

## pycaret.org

## **Tutorials**

Clustering Beginner

Anomaly

Beginner

**Association Rules** 

<u>Beginner</u>

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 Time Series** Clustering **Anomaly Detection** setup() setup() setup() \* model: setup() \* model: \*\* plot= create model(\*) create model(\*) create model() (classification) (regression) create model(\*) 'ts' compare models() 'lr' 'lr' compare models() compare models() assign model() cv, tune model() tune model() ensemble models() 'knn' 'lasso' plot model(\*\*) 'acf' blend models() 'nb' 'ridge' blend models() tune model() evaluate model() 'pacf' plot model() 'dt' plot model() 'en' blend models() predict model() 'decomp stl' predict model() 'svm' 'lar' predict model() stack models() deploy model() 'diagnostics' finalize model() finalize model() 'rbfsvm' 'llar' plot model(\*\*) save model() 'forecast' deploy model() 'gpc' deploy model() 'omp' load model() evaluate model() 'insample' save model() 'mlp' save model() hr' pull() interpret model() 'residuals' load model() load model() 'ridge' 'ard' models() 1 calibrate model() 'train test split' pull() rf' pull() 'par' 1 optimize threshold() get metrics() 'decomp classical' models() 'qda' models() 'ransac' add metric() predict model() get metrics() 'ada' 'tr' get metrics() finalize model() remove metric() add metric() 'gbc' add metrics() 'huber' get logs() deploy model() remove metric() 'lda' 'kr' remove metric() get config() deep check() get\_logs() 'et' 'svm' get logs() set config() save model() get config() 'xgboost' 'knn' get config() save experiment() load model() set\_config() 'lightgbm' 'dt' set config() load experiment() automl() save experiment() 'catboost' save experiment() rf' set current experiment() pull() load experiment() load experiment() 'et' get allowed engines() models() set current experiment() set current experiment() 'ada' get engine() get metrics() get current experiment() 'gbr' get current experiment() get current experiment() add metric() check\_stats() \*\* plot= 'mlp' check stats() remove metric() get allowed engines() 'auc' 'xgboost' get logs() 'threshold' 'lightgbm' get engine() \* model: get config() \* model: 'pr' 'catboost' 'kmeans' 'lasso\_cds\_dt' 'naïve' set config() 'error' \* model: 'ap' 'lar cds dt' 'grand means' save experiment() 'class report' 'abod' \*\* plot= 'llar cds dt' 'meanshift' 'snaive' load experiment() 'boundary' 'cluster' 'residuals' 'br cds dt' 'sc' 'polytrend' get leaderboard() 'rfe' 'histogram' 'error' 'huber cds dt' 'hclust' set current\_experiment() 'learning' 'arima' 'par\_cds\_dt' 'knn' 'cooks' 'dbscan' 'auto arima' 'omp\_cds\_dt' get current experiment() 'lof' 'manifold' 'rfe' 'optics' 'exp smooth' 'knn cds dt' dashboard() 'svm' 'calibration' 'learning' 'birch' 'dt cds dt' 'ets' convert model() 'pca' 'vc' ر ۷۲ ، 'kmodes' 'rf cds dt' 'theta' 'mcd' eda() 'dimension' 'manifold' 'et\_cds\_dt' 'tbats' check fairness() 'sod' 'feature' 'feature' \*\* plot = 'gbr\_cds\_dt' 'bats' 'feature all' 'feature all' 'sos' create api() 'ada\_cds\_dt' 'cluster' 'prophet' create docker() 'parameter' 'lightgbm\_cds\_dt' 'residuals 'tsne' 'lr cds dt' 'lift' \*\* plot= interactive' create app() 'elbow' 'en cds dt' 'parameter' 'tsne' 'gain' get allowed engines() 'silhouette' 'ridge cds dt' 'tree' 'tree' 'umap' get engine() 'distance' 'ks' check drift() 'distribution' 'confusion matrix'



data = None,

target = -1.

index = True,

data func = None.

train size = 0.7,

test data = None,

ordinal features = None.

numeric features = None.

date\_features = None,

text features = None,

keep features = None,

max encoding ohe = 25,

encoding method = None,

rare to value = None,

rare value = 'rare',

normalize = False.

preprocess = True,

ignore features = None.

categorical features = None,

imputation\_type = 'simple',

numeric imputation = 'mean',

categorical imputation = 'mode',

pvcaret.classification.ClassificationExperiment()

