



PYCARET

AutoML com Pycaret

 Stack Academy

Objetivo

- Entender o que é AutoML.
- Desenvolver **experimentos de modelagem** utilizando o Pycaret.
- Ter mais **produtividade e velocidade** no desenvolvimento de projetos de Machine Learning.
- Saber se **comparar modelos, técnicas, algoritmos e pipelines** para implementação de modelos de Machine Learning.
- Estar pronto para **enfrentar qualquer desafio**.



Pra quem é esse curso?

- Cientistas de Dados.
- Engenheiros de Machine Learning.
- Gerentes e Heads de Tecnologia.



Responsabilidades dos Profissionais

- ★ **Cientistas de Dados.**
 - Definir etapas para compor o Pipeline de Machine Learning.
 - Amostragem.
 - Preparação de dados.
 - Missing Values
 - Conversão de tipos de dados.
 - One Hot, Cardinal, Ordinal Encoding.
 - Desbalanceamento de classes.
 - Transformação.
 - Normalização.
 - Escala e Transformação.
 - Engenharia e Seleção de Features.
 - Interação de Features.
 - Agrupamento.
 - Binarização.
 - Importância de Features.
 - Remoção de Multicolinearidade.
 - Análise de componentes principais.
 - Ignorar baixa variância.
 - Definição de tarefas.
 - Otimização e Validação.
 - Definir técnicas e métodos a serem executados.
 - Algoritmos.
 - Estratégia para seleção de hiperparâmetros.
 - Métricas de avaliação.
 - Deploy, Monitoramento e análise.
 - Interpretação de Resultados.
 - Configurar experimentos e integrações.



Responsabilidades dos Profissionais

- Engenheiro de Machine Learning
 - **Todas as atribuições do Cientista de Dados citadas anteriormente e ainda.**
 - Definir ambiente e arquitetura para treinamento e otimização de modelos.
 - On Premise, On Cloud
 - Implementar modelos utilizando Frameworks e Linguagens para realizar previsões em escala.
 - PySpark, Scala, Java, Python, C++
 - Definir formas de processamento.
 - Streaming, Clusters.
 - Monitoramento e avaliação de resultados.



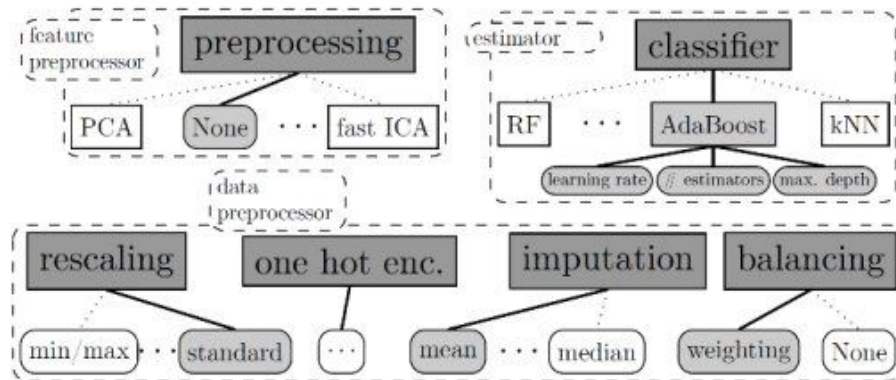
Introdução a AutoML

AutoML - Machine Learning Automatizado.

Fornece métodos, técnicas e processos para fazer Machine Learning disponível a profissionais não experts, melhorando resultados com menos esforço.

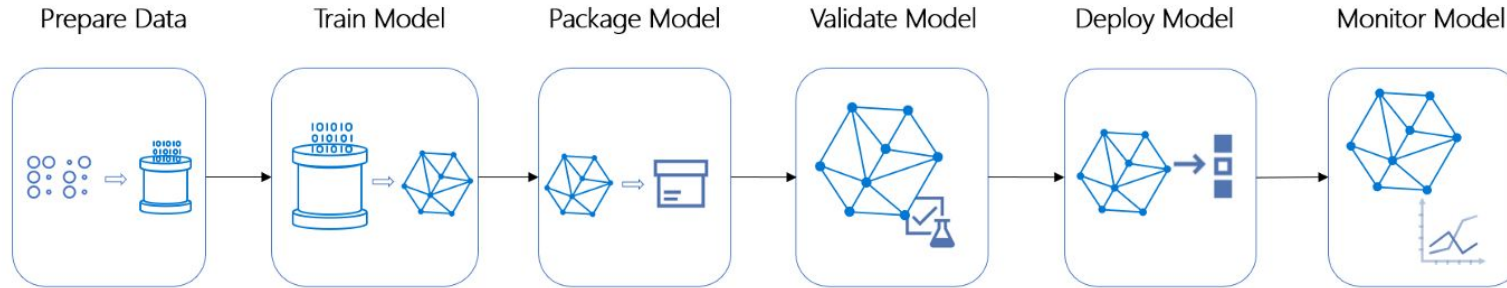
Automação de tarefas como:

- ☐ Pré-processar e limpar dados.
- ☐ Selecionar e construir features apropriadas.
- ☐ Selecionar o melhor modelo.
- ☐ Otimizar hiperparâmetros do modelo.
- ☐ Projetar a topologia de redes neurais.
- ☐ Desenvolver pipelines para predição.
- ☐ Analisar os resultados obtidos.



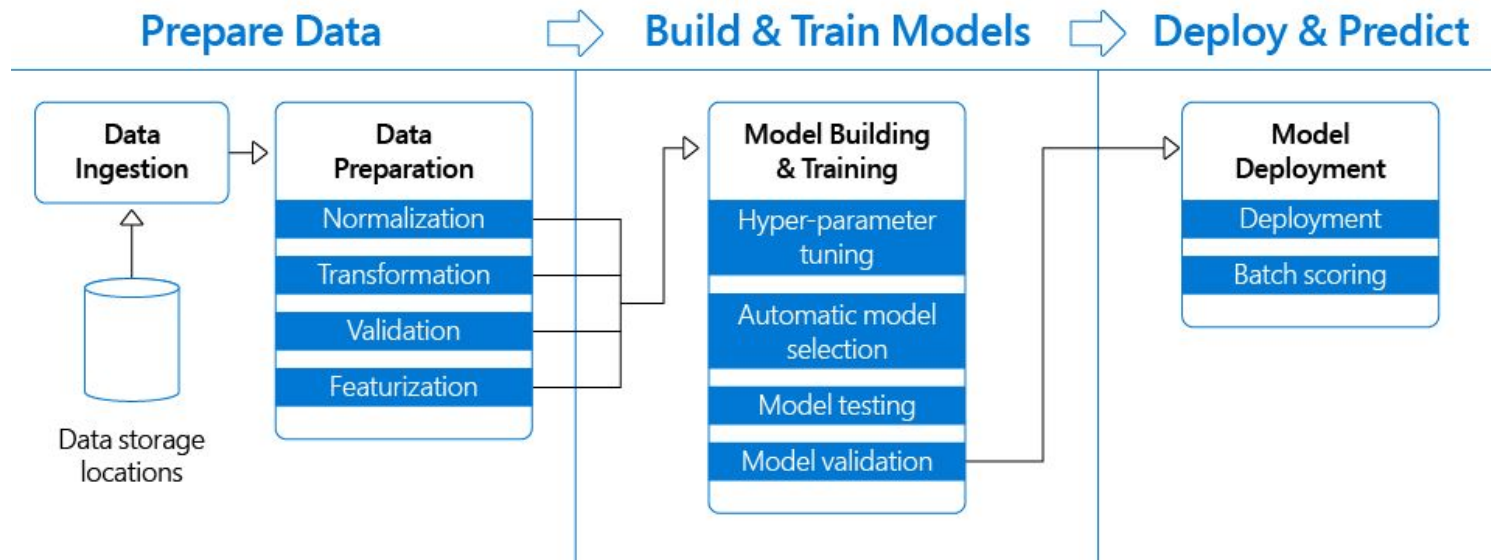
Machine Learning Pipeline

Etapas de um Pipeline para disponibilização de um modelo de Machine Learning.



