

## ▼ Classification using PyCaret

- It is used for Binary or Multi-class classification problems
- It has over 18 algorithms and 14 plots to analyze the performance of models.
- It has hyper-parameter tuning, ensembling or advanced techniques like stacking

## ▼ Steps:

1. **Data Acquisition:** import data from PyCaret or any other repository
2. **Setting up Environment:** How to setup an experiment in PyCaret and get started with building classification models
3. **Model Creation:** How to create a model, perform stratified cross validation and evaluate classification metrics
4. **Model Tuning:** How to automatically tune the hyper-parameters of a classification model
5. **Performance Analysis:** How to analyze model performance using various plots
6. **Finalize Model:** How to finalize the best model at the end of the experiment
7. **Predict Model:** How to make predictions on new / unseen data
8. **Save / Load Model:** How to save / load a model for future use

## ▼ Install PyCaret

```
!pip install pycaret
```

```
Building wheel for alembic (setup.py) ... done
Created wheel for alembic: filename=alembic-1.4.1-py2.py3-none-any.whl size=1581
Stored in directory: /root/.cache/pip/wheels/be/5d/0a/9e13f53f4f5dfb67cd8d245bb7
Building wheel for databricks-cli (setup.py) ... done
Created wheel for databricks-cli: filename=databricks_cli-0.16.2-py3-none-any.whl
Stored in directory: /root/.cache/pip/wheels/f4/5c/ed/e1ce20a53095f63b27b4964abb
Building wheel for pyLDAvis (setup.py) ... done
Created wheel for pyLDAvis: filename=pyLDAvis-3.2.2-py2.py3-none-any.whl size=13
Stored in directory: /root/.cache/pip/wheels/f8/b1/9b/560ac1931796b7303f7b517b94
Building wheel for pyod (setup.py) ... done
Created wheel for pyod: filename=pyod-0.9.5-py3-none-any.whl size=132699 sha256=
Stored in directory: /root/.cache/pip/wheels/3d/bb/b7/62b60fb451b33b0df1ab800669
Building wheel for umap-learn (setup.py) ... done
Created wheel for umap-learn: filename=umap_learn-0.5.2-py3-none-any.whl size=82
Stored in directory: /root/.cache/pip/wheels/84/1b/c6/aaf68a748122632967cef4dffe
Building wheel for pynndescent (setup.py) ... done
Created wheel for pynndescent: filename=pynndescent-0.5.5-py3-none-any.whl size=
Stored in directory: /root/.cache/pip/wheels/af/e9/33/04db1436df0757c42fda8ea679
Successfully built htmlmin imagehash alembic databricks-cli pyLDAvis pyod umap-lea
Installing collected packages: threadpoolctl, tangled-up-in-unicode, smmap, scipy,
Attempting uninstall: scipy
Found existing installation: scipy 1.4.1
Uninstalling scipy-1.4.1:
```

```

Uninstalling scipy 1.4.1.
Successfully uninstalled scipy-1.4.1
Attempting uninstall: scikit-learn
Found existing installation: scikit-learn 0.22.2.post1
Uninstalling scikit-learn-0.22.2.post1:
Successfully uninstalled scikit-learn-0.22.2.post1
Attempting uninstall: requests
Found existing installation: requests 2.23.0
Uninstalling requests-2.23.0:
Successfully uninstalled requests-2.23.0
Attempting uninstall: PyYAML
Found existing installation: PyYAML 3.13
Uninstalling PyYAML-3.13:
Successfully uninstalled PyYAML-3.13
Attempting uninstall: yellowbrick
Found existing installation: yellowbrick 0.9.1
Uninstalling yellowbrick-0.9.1:
Successfully uninstalled yellowbrick-0.9.1

Attempting uninstall: pandas-profiling
Found existing installation: pandas-profiling 1.4.1
Uninstalling pandas-profiling-1.4.1:
Successfully uninstalled pandas-profiling-1.4.1
Attempting uninstall: mlxtend
Found existing installation: mlxtend 0.14.0
Uninstalling mlxtend-0.14.0:
Successfully uninstalled mlxtend-0.14.0
Attempting uninstall: lightgbm
Found existing installation: lightgbm 2.2.3
Uninstalling lightgbm-2.2.3:
Successfully uninstalled lightgbm-2.2.3
Attempting uninstall: imbalanced-learn
Found existing installation: imbalanced-learn 0.4.3
Uninstalling imbalanced-learn-0.4.3:
Successfully uninstalled imbalanced-learn-0.4.3
ERROR: pip's dependency resolver does not currently take into account all the packages that are required by packages of the project
google-colab 1.0.0 requires requests~=2.23.0, but you have requests 2.26.0 which is incompatible
datascience 0.10.6 requires folium==0.2.1, but you have folium 0.8.3 which is incompatible

