PYCARET
pycaret.org

Tutorials

Clustering
Beginner

Anomaly
Beginner

Association Rules
Beginner

NLP
Beginner
Intermediate

Regression
Beginner
Intermediate

Classification
Binary (Beginner)
Binary (Intermediate)
Multiclass (Beginner)

Time Series
Beginner

Supervised Learning

Times Series Analysis

Unsupervised Learning

(1) *classification only*

Regression & Classification

```
setup()
create_model(*)
compare_models()
ensemble_models()
tune_model()
blend_models()
stack_models()
plot_model(**)
evaluate_model()
interpret_model()
1 calibrate_model()
1 optimize_threshold()
predict_model()
finalize_model()
deploy_model()
deep_check()
save_model()
load_model()
automl()
pull()
models()
get_metrics()
add_metric()
remove_metric()
get_logs()
get_config()
set_config()
save_experiment()
load_experiment()
get_leaderboard()
set_current_experiment()
get_current_experiment()
dashboard()
convert_model()
eda()
check_fairness()
create_api()
create_docker()
create_app()
get_allowed_engines()
get_engine()
check_drift()
```

* model:
(classification)
```
'lr'
'knn'
'nb'
'dt'
'svm'
'rbfsvm'
'gpc'
'mlp'
'ridge'
'rf'
'qda'
'ada'
'gbc'
'lda'
'et'
'xgboost'
'lightgbm'
'catboost'
```

** plot=
```
'auc'
'threshold'
'pr'
'error'
'class_report'
'boundary'
'rfe'
'learning'
'manifold'
'calibration'
'vc'
'dimension'
'feature'
'feature_all'
'parameter'
'lift'
'gain'
'tree'
'ks'
'confusion_matrix'
```

* model:
(regression)
```
'lr'
'lasso'
'ridge'
'en'
'lar'
'llar'
'omp'
'br'
'ard'
'par'
'ransac'
'tr'
'huber'
'kr'
'svm'
'knn'
'dt'
'rf'
'et'
'ada'
'gbr'
'mlp'
'xgboost'
'lightgbm'
'catboost'
```

** plot=
```
'residuals'
'error'
'cooks'
'rfe'
'learning'
'vc'
'manifold'
'feature'
'feature_all'
'residuals_
 interactive'
'parameter'
'tree'
```

Time Series

```
setup()
create_model(*)
compare_models()
tune_model()
blend_models()
plot_model()
predict_model()
finalize_model()
deploy_model()
save_model()
load_model()
pull()
models()
get_metrics()
add_metric()
remove_metric()
get_logs()
get_config()
set_config()
save_experiment()
load_experiment()
set_current_experiment()
get_current_experiment()
check_stats()
```

** plot=
```
'ts'
'cv'
'acf'
'pacf'
'decomp_stl'
'diagnostics'
'forecast'
'insample'
'residuals'
'train_test_split'
'decomp_classical'
```

* model:
```
'naïve'           'lasso_cds_dt'
'grand_means'     'lar_cds_dt'
'snaive'          'llar_cds_dt'
'polytrend'       'br_cds_dt'
'arima'           'huber_cds_dt'
'auto_arima'      'par_cds_dt'
'exp_smooth'      'omp_cds_dt'
'ets'             'knn_cds_dt'
'theta'           'dt_cds_dt'
'tbats'           'rf_cds_dt'
'bats'            'et_cds_dt'
'prophet'         'gbr_cds_dt'
'lr_cds_dt'       'ada_cds_dt'
'en_cds_dt'       'lightgbm_cds_dt'
'ridge_cds_dt'
```

Clustering

```
setup()
create_model(*)
assign_model()
plot_model(**)
evaluate_model()
predict_model()
deploy_model()
save_model()
load_model()
pull()
models()
get_metrics()
add_metric()
remove_metric()
get_logs()
get_config()
set_config()
save_experiment()
load_experiment()
set_current_experiment()
get_allowed_engines()
get_engine()
get_current_experiment()
```

* model:
```
'kmeans'
'ap'
'meanshift'
'sc'
'hclust'
'dbscan'
'optics'
'birch'
'kmodes'
```

** plot =
```
'cluster'
'tsne'
'elbow'
'silhouette'
'distance'
'distribution'
```

Anomaly Detection

```
setup()
create_model()
compare_models()
tune_model()
blend_models()
plot_model()
predict_model()
finalize_model()
deploy_model()
save_model()
load_model()
pull()
models()
get_metrics()
add_metrics()
remove_metric()
get_logs()
get_config()
set_config()
save_experiment()
load_experiment()
set_current_experiment()
get_current_experiment()
check_stats()
get_allowed_engines()
get_engine()
```

* model:
```
'abod'
'cluster'
'histogram'
'knn'
'lof'
'svm'
'pca'
'mcd'
'sod'
'sos'
```