Machine Learning Pipeline

Pycaret nos permite automatizar todas as etapas de um Pipeline para disponibilização de um modelo de Machine Learning.



Machine Learning Pipeline

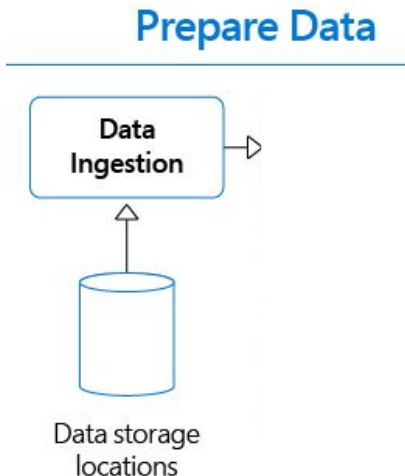
Preparação de Dados e Ingestão

Detecção de tipos de colunas.

- Boolean.
- Numérica discreta/contínua.
- Textual.
- Label/Target

Detecção de tarefas.

- Classificação binária.
- Regressão.
- Clustering.
- Ranking.



Machine Learning Pipeline

Feature Engineer

- Seleção de Features.
- Extração de Features.
- Meta learning e transfer learning
- Detecção e manipulação de dados enviesados.
- Manipulação de valores faltantes.

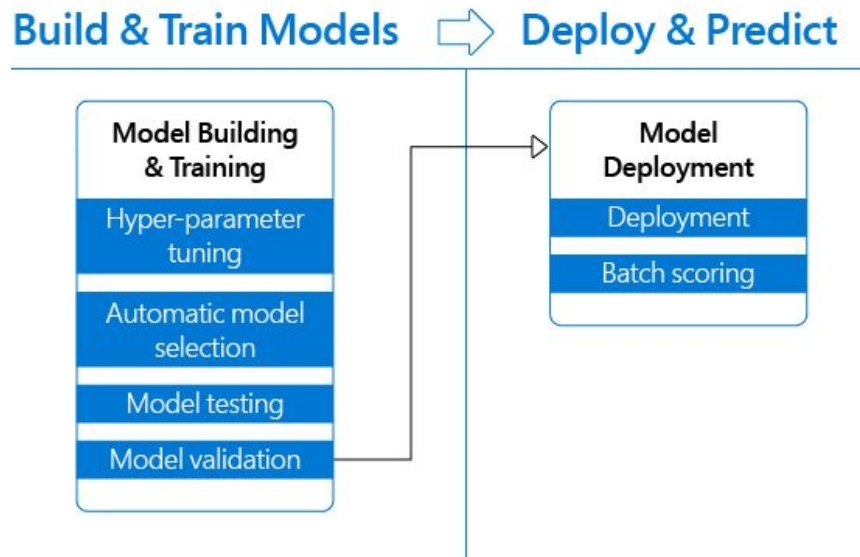
Prepare Data



Machine Learning Pipeline

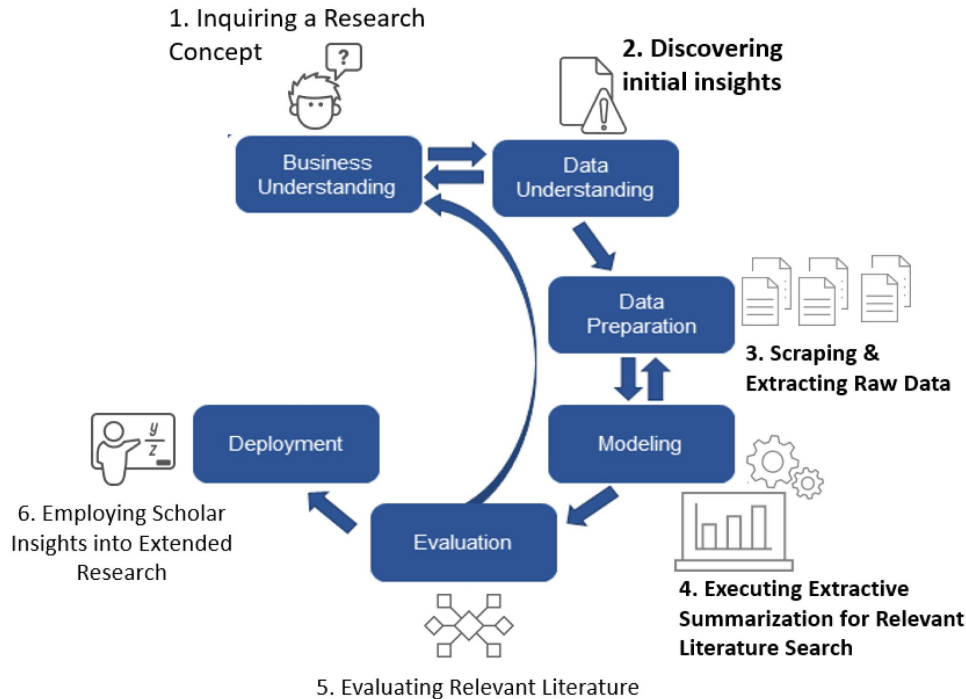
Seleção de Modelos e Otimização de Hiperparâmetros

- Seleção de modelos.
- Otimização de hiperparâmetros.
- Seleção de métricas e validação.
- Deploy e avaliação.



O que NÃO é AutoML

AutoML não é Auto Data Science.



PYCARET



Data
Preparation



Model
Training



Hyperparameter
Tuning



Analysis &
Interpretability



Model
Selection



Experiment
Logging



"PyCaret is an open-source, low-code machine learning library in Python that aims to reduce the cycle time from hypothesis to insights.

It is well suited for seasoned data scientists who want to increase the productivity of their ML experiments by using PyCaret in their workflows or for citizen data scientists and those new to data science with little or no background in coding.

PyCaret allows you to go from preparing your data to deploying your model within seconds using your choice of notebook environment."



Pycaret

Biblioteca Python Open Source e Low Code para trabalhar com Machine Learning.

Pycaret automatiza tarefas para resolução de problemas utilizando Machine Learning.

- Classificação.
- Regressão.
- Clustering.
- Detecção de Anomalias.
- Processamento de Linguagem Natural.
- Mineração de regras de associação.



Data
Preparation



Model
Training



Hyperparameter
Tuning



Analysis &
Interpretability



Model
Selection



Experiment
Logging

Pycaret

Pycaret encapsula funções, técnicas, métodos e algoritmos de outras bibliotecas e frameworks como:

- ☐ CatBoost.
- ☐ LightGBM.
- ☐ XG Boost.
- ☐ Scikit-learn.
- ☐ Spacy.
- ☐ Optuna.
- ☐ Ray.

