

Tutorials

Clustering

[Beginner](#)

Anomaly

[Beginner](#)

Association Rules

[Beginner](#)

NLP

[Beginner](#)

[Intermediate](#)

Regression

[Beginner](#)

[Intermediate](#)

Classification

[Binary \(Beginner\)](#)

[Binary \(Intermediate\)](#)

[Multiclass \(Beginner\)](#)

Time Series

[Beginner](#)

Unsupervised Learning

Supervised Learning

Time Series Analysis

| Clustering | Anomaly Detection | Association Rule | Regression & Classification | | Time Series |
|--|---|---|---|---|--|
| <pre>setup(data=df) create_model(*) assign_model(*) plot_model(**) evaluate_model() tune_model() predict_model() deploy_model() save_model() load_model() pull() models() get_metrics() add_metric() remove_metric() get_logs() get_config() set_config() save_config() load_config() get_clusters()</pre> <pre>* model: 'kmeans' 'ap' 'meanshift' 'sc' 'hclust' 'dbscan' 'optics' 'birch' 'kmodes'</pre> <pre>** plot = 'cluster' 'tsne' 'elbow' 'silhouette' 'distance' 'distribution'</pre> | <pre>setup(data=df) create_model(*) assign_model(*) plot_model(**) evaluate_model() tune_model() predict_model() deploy_model() save_model() load_model() models() get_logs() get_config() set_config() save_config() load_config() get_outliers()</pre> <pre>* model: 'abod' 'cluster' 'histogram' 'knn' 'lof' 'svm' 'pca' 'mcd' 'sod' 'sos'</pre> <pre>** plot= 'tsne' 'umap'</pre> | <pre>setup(data=df, transaction_id='column', item_id='column') create_model() plot_model(**) get_rules()</pre> <pre>** plot= '2d' '3d'</pre> <div>NLP</div> <pre>setup(data=df) create_model(*) assign_model() plot_model() tune_model() evaluate_model() save_model() load_model() models() get_logs() get_config() set_config() set_config() get_topics()</pre> <pre>* model: 'lda' 'lsi' 'hdp' 'rp' 'nmf' 'frequency' 'distribution' 'bigram' 'trigram' 'sentiment' 'pos' 'tsne' 'topic_model' 'topic_distribution' 'wordcloud' 'umap'</pre> | <pre>setup(data=df, target='column') models(*) compare_models() create_model(*) tune_model() ensemble_model() blend_models() stack_models() plot_model(**) evaluate_model() interpret_model() (1) calibrate_model() (1) optimize_threshold() assign_model() calibrate_model() optimize_threshold() get_leaderboard() dashboard() eda() check_fairness() predict_model() finalize_model() convert_model() deploy_model() save_model() save_experiment() save_config() load_config() load_model() convert_model() create_api() create_docker() create_app() automl() pull() models() get_logs() get_config() set_config() get_system_logs()</pre> <pre>(1) classification only</pre> | <pre>* model: (classification) 'lr' 'knn' 'nb' 'dt' 'svm' 'rbfsvm' 'gpc' 'mlp' 'ridge' 'rf' 'qda' 'ada' 'gbc' 'lda' 'et' 'xgboost' 'lightgbm' 'catboost'</pre> <pre>* model: (regression) 'lr' 'lasso' 'ridge' 'en' 'lar' 'llar' 'omp' 'br' 'ard' 'par' 'ransac' 'tr' 'huber' 'kr' 'svm' 'knn' 'dt' 'rf' 'ada' 'gbr' 'mlp' 'xgboost' 'threshold' 'pr' 'error' 'class_report' 'boundary' 'rfe' 'learning' 'manifold' 'calibration' 'vc' 'dimension' 'feature' 'feature_all' 'parameter' 'lift' 'gain' 'tree' 'ks' 'confusion_matrix'</pre> <pre>** plot= 'auc' 'threshold' 'pr' 'error' 'boundary' 'rfe' 'learning' 'manifold' 'calibration' 'vc' 'dimension' 'feature' 'feature_all' 'parameter' 'lift' 'gain' 'tree' 'ks' 'confusion_matrix'</pre> | <pre>setup(data=df) models(*) compare_models() create_model(*) tune_model() blend_models() plot_model() finalize_model() deploy_model() save_model() load_model() pull() models() predict_model() get_metrics() add_metric() remove_metric() get_logs() get_config() set_config() save_config() load_config()</pre> <pre>* model: 'naive' 'grand_means' 'snaive' 'polytrend' 'arima' 'auto_arima' 'exp_smooth' 'ets' 'theta' 'tbats' 'bats' 'prophet' 'lr_cds_dt' 'en_cds_dt' 'ridge_cds_dt' 'lasso_cds_dt' 'lar_cds_dt' 'llar_cds_dt' 'br_cds_dt' 'huber_cds_dt' 'par_cds_dt' 'omp_cds_dt' 'knn_cds_dt' 'dt_cds_dt' 'rf_cds_dt' 'et_cds_dt' 'gbr_cds_dt' 'ada_cds_dt' 'lightgbm_cds_dt'</pre> <pre>** plot= 'ts' 'cv' 'acf' 'pacf' 'decom_stl' 'diagnostics' 'forecast' 'insample' 'residuals' 'train_test_split' 'decom_classical'</pre> |