** plot=
```
'tsne'
'umap'
```

Parameters of `setup()` and its *default values*

Regression & Classification

```python
pycaret.classification.ClassificationExperiment()
setup(
data = None,
data_func = None,
target = -1,
index = True,
train_size = 0.7,
test_data = None,
ordinal_features = None,
numeric_features = None,
categorical_features = None,
date_features = None,
text_features = None,
ignore_features = None,
keep_features = None,
preprocess = True,
create_date_columns = ['day','month','year'],
imputation_type = 'simple',
numeric_imputation = 'mean',
categorical_imputation = 'mode',
iterative_imputation_iters = 5,
numeric_iterative_imputer = 'lightgbm',
categorical_iterative_imputer = 'lightgbm',
text_features_method = 'tf-idf',
max_encoding_ohe = 25,
encoding_method = None,
rare_to_value = None,
rare_value = 'rare',
polynomial_features = False,
polynomial_degree = 2,
low_variance_threshold = None,
group_features = None,
group_names = None,
drop_groups = False,
remove_multicollinearity = False,
multicollinearity_threshold = 0.9,
bin_numeric_features = None,
remove_outliers = False,
outliers_method = 'iforest',
outliers_threshold = 0.05,
(c) fix_imbalance = False,
(c) fix_imbalance_method = 'SMOTE',
transformation = False,
transformation_method = 'yeo-johnson',
normalize = False,
normalize_method = 'zscore',

pca = False,
pca_method = 'linear',
pca_components = None,
feature_selection = False,
feature_selection_method = 'classic',
feature_selection_estimator = 'lightgbm',
n_features_to_select = 0.2,
(r) transform_target = False,
(r) transform_target_method = 'yeo-johnson',
custom_pipeline = None,
custom_pipeline_position = -1,
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,
html = True,
session_id = None,
system_log = True,
log_experiment = False,
experiment_name = None,
experiment_custom_tags = None,
log_plots = False,
log_profile = False,
log_data = False,
engine = None,
verbose = True,
memory = True,
profile = False,
profile_kwargs = None)
```

(c) Classification only
(r) Regression only

Time Series

```python
pycaret.time_series.TSForecastingExperiment()
setup(
data = None,
data_func = None,
target = None,
index = None,
ignore_features = None,
numeric_imputation_target = None,
numeric_imputation_exogenous = None,
transform_target = None,
transform_exogenous = None,
scale_target = None,
scale_exogenous = None,
fe_target_rr = None,
fe_exogenous = None,
fold_strategy = 'expanding',
fold = 3,
fh = 1,
hyperparameter_split = 'all',
seasonal_period = None,
ignore_seasonality_test = False,
sp_detection = 'auto',
max_sp_to_consider = 60,
remove_harmonics = False,
harmonic_order_method = 'harmonic_max',
num_sps_to_use = 1,
point_alpha = None,
coverage = 0.9,
enforce_exogenous = True,
n_jobs = -1,
use_gpu = False,
custom_pipeline = None,
html = True,
session_id = None,
system_log = True,
log_experiment = False,
experiment_name = None,
experiment_custom_tags = None,
log_plots = False,
log_profile = False,
log_data = False,
engine = None,
verbose = True,
profile = False,
profile_kwargs = None,
fig_kwargs = None)
```

Clustering

```python
pycaret.clustering.ClusteringExperiment()
setup(
data = None,
data_func = None,
index = True,
ordinal_features = None,
numeric_features = None,
categorical_features = None,
date_features = None,
text_features = None,
ignore_features = None,
keep_features = None,
preprocess = True,
create_date_columns = ['day','month','year'],
imputation_type = 'simple',
numeric_imputation = 'mean',
categorical_imputation = 'mode',
text_features_method = 'tf-idf',
max_encoding_ohe = -1,
encoding_method = None,
rare_to_value = None,
rare_value = 'rare',
polynomial_features = False,
polynomial_degree = 2,
low_variance_threshold = None,
remove_multicollinearity = False,
multicollinearity_threshold = 0.9,
bin_numeric_features = None,
remove_outliers = False,
outliers_method = 'iforest',
outliers_threshold = 0.05,
transformation = False,
transformation_method = 'yeo-johnson',
normalize = False,
normalize_method = 'zscore',
pca = False,
pca_method = 'linear',
pca_components = None,
custom_pipeline = None,
custom_pipeline_position = -1,
n_jobs = -1,
use_gpu = False,
html = True,
session_id = None,
system_log = True,
log_experiment = False,
experiment_name = None,
experiment_custom_tags = None,
log_plots = False,
log_profile = False,
log_data = False,
verbose = True,
memory = True,
profile = False,
profile_kwargs = None)
```

Anomaly Detection

```python
pycaret.anomaly.AnomalyExperiment()
setup(
data = None,
data_func = None,
index = True,
ordinal_features = None,
numeric_features = None,
categorical_features = None,
date_features = None,
text_features = None,
ignore_features = None,
keep_features = None,
preprocess = True,
create_date_columns = ['day','month','year'],
imputation_type = 'simple',
numeric_imputation = 'mean',
categorical_imputation = 'mode',
text_features_method = 'tf-idf',
max_encoding_ohe = -1,
encoding_method = None,
rare_to_value = None,
rare_value = 'rare',
polynomial_features = False,
polynomial_degree = 2,
low_variance_threshold = None,
group_features = None,
group_names = None,
drop_groups = False,
remove_multicollinearity = False,
multicollinearity_threshold = 0.9,
bin_numeric_features = None,
remove_outliers = False,
outliers_method = 'iforest',
outliers_threshold = 0.05,
transformation = False,
transformation_method = 'yeo-johnson',
normalize = False,
normalize_method = 'zscore',
pca = False, pca_method = 'linear',
pca_components = None,
custom_pipeline = None,
custom_pipeline_position = -1,
n_jobs = -1,
use_gpu = False,
html = True,
session_id = None,
system_log = True,
log_experiment = False,
experiment_name = None,
experiment_custom_tags = None,
log_plots = False,
log_profile = False,
log_data = False,
verbose = True,
memory = True,
profile = False,
profile_kwargs = None)
```