PYCARET



CatBoost



LightGBM

XGBoost



spaCy



OPTUNA



HYPEROPT



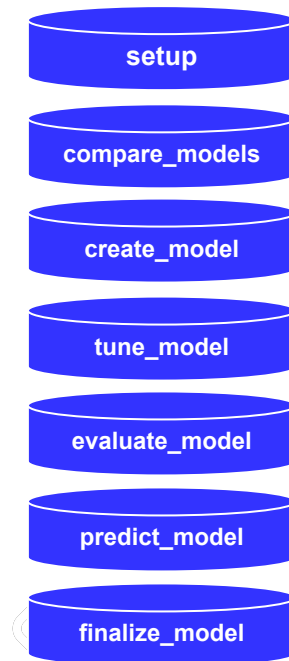
RAY

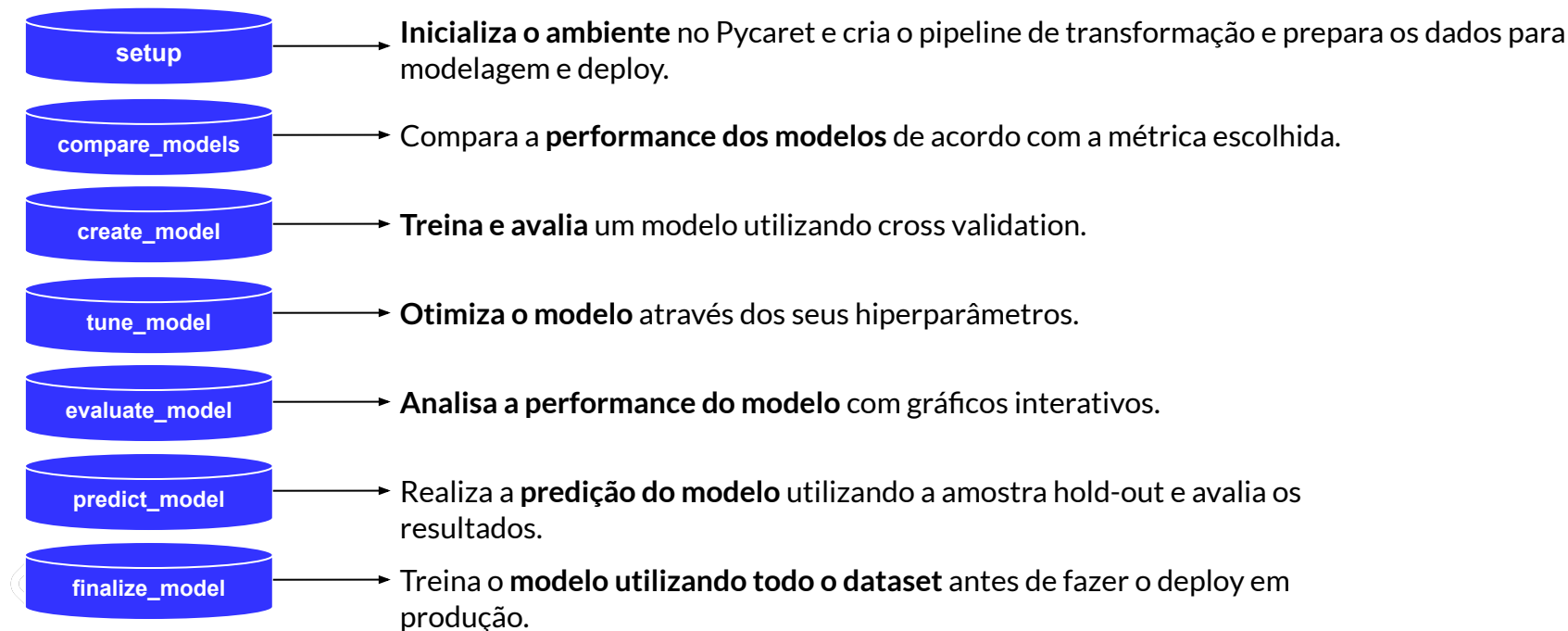


HANDS ON!



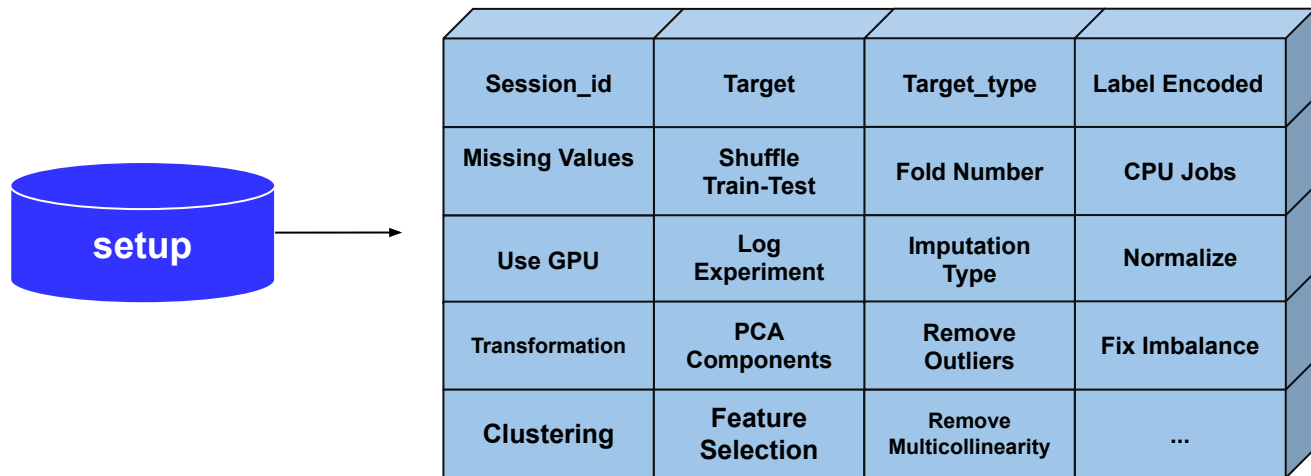
Funções para implementar o Pipeline de Machine Learning do Pycaret





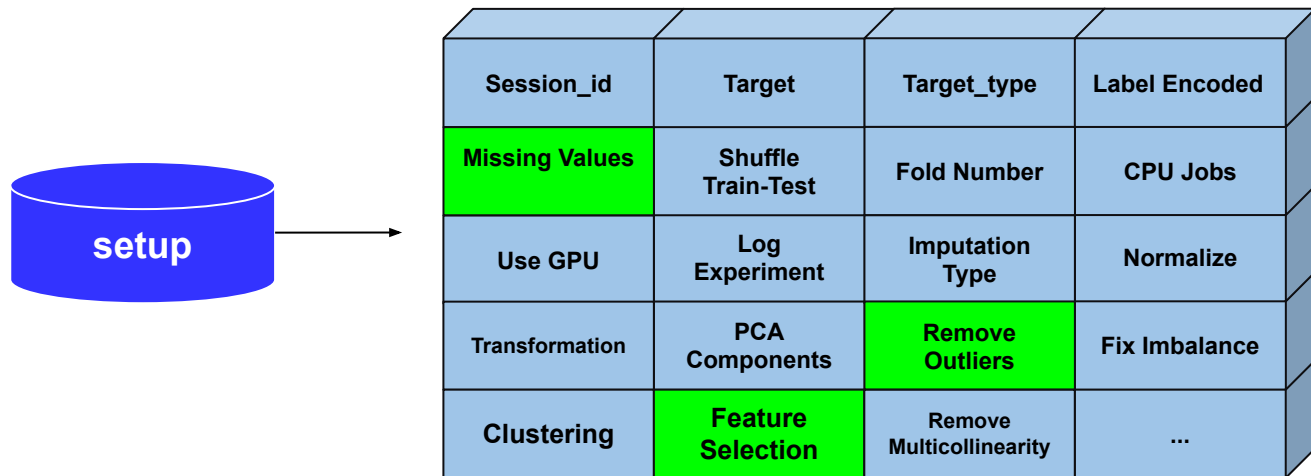


Inicializa o ambiente no Pycaret e cria o pipeline de transformação e prepara os dados para modelagem e deploy.



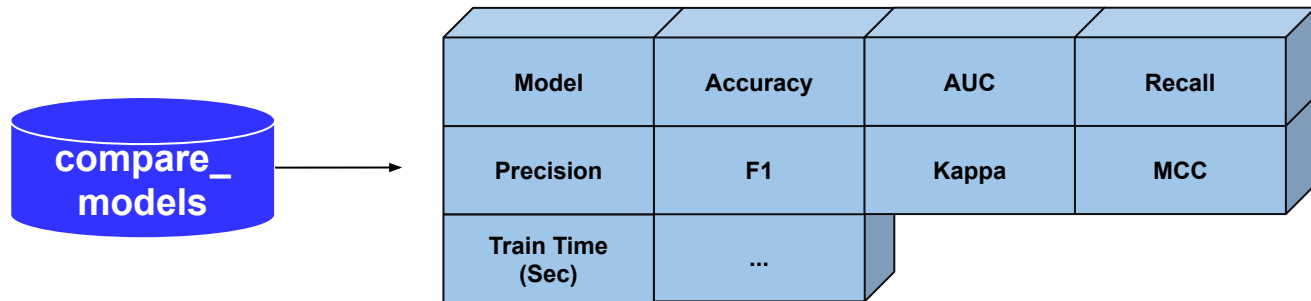
PYCARET

Inicializa o ambiente no Pycaret e cria o pipeline de transformação e prepara os dados para modelagem e deploy.