```

## ▼ Dataset source

<https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records>

## ▼ Upload dataset from user-system

```

from google.colab import files
files.upload()

```

Choose files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving heart\_failure\_dataset.csv to heart\_failure\_dataset.csv

```
{'heart_failure_dataset.csv': b'age,anaemia,creatinine_phosphokinase,diabetes,ejec
```

## ▼ Read dataset

```
import pandas as pd
dataset = pd.read_csv('heart_failure_dataset.csv')
```

```
dataset.head()
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_bloo
0	75.0	0	582	0	20	
1	55.0	0	7861	0	38	
2	65.0	0	146	0	20	
3	50.0	1	111	0	20	
4	65.0	1	160	1	20	

## ▼ Upload dataset to google drive, mount google drive and read dataset

```
from google.colab import drive
drive.mount('/content/drive')
Dataset = pd.read_csv('/content/heart_failure_dataset.csv')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m



## ▼ Load Datasets from PyCaret

1. from pycaret.datasets import get\_data
2. datasets = get\_data('index')
3. dataset = get\_data("diabetes")

## ▼ import all classification methods

```
from pycaret.classification import *
```

## ▼ Setup function initializes the training environment

```
s= setup(data=dataset, target='DEATH_EVENT', train_size=0.75, silent=True)
```

	Description	Value
0	session_id	8441
1	Target	DEATH_EVENT
2	Target Type	Binary
3	Label Encoded	None
4	Original Data	(291, 13)
5	Missing Values	False
6	Numeric Features	6
7	Categorical Features	6
8	Ordinal Features	False
9	High Cardinality Features	False
10	High Cardinality Method	None
11	Transformed Train Set	(218, 26)
12	Transformed Test Set	(73, 26)
13	Shuffle Train-Test	True
14	Stratify Train-Test	False
15	Fold Generator	StratifiedKFold
16	Fold Number	10
17	CPU Jobs	-1
18	Use GPU	False
19	Log Experiment	False
20	Experiment Name	clf-default-name
21	USI	f80e
22	Imputation Type	simple
23	Iterative Imputation Iteration	None
24	Numeric Imputer	mean
25	Iterative Imputation Numeric Model	None
26	Categorical Imputer	constant
27	Iterative Imputation Categorical Model	None
28	Unknown Categoricals Handling	least_frequent
29	Normalize	False
30	Normalize Method	None

## ▼ Run and Compare all ML models

```
cm = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	(Se
<b>rf</b>	Random Forest Classifier	0.8530	0.8942	0.6238	0.8614	0.7143	0.6206	0.6407	0.41
<b>lightgbm</b>	Light Gradient Boosting Machine	0.8258	0.8605	0.6548	0.7556	0.6880	0.5703	0.5818	0.01
<b>gbc</b>	Gradient Boosting Classifier	0.8115	0.8783	0.6071	0.7302	0.6523	0.5264	0.5379	0.01
<b>lr</b>	Logistic Regression	0.8071	0.8283	0.5952	0.7414	0.6506	0.5206	0.5331	0.31
<b>ada</b>	Ada Boost Classifier	0.8024	0.8260	0.5929	0.7073	0.6404	0.5070	0.5130	0.01
<b>ridge</b>	Ridge Classifier	0.7883	0.0000	0.5881	0.6630	0.6045	0.4655	0.4797	0.01
<b>nb</b>	Naive Bayes	0.7840	0.8131	0.4738	0.7267	0.5597	0.4291	0.4522	0.01
<b>49</b>	Trigonometry Features				False				

## ▼ Create a model - Random Forest model

```
m = create_model('rf')
print (m)
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
<b>0</b>	0.9091	0.9688	0.6667	1.0000	0.8000	0.7442	0.7698
<b>1</b>	0.9091	0.8857	0.8571	0.8571	0.8571	0.7905	0.7905
<b>2</b>	0.8182	0.8571	0.5714	0.8000	0.6667	0.5464	0.5610
<b>3</b>	0.8636	0.8667	0.5714	1.0000	0.7273	0.6452	0.6901
<b>4</b>	0.7727	0.8905	0.4286	0.7500	0.5455	0.4086	0.4370
<b>-</b>	-	-	-	-	-	-	-

