LightGBM is a gradient boosting framework that uses tree based learning algorithms. It is designed to be distributed and efficient with the following advantages:

- Faster training speed and higher efficiency.
- Lower memory usage.
- Better accuracy.
- Support of parallel, distributed, and GPU learning.
- Capable of handling large-scale data.

```
!pip install mlbox

```

```
Collecting mlbox
  Downloading mlbox-0.8.5.tar.gz (31 kB)
Collecting numpy==1.18.2
  Downloading numpy-1.18.2-cp37-cp37m-manylinux1_x86_64.whl (20.2 MB)
     |████████████████████████████████| 20.2 MB 1.3 MB/s
Requirement already satisfied: scipy==1.4.1 in /usr/local/lib/python3.7/dist-packages (from mlbox) (1.4.1)
Collecting matplotlib==3.0.3
  Downloading matplotlib-3.0.3-cp37-cp37m-manylinux1_x86_64.whl (13.0 MB)
     |████████████████████████████████| 13.0 MB 26.7 MB/s
Collecting hyperopt==0.2.3
  Downloading hyperopt-0.2.3-py3-none-any.whl (1.9 MB)
     |████████████████████████████████| 1.9 MB 37.4 MB/s
Collecting pandas==0.25.3
  Downloading pandas-0.25.3-cp37-cp37m-manylinux1_x86_64.whl (10.4 MB)
     |████████████████████████████████| 10.4 MB 31.2 MB/s
Collecting joblib==0.14.1
  Downloading joblib-0.14.1-py2.py3-none-any.whl (294 kB)
     |████████████████████████████████| 294 kB 40.0 MB/s
Collecting scikit-learn==0.22.1
  Downloading scikit_learn-0.22.1-cp37-cp37m-manylinux1_x86_64.whl (7.0 MB)
     |████████████████████████████████| 7.0 MB 29.9 MB/s
Collecting tensorflow==2.0.0
  Downloading tensorflow-2.0.0-cp37-cp37m-manylinux2010_x86_64.whl (86.3 MB)
     |████████████████████████████████| 86.3 MB 24 kB/s
Collecting lightgbm==2.3.1
  Downloading lightgbm-2.3.1-py2.py3-none-manylinux1_x86_64.whl (1.2 MB)
     |████████████████████████████████| 1.2 MB 29.0 MB/s
Collecting tables==3.5.2
  Downloading tables-3.5.2-cp37-cp37m-manylinux1_x86_64.whl (4.3 MB)
     |████████████████████████████████| 4.3 MB 23.0 MB/s
Collecting xlrd==1.2.0
  Downloading xlrd-1.2.0-py2.py3-none-any.whl (103 kB)
     |████████████████████████████████| 103 kB 60.4 MB/s
Requirement already satisfied: future in /usr/local/lib/python3.7/dist-packages (from hyperopt==0.2.3->mlbox) (0.16.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from hyperopt==0.2.3->mlbox) (4.41.1)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.7/dist-packages (from hyperopt==0.2.3->mlbox) (1.3.0)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from hyperopt==0.2.3->mlbox) (1.15.0)
Collecting networkx==2.2
  Downloading networkx-2.2.zip (1.7 MB)
     |████████████████████████████████| 1.7 MB 21.6 MB/s
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib==3.0.3->mlbox) (1.3.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib==3.0.3->mlbox) (2.4.7)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib==3.0.3->mlbox) (0.10.0)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib==3.0.3->mlbox) (2.8.1)
Requirement already satisfied: decorator>=4.3.0 in /usr/local/lib/python3.7/dist-packages (from networkx==2.2->hyperopt==0.2.3->mlbox) (4.4.2)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-packages (from pandas==0.25.3->mlbox) (2018.9)
Collecting mock>=2.0
  Downloading mock-4.0.3-py3-none-any.whl (28 kB)
Requirement already satisfied: numexpr>=2.6.2 in /usr/local/lib/python3.7/dist-packages (from tables==3.5.2->mlbox) (2.7.3)
Requirement already satisfied: wrapt>=1.11.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.0.0->mlbox) (1.12.1)
Collecting gast==0.2.2
  Downloading gast-0.2.2.tar.gz (10 kB)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.0.0->mlbox) (3.3.0)
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.0.0->mlbox) (0.36.2)
Requirement already satisfied: astor>=0.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.0.0->mlbox) (0.8.1)
Requirement already satisfied: protobuf>=3.6.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.0.0->mlbox) (3.17.3)
Collecting tensorboard<2.1.0,>=2.0.0
  Downloading tensorboard-2.0.2-py3-none-any.whl (3.8 MB)
     |████████████████████████████████| 3.8 MB 22.3 MB/s
Requirement already satisfied: grpcio>=1.8.6 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.0.0->mlbox) (1.34.1)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.0.0->mlbox) (1.1.0)
Requirement already satisfied: absl-py>=0.7.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.0.0->mlbox) (0.12.0)
Collecting tensorflow-estimator<2.1.0,>=2.0.0
  Downloading tensorflow_estimator-2.0.1-py2.py3-none-any.whl (449 kB)
     |████████████████████████████████| 449 kB 57.1 MB/s
Collecting keras-applications>=1.0.8
  Downloading Keras_Applications-1.0.8-py3-none-any.whl (50 kB)
     |████████████████████████████████| 50 kB 4.7 MB/s
Requirement already satisfied: keras-preprocessing>=1.0.5 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.0.0->mlbox) (1.1.2)
Requirement already satisfied: google-pasta>=0.1.6 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.0.0->mlbox) (0.2.0)
Requirement already satisfied: h5py in /usr/local/lib/python3.7/dist-packages (from keras-applications>=1.0.8->tensorflow==2.0.0->mlbox) (3.1.0)
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.1.0,>=2.0.0->tensorflow==2.0.0->mlbox) (57.2.0)
Requirement already satisfied: google-auth<2,>=1.6.3 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.1.0,>=2.0.0->tensorflow==2.0.0->mlbox) (1.32.1)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.1.0,>=2.0.0->tensorflow==2.0.0->mlbox) (0.4.4)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.1.0,>=2.0.0->tensorflow==2.0.0->mlbox) (3.3.4)
```

```
from mlbox.preprocessing import *
from mlbox.optimisation import *
from mlbox.prediction import *
```

```
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from google-auth<2,>=1.6.3->tensorboard<2.1.0,>=2.0.0->tensorflow==2.0.0->mlbox) (0.2.8)
```

```
from numpy.random import RandomState
import pandas as pd

df = pd.read_csv('/content/diabetes2.csv')
rng = RandomState()

train = df.sample(frac=0.7, random_state=rng)
test = df.loc[~df.index.isin(train.index)]
```

```
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->markdown>=2.6.8->tensorboard<2.1.0,>=2.0.0->tensorflow==2.0.0->mlbox) (3.5.0)
```

```
train.to_csv('train.csv')
test.to_csv('test.csv')
```

```
  Stored in directory: /root/.cache/pip/wheels/c1/38/7d/46e966345596e58b8956650ec129f9386b9e63cac3a572c6f5
```

```
paths = ["/content/train.csv","/content/test.csv"]
target_name = "Outcome"
```

```
  Building wheel for gast (setup.py) ... done
```

```
rd = Reader(sep = ",")
df = rd.train_test_split(paths, target_name)
```

```
reading csv : train.csv ...
cleaning data ...
CPU time: 5.495320796966553 seconds

reading csv : test.csv ...
cleaning data ...
CPU time: 0.05363774299621582 seconds

You have no test dataset !

> Number of common features : 8

gathering and crunching for train and test datasets ...
reindexing for train and test datasets ...
dropping training duplicates ...
dropping constant variables on training set ...

> Number of categorical features: 0
> Number of numerical features: 8
> Number of training samples : 768
> Number of test samples : 0

> You have no missing values on train set...
```