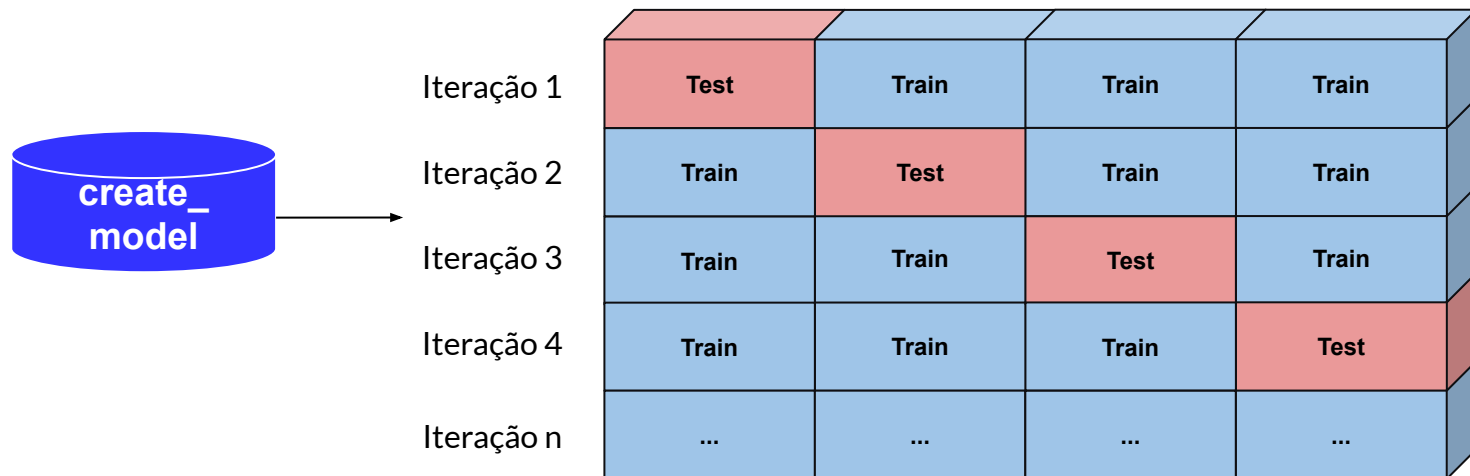


Esta função **treina e compara a performance de todos os modelos** para a determinada tarefa escolhida (Classificação, Regressão, Clustering etc) utilizando cross validation e retorna as métricas para avaliação e o tempo de treinamento.





Esta função **treina o modelo especificado** utilizando cross validation e retorna as métricas para avaliação..



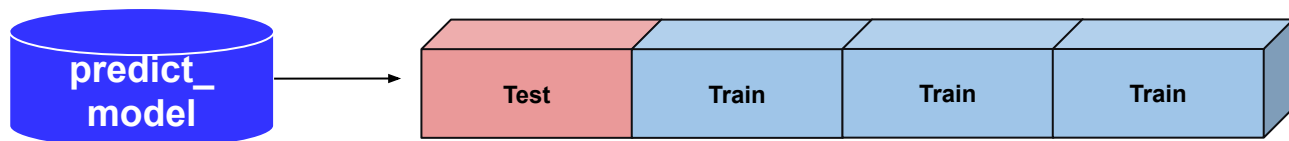
PYCARET

Esta função **otimiza o modelo através dos seus hiperparâmetros** utilizando diferentes técnicas e ferramentas de otimização como scikit-learn, scikit-optimize, hyperopt e optuna.



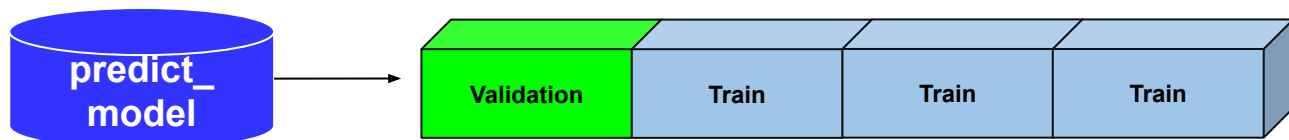


Esta função **realiza a predição** utilizando o modelo especificado utilizando a amostra separada de teste.



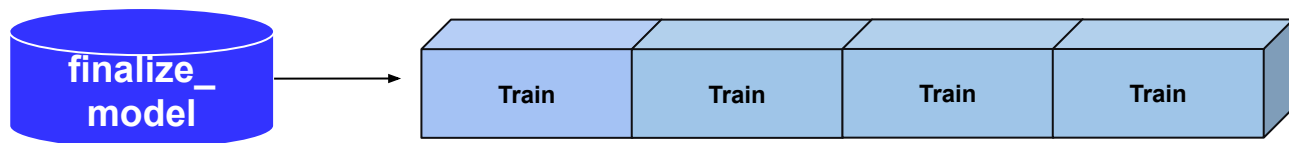


Esta função realiza a predição utilizando o modelo especificado utilizando a amostra separada como um conjunto de validação.





Esta função realiza a predição utilizando o modelo especificado utilizando a amostra separada como um conjunto de validação.



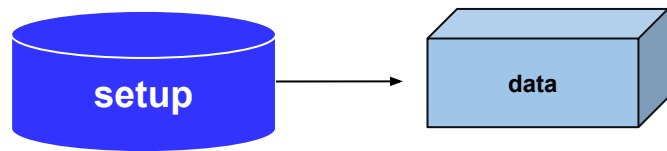


SETUP - Inicializa o ambiente no Pycaret e cria o pipeline de transformação e prepara os dados para modelagem e deploy.



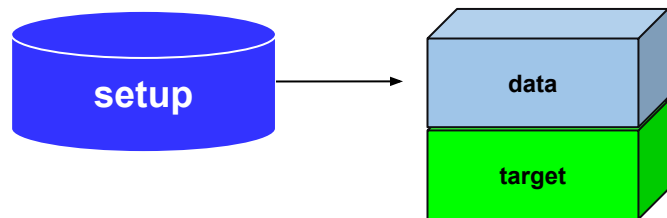
| | | | |
|----------------|--------------------|--------------------------|---------------|
| Session_id | Target | Target_type | Label Encoded |
| Missing Values | Shuffle Train-Test | Fold Number | CPU Jobs |
| Use GPU | Log Experiment | Imputation Type | Normalize |
| Transformation | PCA Components | Remove Outliers | Fix Imbalance |
| Clustering | Feature Selection | Remove Multicollinearity | ... |

PYCARET



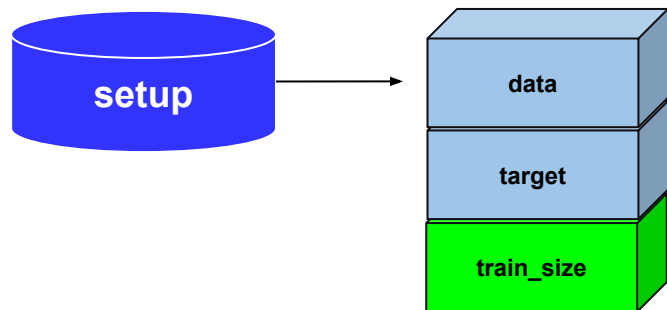
Objeto **pandas DataFrame** com os dados que serão utilizados na modelagem.