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

### **Regression & Classification**

#### **Time Series** pca = False,data = None, pca method = 'linear'. data func = None. pca components = None, target = None. feature selection = False, index = None, feature selection method = 'classic', ignore features = None, feature\_selection\_estimator = 'lightgbm', n features to select = 0.2. (r) transform target = False. transform target = None, (r) transform target method = 'yeo-johnson', custom pipeline = None, scale target = None, custom pipeline position = -1,

- data split shuffle = True, data split stratify = False. fold strategy = 'kfold', fold = 10, create date columns = ['day', 'month', 'year'], fold shuffle = False, fold groups = None. n jobs = -1, use gpu = False,
- iterative imputation iters = 5, html = True,numeric iterative imputer = 'lightqbm', session id = None, categorical\_iterative\_imputer = 'lightgbm', system log = True, text features method = 'tf-idf', log experiment = False, experiment name = None, experiment custom tags = None,

log plots = False,

memory = True,

profile = False.

profile kwargs = None)

- log profile = False, polynomial features = False, log data = False. polynomial degree = 2. engine = None, low variance threshold = None, verbose = True, 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 = 'veo-iohnson'.

normalize method = 'zscore',

- - (c) Classification only (r) Regression only

Clustering

```
pycaret.time series.TSForecastingExperiment()
numeric imputation target = None,
numeric imputation exogenous = None,
transform exogenous = 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)
```

```
pvcaret.clustering.ClusteringExperiment()
                                               pycaret.anomaly.AnomalyExperiment()
data = None,
                                               data = None.
data func = None.
                                               data func = None,
index = True,
                                               index = True.
                                               ordinal features = None,
ordinal features = None,
numeric features = None,
                                               numeric features = None.
categorical features = None,
                                               categorical features = None,
date features = None.
                                               date features = None,
text features = None,
                                               text features = None,
                                               ignore features = None,
ignore features = None,
keep features = None,
                                               keep features = None,
preprocess = True,
                                               preprocess = True,
create date columns = ['day', 'month', 'year'],
                                              create date columns = ['day', 'month', 'year'],
                                               imputation type = 'simple',
imputation type = 'simple',
numeric imputation = 'mean',
                                               numeric imputation = 'mean'.
                                               categorical imputation = 'mode',
categorical imputation = 'mode',
                                               text features method = 'tf-idf',
text_features_method = 'tf-idf',
                                               max encoding ohe = -1,
max encoding ohe = -1,
encoding method = None,
                                               encoding method = None,
                                               rare to value = None,
rare to value = None,
rare_value = 'rare',
                                               rare value = 'rare'.
                                               polynomial features = False,
polynomial features = False,
polynomial degree = 2,
                                               polynomial degree = 2,
low variance threshold = None,
                                               low variance threshold = None,
                                               group features = None.
remove multicollinearity = False,
multicollinearity threshold = 0.9,
                                               group names = None,
bin numeric features = None,
                                               drop groups = False,
remove outliers = False,
                                               remove multicollinearity = False,
outliers method = 'iforest',
                                               multicollinearity threshold = 0.9,
                                               bin numeric features = None.
outliers threshold = 0.05,
                                               remove outliers = False,
transformation = False,
transformation method = 'yeo-johnson',
                                               outliers method = 'iforest',
normalize = False.
                                               outliers threshold = 0.05,
normalize method = 'zscore'.
                                               transformation = False,
                                               transformation method = 'yeo-johnson',
pca = False,
pca_method = 'linear',
                                               normalize = False.
                                               normalize method = 'zscore',
pca components = None,
custom pipeline = None,
                                               pca = False, pca method = 'linear',
custom pipeline position = -1,
                                               pca components = None,
n jobs = -1,
                                               custom pipeline = None.
                                               custom pipeline position = -1,
use gpu = False,
html = True,
                                               n jobs = -1,
session id = None,
                                               use gpu = False,
system log = True,
                                               html = True,
                                               session id = None.
log experiment = False,
                                               system log = True,
experiment name = None,
experiment_custom_tags = None,
                                               log experiment = False,
log plots = False,
                                               experiment name = None,
log profile = False,
                                               experiment custom tags = None,
                                               log plots = False,
log data = False,
                                               log profile = False.
verbose = True,
                                               log data = False,
memory = True.
profile = False,
                                               verbose = True,
profile kwargs = None)
                                               memory = True,
                                               profile = False.
                                               profile kwargs = None)
```

Anomaly Detection