## ▼ Tune the Model

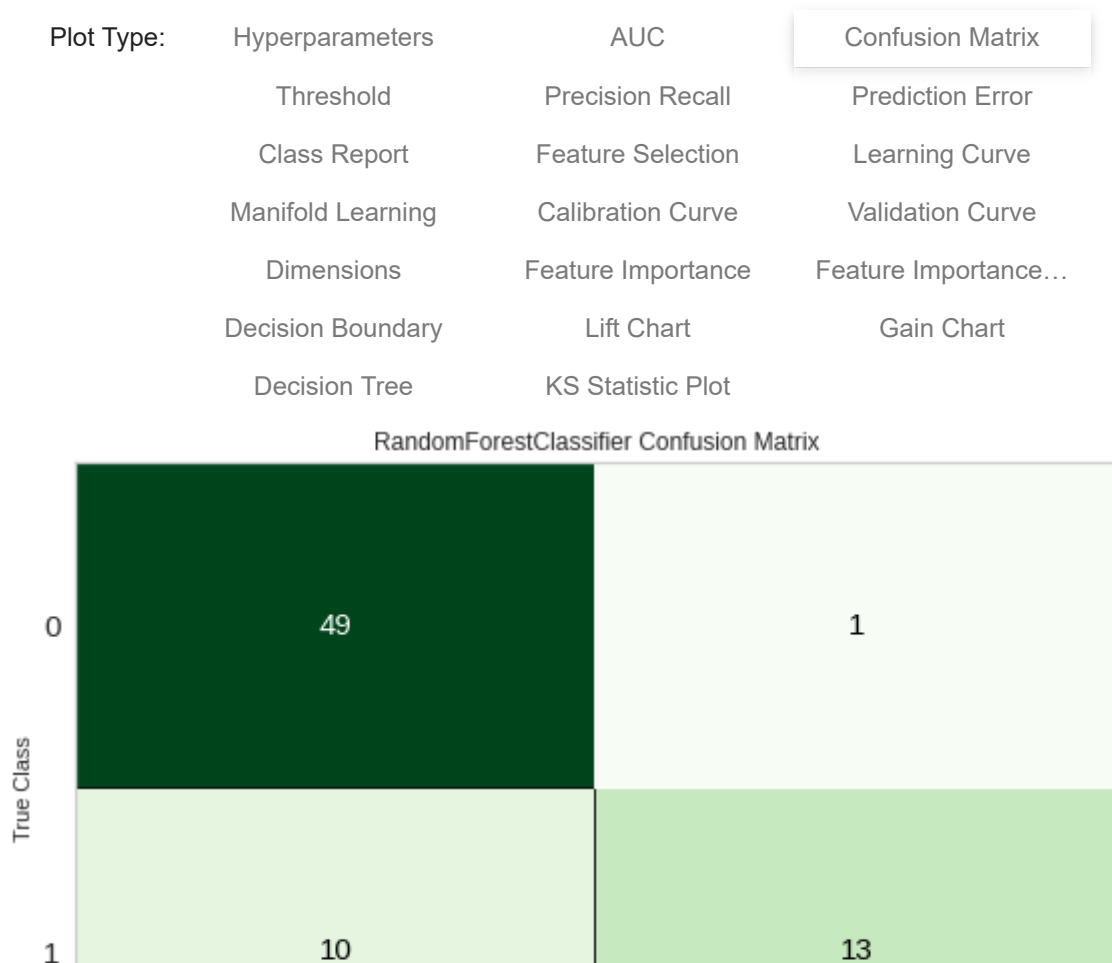
```
tuned_m = tune_model(m,n_iter = 50)
print(tuned_m)
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
<b>0</b>	0.9091	0.9583	0.6667	1.0000	0.8000	0.7442	0.7698
<b>1</b>	0.9091	0.8857	0.8571	0.8571	0.8571	0.7905	0.7905
<b>2</b>	0.7273	0.8476	0.2857	0.6667	0.4000	0.2584	0.2973
<b>3</b>	0.8636	0.8952	0.5714	1.0000	0.7273	0.6452	0.6901
<b>4</b>	0.7273	0.9048	0.2857	0.6667	0.4000	0.2584	0.2973
<b>5</b>	0.9545	0.9714	0.8571	1.0000	0.9231	0.8911	0.8964
<b>6</b>	0.9545	1.0000	1.0000	0.8750	0.9333	0.8991	0.9037
<b>7</b>	0.7273	0.8381	0.4286	0.6000	0.5000	0.3196	0.3281
<b>8</b>	0.8571	0.9111	0.5000	1.0000	0.6667	0.5882	0.6455
<b>9</b>	0.8571	0.7778	0.5000	1.0000	0.6667	0.5882	0.6455
<b>Mean</b>	0.8487	0.8990	0.5952	0.8665	0.6874	0.5983	0.6264
<b>SD</b>	0.0862	0.0635	0.2333	0.1550	0.1898	0.2336	0.2253

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight={},
                        criterion='gini', max_depth=7, max_features='log2',
                        max_leaf_nodes=None, max_samples=None,
                        min_impurity_decrease=0.0005, min_impurity_split=None,
                        min_samples_leaf=4, min_samples_split=5,
                        min_weight_fraction_leaf=0.0, n_estimators=170,
                        n_jobs=-1, oob_score=False, random_state=8441, verbose=0,
                        warm_start=False)
```