PYCARET



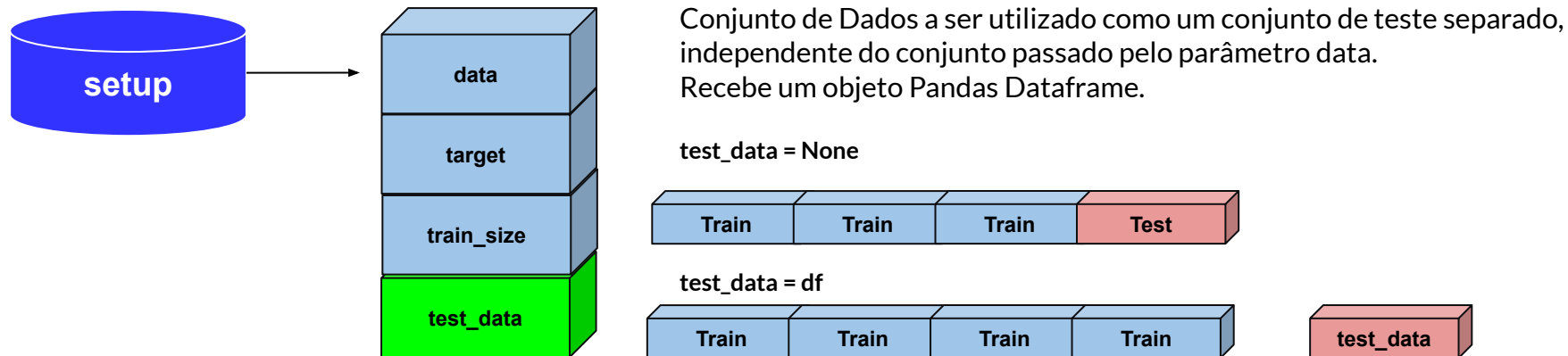
Nome da coluna alvo a ser passada como uma string.

PYCARET

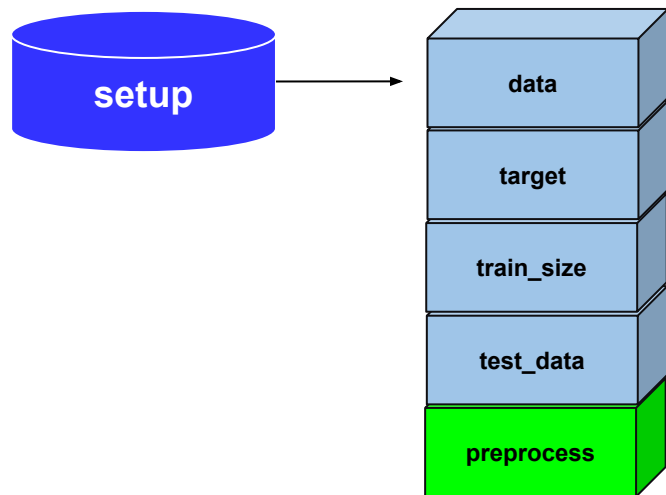


Proporção do conjunto de dados que será utilizado para treino dos algoritmos e criação dos modelos.
O valor deve ser passado entre os intervalos 0.0 a 1.0.
Por padrão: 0.7

PYCARET



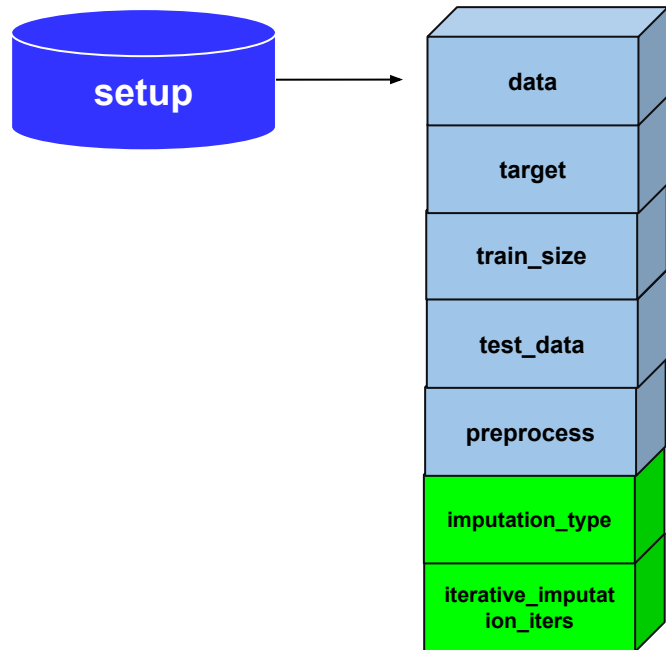
PYCARET



Aplica tarefas de pré processamento aos dados de entrada. Quando setado para False, não aplica as transformações apenas o **train_test_split** e as transformações personalizadas no parâmetro **custom_pipeline**.

Quando definido como False os dados já devem ter passado por transformações como **missing_values**, **encoding** e estarem prontos para modelagem.

PYCARET

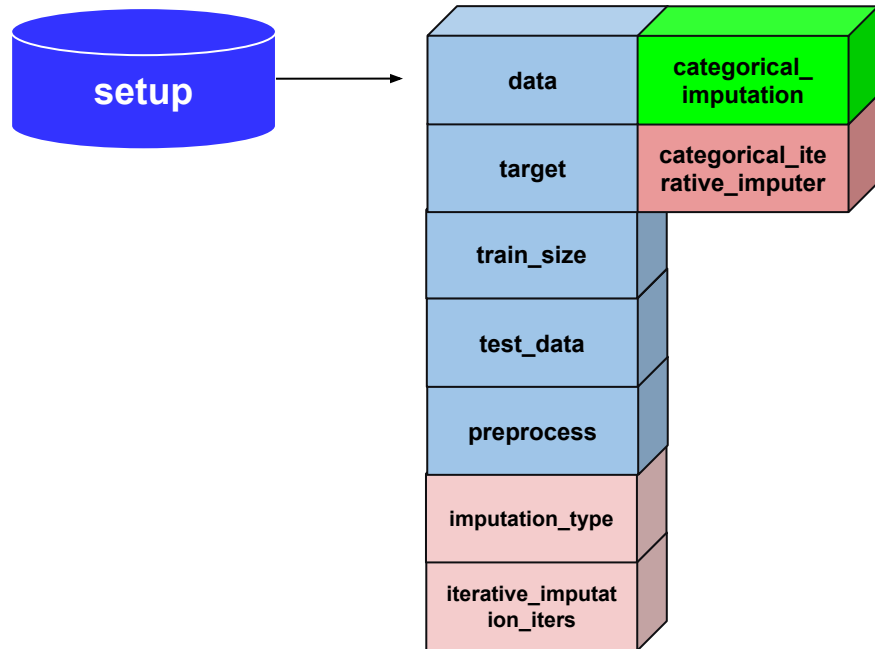


Estabelece os tipos de input e iterações para tratamento de dados missing.

Os valores podem ser:

- simple (default) ou interactive.
- default 5 iterações.

PYCARET



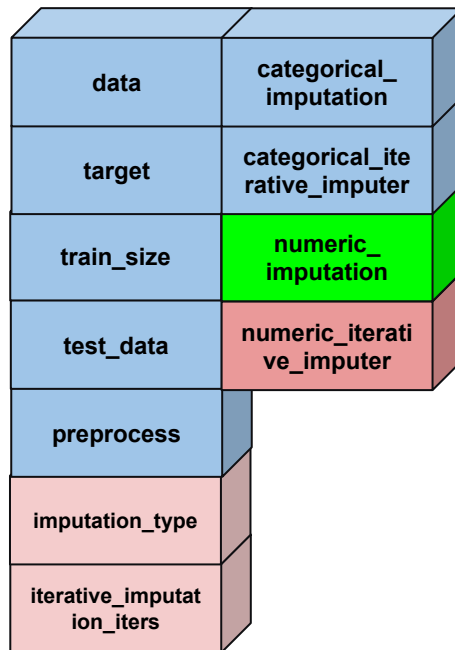
Como será feito o input em features categóricas.