pycaret.org

Parameters of setup() and its *default values*

Clustering

```
data,
preprocess = True,
imputation_type = 'simple',
iterative_imputation_iters = 5,
categorical_features = None,
categorical_imputation = 'mode',
categorical_iterative_imputer = 'Lightgbm',
ordinal_features = None,
high_cardinality_features = None,
high_cardinality_method = 'frequency',
numeric_features = None,
numeric_imputation = 'mean',
numeric_iterative_imputer = 'Lightgbm',
date_features = None,
ignore_features = None,
normalize = False,
normalize_method = 'zscore',
transformation = False,
transformation_method = 'yeo-johnson',
handle_unknown_categorical = True,
unknown_categorical_method = 'Least_frequent',
pca = False,
pca_method = 'Linear',
pca_components = None,
ignore_low_variance = False,
combine_rare_levels = False,
rare_level_threshold = 0.1,
bin_numeric_features = None,
remove_multicollinearity = False,
multicollinearity_threshold = 0.9,
remove_perfect_collinearity = False,
group_features = None,
group_names = None,
n_jobs = -1,
use_gpu = False,
custom_pipeline = None,
html = True,
session_id = None,
system_log = True,
log_experiment = False,
experiment_name = None,
log_plots = False,
log_profile = False,
log_data = False,
silent = False,
verbose = True,
profile = False,
profile_kwargs = None
```

Color code

required

optional

Anomaly Detection

```
data,
Preprocess = True,
imputation_type = 'simple',
iterative_imputation_iters = 5,
categorical_features = None,
categorical_imputation = 'mode',
categorical_iterative_imputer = 'Lightgbm',
ordinal_features = None,
high_cardinality_features = None,
high_cardinality_method = 'frequency',
numeric_features = None,
numeric_imputation = 'mean',
numeric_iterative_imputer = 'Lightgbm',
date_features = None,
ignore_features = None,
normalize = False,
normalize_method = 'zscore',
transformation = False,
transformation_method = 'yeo-johnson',
handle_unknown_categorical = True,
unknown_categorical_method = 'Least_frequent',
pca = False,
pca_method = 'Linear',
pca_components = None,
ignore_low_variance = False,
combine_rare_levels = False,
rare_level_threshold = 0.1,
bin_numeric_features = None,
remove_multicollinearity = False,
multicollinearity_threshold = 0.9,
remove_perfect_collinearity = False,
group_features = None,
group_names = None,
n_jobs = -1,
use_gpu = False,
custom_pipeline = None,
html = True,
session_id = None,
system_log = True,
log_experiment = False,
experiment_name = None,
log_plots = False,
log_profile = False,
log_data = False,
silent = False,
verbose = True,
profile = False,
profile_kwargs = None
```

Regression & Classification

```
data = DataFrame, target = 'column_name',
train_size = 0.7,
test_data = None,
preprocess = True,
imputation_type = 'simple',
iterative_imputation_iters = 5,
categorical_features = None,
categorical_imputation = 'constant',
categorical_iterative_imputer = 'Lightgbm',
ordinal_features = None,
high_cardinality_features = None,
high_cardinality_method = 'frequency',
numeric_features = None,
numeric_imputation = 'mean',
numeric_iterative_imputer = 'Lightgbm',
date_features = None,
ignore_features = None,
normalize = False,
normalize_method = 'zscore',
transformation = False,
transformation_method = 'yeo-johnson',
handle_unknown_categorical = True,
unknown_categorical_method = 'Least_frequent',
pca = False,
pca_method = 'Linear',
pca_components = None,
ignore_low_variance = False,
combine_rare_levels = False,
rare_level_threshold = 0.1,
bin_numeric_features = None,
remove_outliers = False,
outliers_threshold = 0.05,
remove_multicollinearity = False,
multicollinearity_threshold = 0.9,
remove_perfect_collinearity = True,

create_clusters = False,
cluster_iter = 20,
polynomial_features = False,
polynomial_degree = 2,
trigonometry_features = False,
polynomial_threshold = 0.1,
group_features = None,
group_names = None,
feature_selection = False,
feature_selection_threshold = 0.8,
feature_selection_method = 'classic',
feature_interaction = False,
feature_ratio = False,
interaction_threshold = 0.01,
transform_target = False,
transform_target_method = 'box-cox',
data_split_shuffle = True,
data_split_stratify = False,
fold_strategy = 'kfold',
fold = 10,
fold_shuffle = False,
fold_groups = None,
n_jobs = -1,
use_gpu = False,
custom_pipeline = None,
html = True,
session_id = None,
log_experiment = False,
experiment_name = None,
log_plots = False,
log_profile = False,
log_data = False,
silent = False,
verbose = True,
profile = False,
profile_kwargs = None
```

Time Series

```
data = [.Series, .DataFrame],
preprocess = True,
imputation_type = 'simple',
fold_strategy = 'expanding',
fold = 3,
fh = 1,
seasonal_period = None,
enforce_pi = False,
n_jobs = -1,
use_gpu = False,
custom_pipeline = None,
html = True,
session_id = None,
system_log = True,
log_experiment = False,
experiment_name = None,
log_plots = False,
log_profile = False,
log_data = False,
verbose = True,
profile = False,
profile_kwargs = None
```

Association Rule

```
data,
transaction_id = 'column_name',
item_id = 'column_name',
ignore_items = None,
session_id = None
```

NLP

```
data,
Target = 'column_name',
custom_stopwords = None,
Html = True,
session_id = None,
log_experiment = False,
experiment_name = None,
log_plots = False,
log_data = False,
Verbose = True
```