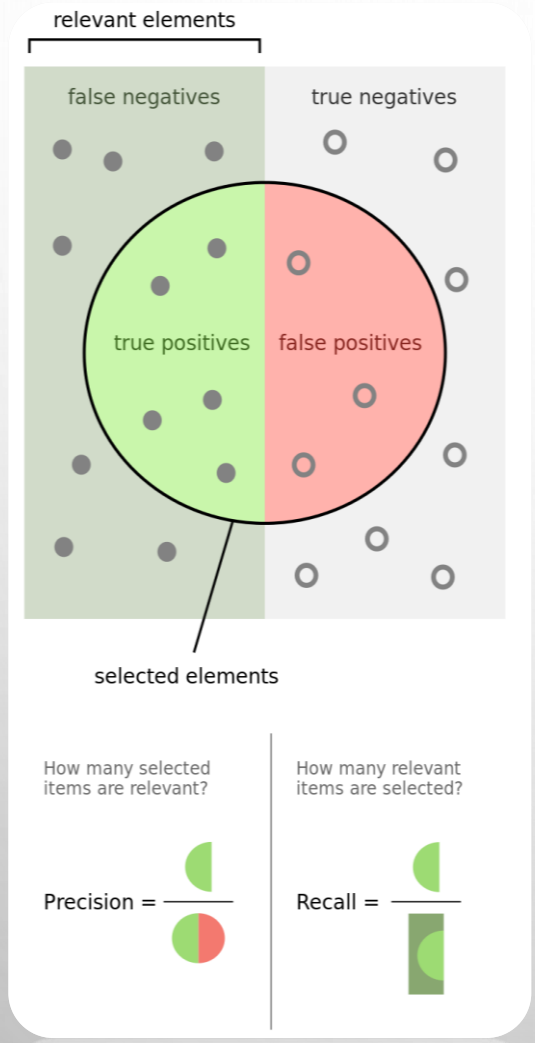
random_state : *int, RandomState instance or None, default=None*

Controls the shuffling applied to the data before applying the split. Pass an int for reproducible output across multiple function calls. See Glossary.

shuffle : *bool, default=True*

Whether or not to shuffle the data before splitting. If shuffle=False then stratify must be None.

FIGURAS DE MÉRITO CLASSIFICAÇÃO



Acurácia

- (TP+TN)/(P+N)

Taxa de Erro

- 1-Acurácia

Sensibilidade (Recall)

- TP/(TP+FN)

Especificidade

- TN/(TN+FP)

Precisão

- TP/(TP+FP)

Produto Sp

- SQRT[SQRT
- (2)/2]

ACURÁCIA

accuracy_score

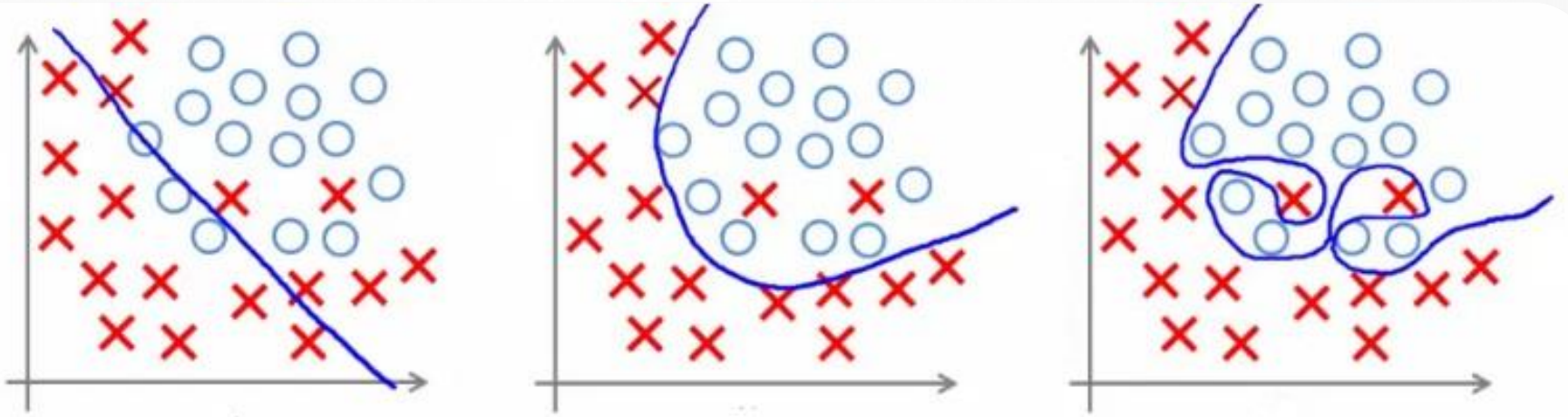
```
sklearn.metrics.accuracy_score(y_true, y_pred, *, normalize=True,
sample_weight=None)
```

[source]

Accuracy classification score.

In multilabel classification, this function computes subset accuracy: the set of labels predicted for a sample must *exactly* match the corresponding set of labels in y_true.

HIPERPARÂMETRO: QUAL O MELHOR VALOR DE K?



CRIANDO MODELOS SIMPLES DE MACHINE LEARNING II