- constant (default) = "not_available"
- mode = valor mais frequente da distribuição.

Estimator utilizado para input iterativo.

- default lightgbm

PYCARET



Como será feito o input em features numéricas.

- mean (default) = valor médio dos dados de treinamento.
- median = valor da mediana dos dados de treinamento.
- zero.

Estimator utilizado para input iterativo.

- default lightgbm

Métodos de input simple - Features categóricas

Tratando missing values em features categóricas com input do tipo **constant** (default).

| Sex | Occupation |
|--------|----------------|
| Male | Analyst |
| Female | Data Scientist |
| NaN | Data Scientist |
| Male | Data Scientist |
| Male | NaN |



| Sex | Occupation |
|---------------|----------------|
| Male | Analyst |
| Female | Data Scientist |
| not_available | Data Scientist |
| Male | Data Scientist |
| Male | not_available |

Métodos de input simple - Features categóricas

Tratando missing values em features categóricas com input do tipo **mode**.

| Sex | Occupation |
|--------|----------------|
| Male | Analyst |
| Female | Data Scientist |
| NaN | Data Scientist |
| Male | Data Scientist |
| Male | NaN |



| Sex | Occupation |
|--------|----------------|
| Male | Analyst |
| Female | Data Scientist |
| Male | Data Scientist |
| Male | Data Scientist |
| Male | Data Scientist |

Métodos de input simple - Features numéricas

Tratando missing values em features numéricas com input como valor da **média** (default).

| Salary | Vacation |
|----------|----------|
| \$ 7.300 | 25.0 |
| \$ 5.900 | 15.0 |
| NaN | 0 |
| \$ 7.300 | 15.0 |
| \$ 6.500 | NaN |



| Salary | Vacation |
|----------|----------|
| \$ 7.300 | 25.0 |
| \$ 5.900 | 15.0 |
| \$ 6.750 | 0 |
| \$ 7.300 | 15.0 |
| \$ 6.500 | 13.7 |

Métodos de input simple - Features numéricas

Tratando missing values em features numéricas com input com o valor da mediana.

| Salary | Vacation |
|----------|----------|
| \$ 7.300 | 25.0 |
| \$ 5.900 | 15.0 |
| NaN | 0 |
| \$ 7.300 | 15.0 |
| \$ 6.500 | NaN |



| Salary | Vacation |
|----------|----------|
| \$ 7.300 | 25.0 |
| \$ 5.900 | 15.0 |
| \$ 6.900 | 0 |
| \$ 7.300 | 15.0 |
| \$ 6.500 | 15.0 |

Métodos de input simple - Features numéricas

Tratando missing values em features numéricas com input de valores zero.

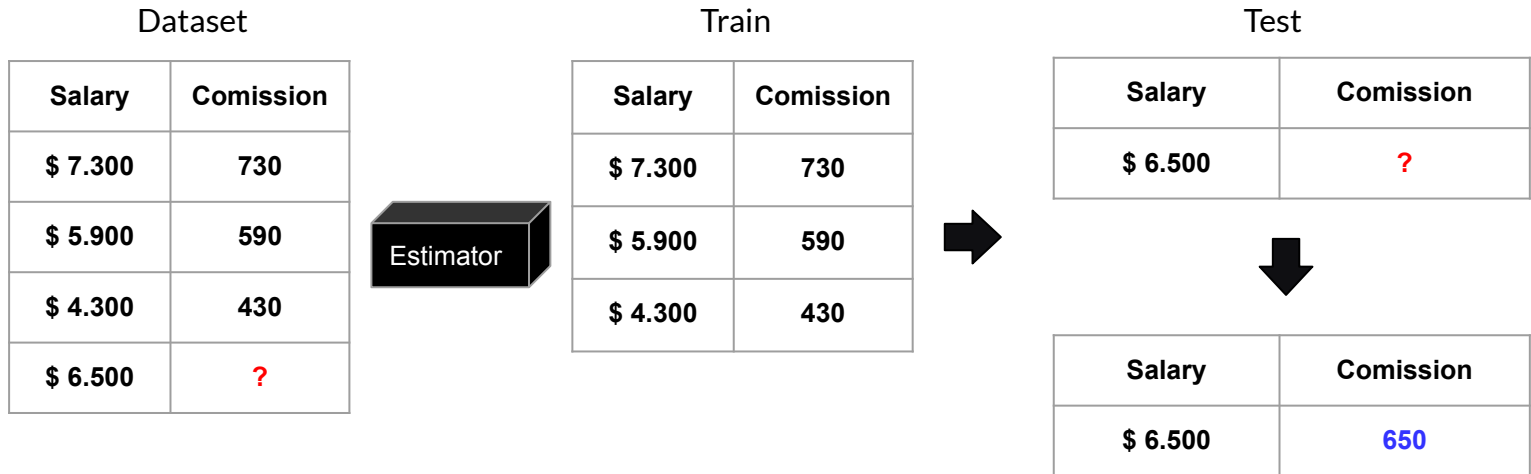
| Salary | Vacation |
|----------|----------|
| \$ 7.300 | 25.0 |
| \$ 5.900 | 15.0 |
| NaN | 20.0 |
| \$ 7.300 | 15.0 |
| \$ 6.500 | NaN |



| Salary | Vacation |
|----------|----------|
| \$ 7.300 | 25.0 |
| \$ 5.900 | 15.0 |
| 0 | 20.0 |
| \$ 7.300 | 15.0 |
| \$ 6.500 | 0 |

Métodos de input iterative - Features numéricas

Tratando missing values em features utilizando o método iterativo.



PYCARET



| | |
|----------------------------|-------------------------------|
| data | categorical_imputation |
| target | categorical_iterative_imputer |
| train_size | numeric_imputation |
| test_data | numeric_iterative_imputer |
| preprocess | categorical_features |
| imputation_type | |
| iterative_imputation_iters | |

- Lista de string usada para sobrescrever a inferência de data types para features categóricas.
- Default = None.

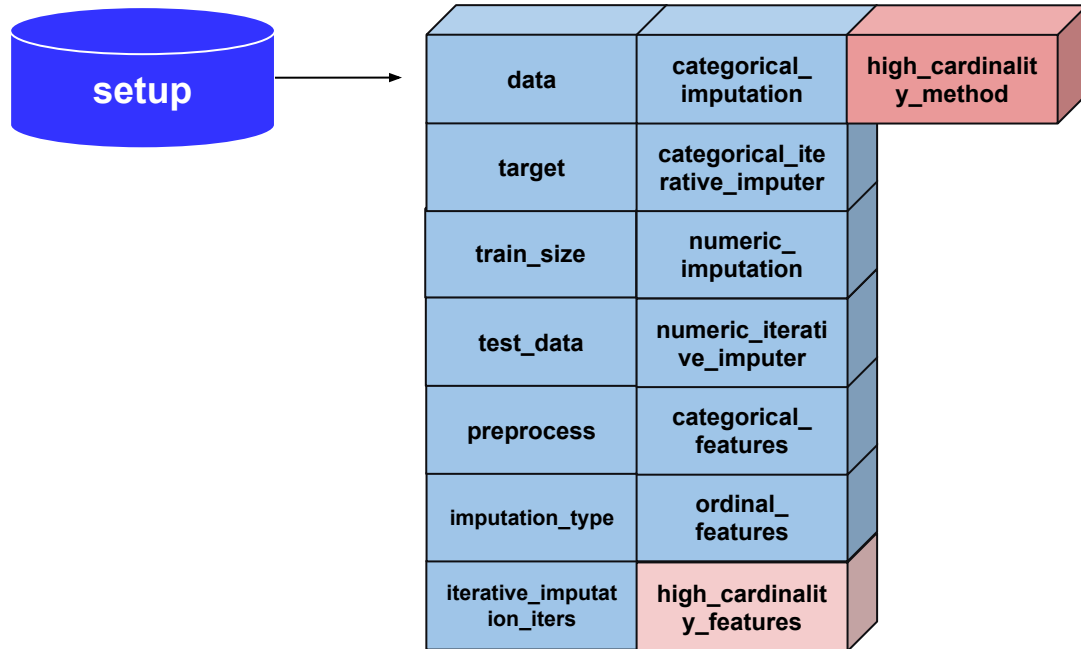
PYCARET



| | |
|----------------------------|-------------------------------|
| data | categorical_imputation |
| target | categorical_iterative_imputer |
| train_size | numeric_imputation |
| test_data | numeric_iterative_imputer |
| preprocess | categorical_features |
| imputation_type | ordinal_features |
| iterative_imputation_iters | |

- Codifica dados de uma feature categórica para seguir uma ordem de grandeza.
- Exemplo: "low", "medium", "high" onde $low < medium < high$.
- Default = None.