## ▼ Performance Analysis

```
evaluate_model(tuned_m)
```



Double-click (or enter) to edit

## ▼ Predict using Tunned Model

```
predict_model(tuned_m);
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Random Forest Classifier	0.8493	0.9226	0.5652	0.9286	0.7027	0.6096	0.6433

## ▼ Finalize the Model

```
final_m = finalize_model(tuned_m)
```

## ▼ Save the Model



```
save_model(final_m, 'Final-Model')
```

Transformation Pipeline and Model Successfully Saved

```
(Pipeline(memory=None,
  steps=[('dtypes',
    DataTypes_Auto_infer(categorical_features=[],
                        display_types=False, features_todrop=[],
                        id_columns=[],
                        ml_usecase='classification',
                        numerical_features=[],
                        target='DEATH_EVENT', time_features=[])),
    ('imputer',
     Simple_Imputer(categorical_strategy='not_available',
                    fill_value_categorical=None,
                    fill_value_numerical=None,
                    numeric...
    RandomForestClassifier(bootstrap=True, ccp_alpha=0.0,
                          class_weight={}, criterion='gini',
                          max_depth=7, max_features='log2',
                          max_leaf_nodes=None, max_samples=None,
                          min_impurity_decrease=0.0005,
                          min_impurity_split=None,
                          min_samples_leaf=4, min_samples_split=5,
                          min_weight_fraction_leaf=0.0,
                          n_estimators=170, n_jobs=-1,
                          oob_score=False, random_state=8441,
                          verbose=0, warm_start=False)]],
  verbose=False), 'Final-Model.pkl')
```

## ▼ Download Pickle File

```
from google.colab import files
files.download('Final-Model.pkl')
```

## ▼ Upload Pickle File

```
from google.colab import files
files.upload()
```

Choose files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving Final-Model.pkl to Final-Model.pkl

```
{'Final-Model.pkl': b'\x80\x03csklearn.pipeline\nPipeline\nq\x00)\x81q\x01}q\x02(X
```

## ▼ Load the Model

```
saved_final_m = load_model('Final-Model')
```

## Transformation Pipeline and Model Successfully Loaded

## ▼ Predict using unseen data

```
files.upload()
```

No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving heart\_failure\_unseen\_dataset.csv to heart\_failure\_unseen\_dataset.csv

```
{'heart_failure_unseen_dataset.csv': b'age,anaemia,creatinine_phosphokinase,diabet
```

```
data_unseen=pd.read_csv('heart_failure_unseen_dataset.csv')
```

```
new_prediction = predict_model(saved_final_m,data=data_unseen)
```

```
new_prediction.head()
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood
0	70	1	125	0	25	
1	48	1	582	1	55	
2	65	1	52	0	25	
3	65	1	128	1	30	
4	68	1	220	0	35	

Basic Level: <https://www.pycaret.org/tutorials/html/CLF101.html>

Intermediate Level: <https://www.pycaret.org/tutorials/html/CLF102.html>

Double-click (or enter) to edit