PYCARET



- Nos permite fazer um agrupamento de features que contém alta cardinalidade em poucos níveis.
- Método por frequência: Dados são substituídos pela sua frequência.
- Método por clustering: Treina o K-Means sobre os dados de treinamento e substitui os dados pela label do clustering. O número de clusters é otimizado pelos critérios Calinski-Harabasz e Silhouette.



Tratando features com alta cardinalidade.
Cenário com **One Hot Encoding**.

| Age | Capital-Gain | Sex | Native-Country |
|-----|--------------|--------|----------------|
| 38 | 27700 | Male | United-States |
| 29 | 54841 | Female | México |
| 55 | 69880 | NaN | Germany |
| 32 | 35444 | Male | India |
| 18 | 25699 | Male | Cuba |



| Age | Capital-Gain | Sex | Native-Country_ United_States | Native-Country_ _México | Native-Country_ India |
|-----|--------------|--------|----------------------------------|----------------------------|--------------------------|
| 38 | 27700 | Male | 1.0 | 0.0 | 0.0 |
| 29 | 54841 | Female | 0.0 | 1.0 | 0.0 |
| 44 | 54887 | Female | 0.0 | 0.0 | 1.0 |
| ... | ... | ... | ... | ... | ... |



Tratando features com alta cardinalidade utilizando o método de **frequência**.

Dados de treinamento

| Age | Capital-Gain | Sex | Native-Country |
|-----|--------------|--------|----------------|
| 38 | 27700 | Male | United-States |
| 29 | 54841 | Female | México |
| 55 | 69880 | NaN | Germany |
| 32 | 35444 | Male | India |
| 18 | 25699 | Male | United-States |



Feature transformada

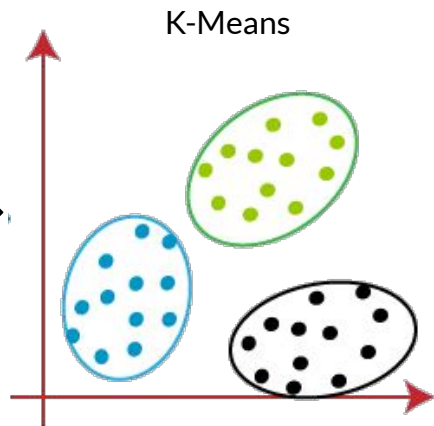
| Age | Capital-Gain | Sex | Native-Country |
|-----|--------------|--------|----------------|
| 38 | 27700 | Male | 2 |
| 29 | 54841 | Female | 1 |
| 55 | 69880 | NaN | 1 |
| 32 | 35444 | Male | 1 |
| 18 | 25699 | Male | 2 |



Tratando features com alta cardinalidade utilizando o método de **clustering**.

Dados de treinamento

| Age | Capital-Gain | Sex |
|-----|--------------|--------|
| 38 | 27700 | Male |
| 29 | 54841 | Female |
| 55 | 69880 | NaN |
| 32 | 35444 | Male |
| 18 | 25699 | Male |



Features criadas

| Native-country_0 | Native-country_1 | Native-country_2 |
|------------------|------------------|------------------|
| 1.0 | 0.0 | 0.0 |
| 1.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 1.0 |
| 0.0 | 1.0 | 0.0 |
| 1.0 | 0.0 | 0.0 |

PYCARET



| | | |
|----------------------------|-------------------------------|-------------------------|
| data | categorical_imputation | high_cardinality_method |
| target | categorical_iterative_imputer | numeric_features |
| train_size | numeric_imputation | |
| test_data | numeric_iterative_imputer | |
| preprocess | categorical_features | |
| imputation_type | ordinal_features | |
| iterative_imputation_iters | high_cardinality_features | |

- Definição da lista de features numéricas. Sobrescreve a inferência automática do Pycaret.

PYCARET



| | | |
|----------------------------|-------------------------------|-------------------------|
| data | categorical_imputation | high_cardinality_method |
| target | categorical_iterative_imputer | numeric_features |
| train_size | numeric_imputation | date_features |
| test_data | numeric_iterative_imputer | |
| preprocess | categorical_features | |
| imputation_type | ordinal_features | |
| iterative_imputation_iters | high_cardinality_features | |

- Definição da lista de features do tipo datetime. Sobrescreve a inferência automática do Pycaret.

PYCARET



| | | |
|----------------------------|-------------------------------|-------------------------|
| data | categorical_imputation | high_cardinality_method |
| target | categorical_iterative_imputer | numeric_features |
| train_size | numeric_imputation | date_features |
| test_data | numeric_iterative_imputer | ignore_features |
| preprocess | categorical_features | |
| imputation_type | ordinal_features | |
| iterative_imputation_iters | high_cardinality_features | |

- Definição da lista de features que deve ser ignorada durante o processo de treinamento.

PYCARET



| | | |
|----------------------------|-------------------------------|-------------------------|
| data | categorical_imputation | high_cardinality_method |
| target | categorical_iterative_imputer | numeric_features |
| train_size | numeric_imputation | date_features |
| test_data | numeric_iterative_imputer | normalize |
| preprocess | categorical_features | normalize_method |
| imputation_type | ordinal_features | |
| iterative_imputation_iters | high_cardinality_features | |

- Quando definida como **True** aplica um método de scaling nas features numéricas.
- **normalize_method** especifica qual o método de scaling a ser utilizado. Default: z-score. Outras opções são:
 - minmax, maxabs, robust.

PYCARET



| | | |
|----------------------------|-------------------------------|-------------------------|
| data | categorical_imputation | high_cardinality_method |
| target | categorical_iterative_imputer | numeric_features |
| train_size | numeric_imputation | date_features |
| test_data | numeric_iterative_imputer | normalize |
| preprocess | categorical_features | normalize_method |
| imputation_type | ordinal_features | transformation |
| iterative_imputation_iters | high_cardinality_features | transformation_method |

- Quando definida como **True** aplica um método de transformação nos dados.
- **transformation_method** especifica qual o método de transformação a ser utilizado. Default: yeo-johnson. Outras opções são:
 - **quantile**.

PYCARET



| | | | |
|----------------------------|-------------------------------|-------------------------|----------------------------|
| data | categorical_imputation | high_cardinality_method | handle_unknown_categorical |
| target | categorical_iterative_imputer | numeric_features | unknown_categorical_method |
| train_size | numeric_imputation | date_features | |
| test_data | numeric_iterative_imputer | normalize | |
| preprocess | categorical_features | normalize_method | |
| imputation_type | ordinal_features | transformation | |
| iterative_imputation_iters | high_cardinality_features | transformation_method | |

- Como manipular dados desconhecidos em features categóricas.
- **unknown_categorical_method** especifica qual o método de substituição dos dados desconhecidos: **least_frequency** (default), **most_frequency** dos dados de treinamento.

PYCARET



| | | | |
|----------------------------|-------------------------------|-------------------------|----------------------------|
| data | categorical_imputation | high_cardinality_method | handle_unknown_categorical |
| target | categorical_iterative_imputer | numeric_features | unknown_categorical_method |
| train_size | numeric_imputation | date_features | pca |
| test_data | numeric_iterative_imputer | normalize | pca_method |
| preprocess | categorical_features | normalize_method | pca_components |
| imputation_type | ordinal_features | transformation | |
| iterative_imputation_iters | high_cardinality_features | transformation_method | |

- Habilita a aplicação do PCA para reduzir a dimensionalidade dos dados.
- **pca_method** especifica qual o método a ser utilizado.
- Opções:
 - linear, kernel e incremental.
- **pca_components** define o número de componentes a serem mantidos.

PYCARET



| | | | |
|----------------------------|-------------------------------|-------------------------|----------------------------|
| data | categorical_imputation | high_cardinality_method | handle_unknown_categorical |
| target | categorical_iterative_imputer | numeric_features | unknown_categorical_method |
| train_size | numeric_imputation | date_features | pca |
| test_data | numeric_iterative_imputer | normalize | pca_method |
| preprocess | categorical_features | normalize_method | pca_components |
| imputation_type | ordinal_features | transformation | ignore_low_variance |
| iterative_imputation_iters | high_cardinality_features | transformation_method | |

- Quando habilitada essa configuração todas as features com variância insignificante são removidas do dataset.
- O cálculo da variância se dar pela taxa do número de registros únicos pela quantidade de registros e a taxa do valor mais comum pela frequência do segundo valor mais comum.

PYCARET

setup

| | | | | |
|----------------------------|-------------------------------|-------------------------|----------------------------|----------------------|
| data | categorical_imputation | high_cardinality_method | handle_unknown_categorical | rare_level_threshold |
| target | categorical_iterative_imputer | numeric_features | unknown_categorical_method | |
| train_size | numeric_imputation | date_features | pca | |
| test_data | numeric_iterative_imputer | normalize | pca_method | |
| preprocess | categorical_features | normalize_method | pca_components | |
| imputation_type | ordinal_features | transformation | ignore_low_variance | |
| iterative_imputation_iters | high_cardinality_features | transformation_method | combine_rare_levels | |

- Quando habilitada essa configuração combina níveis raros em features categóricas. Isto é, valores abaixo do limiar determinado em `rare_level_threshold`. Default 0.1

PYCARET

setup

| | | | | |
|----------------------------|-------------------------------|-------------------------|----------------------------|----------------------|
| data | categorical_imputation | high_cardinality_method | handle_unknown_categorical | rare_level_threshold |
| target | categorical_iterative_imputer | numeric_features | unknown_categorical_method | bin_numeric_features |
| train_size | numeric_imputation | date_features | pca | |
| test_data | numeric_iterative_imputer | normalize | pca_method | |
| preprocess | categorical_features | normalize_method | pca_components | |
| imputation_type | ordinal_features | transformation | ignore_low_variance | |
| iterative_imputation_iters | high_cardinality_features | transformation_method | combine_rare_levels | |

Transforma features contínuas em categóricas através de grupos com bins semelhantes gerados pelo algoritmo K-means.

PYCARET

setup

| | | | | |
|----------------------------|-------------------------------|-------------------------|----------------------------|----------------------|
| data | categorical_imputation | high_cardinality_method | handle_unknown_categorical | rare_level_threshold |
| target | categorical_iterative_imputer | numeric_features | unknown_categorical_method | bin_numeric_features |
| train_size | numeric_imputation | date_features | pca | remove_outliers |
| test_data | numeric_iterative_imputer | normalize | pca_method | outliers_threshold |
| preprocess | categorical_features | normalize_method | pca_components | |
| imputation_type | ordinal_features | transformation | ignore_low_variance | |
| iterative_imputation_iters | high_cardinality_features | transformation_method | combine_rare_levels | |

Aplicar o algoritmo SVD para remover registros outliers. Por padrão são removidos 0.05% do conjunto de dados.

PYCARET

setup

| | | | | |
|----------------------------|-------------------------------|-------------------------|----------------------------|-----------------------------|
| data | categorical_imputation | high_cardinality_method | handle_unknown_categorical | rare_level_threshold |
| target | categorical_iterative_imputer | numeric_features | unknown_categorical_method | bin_numeric_features |
| train_size | numeric_imputation | date_features | pca | remove_outliers |
| test_data | numeric_iterative_imputer | normalize | pca_method | outliers_threshold |
| preprocess | categorical_features | normalize_method | pca_components | remove_multicollinearity |
| imputation_type | ordinal_features | transformation | ignore_low_variance | multicollinearity_threshold |
| iterative_imputation_iters | high_cardinality_features | transformation_method | combine_rare_levels | remove_perfect_collinearity |

Remoção de multicollinearity baseado em um limite estabelecido através do **multicollinearity_threshold**.

Por padrão em features altamente correlacionadas (correlação igual a 1) a feature menos correlacionada com a variável alvo é removida do dataset.

PYCARET

setup

| | | | | | |
|----------------------------|-------------------------------|-------------------------|-----------------------------------|-----------------------------|-----------------------------|
| data | categorical_imputation | high_cardinality_method | handle_unknown_categorical | rare_level_threshold | feature_selection |
| target | categorical_iterative_imputer | numeric_features | handle_unknown_categorical_method | bin_numeric_features | feature_selection_threshold |
| train_size | numeric_imputation | date_features | pca | remove_outliers | feature_selection_method |
| test_data | numeric_iterative_imputer | normalize | pca_method | outliers_threshold | |
| preprocess | categorical_features | normalize_method | pca_components | remove_multicollinearity | |
| imputation_type | ordinal_features | transformation | ignore_low_variance | multicollinearity_threshold | |
| iterative_imputation_iters | high_cardinality_features | transformation_method | combine_rare_levels | remove_perfect_collinearity | |

Configurações para realizar feature selection. Métodos utilizando o algoritmo boruta ou permutação ("classic")

PYCARET

setup

| | | | | | |
|----------------------------|-------------------------------|-------------------------|----------------------------|-----------------------------|-----------------------------|
| data | categorical_imputation | high_cardinality_method | handle_unknown_categorical | rare_level_threshold | feature_selection |
| target | categorical_iterative_imputer | numeric_features | unknown_categorical_method | bin_numeric_features | feature_selection_threshold |
| train_size | numeric_imputation | date_features | pca | remove_outliers | feature_selection_method |
| test_data | numeric_iterative_imputer | normalize | pca_method | outliers_threshold | ... |
| preprocess | categorical_features | normalize_method | pca_components | remove_multicollinearity | use_gpu |
| imputation_type | ordinal_features | transformation | ignore_low_variance | multicollinearity_threshold | log_experiment |
| iterative_imputation_iters | high_cardinality_features | transformation_method | combine_rare_levels | remove_perfect_collinearity | profile |

use_gpu Habilita a execução de algoritmos utilizando gpu.

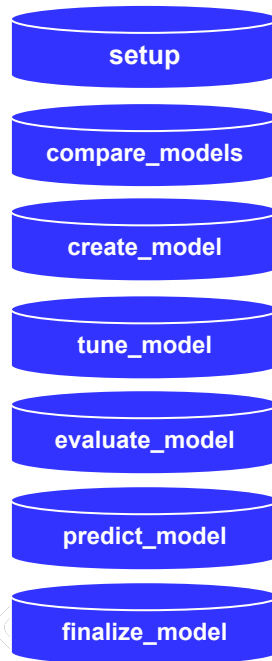
log_experiment: integra o Pycaret com o MLFlow.

profile: Integração do Pycaret com o Pandas-Profiling.



Módulo de Regressão

- Algoritmos
- Métricas
- Parâmetros





Módulo de Clustering

- Algoritmos
- Métricas
- Parâmetros