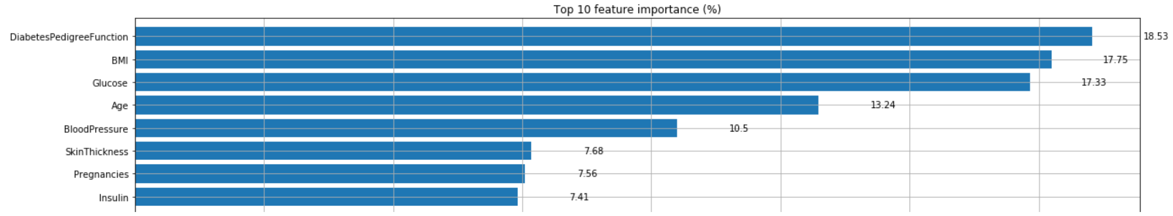
```
> Task : classification
0.0    500
1.0    268
Name: Outcome, dtype: int64

encoding target ...
  Attempting uninstall: tensorflow
```

```
#Defining Optimiser
opt = Optimiser(scoring = "accuracy", n_folds = 5)
#Defining the model
space = {
'est__strategy':{"search":"choice",
"space":["LightGBM"]},
'est__n_estimators':{"search":"choice",
"space":[150]},
'est__colsample_bytree':{"search":"uniform",
"space":[0.8,0.95]},
'est__subsample':{"search":"uniform",
"space":[0.8,0.95]},
'est__max_depth':{"search":"choice",
"space":[5,6,7,8,9]},
'est__learning_rate':{"search":"choice",
"space":[0.07]}
}
params = opt.optimise(space, df,15)
```

```
>>> NA ENCODER :{'numerical_strategy': 'mean', 'categorical_strategy': '<NULL>'}
>>> CA ENCODER :{'strategy': 'label_encoding'}
>>> ESTIMATOR :{'strategy': 'LightGBM', 'colsample_bytree': 0.8104138977154715, 'learning_rate': 0.07, 'max_depth': 9, 'n_estimators': 150, 'subsample': 0.8649816332998397, 'boosting_type': 'gbdt', '
MEAN SCORE : accuracy = 0.7357355063237416
VARIANCE : 0.038574222197282924 (fold 1 = 0.7662337662337663, fold 2 = 0.6818181818181818, fold 3 = 0.7142857142857143, fold 4 = 0.7254901960784313, fold 5 = 0.7908496732026143)
CPU time: 0.5867433547973633 seconds
##################################################### testing hyper-parameters... #####################################################
>>> NA ENCODER :{'numerical_strategy': 'mean', 'categorical_strategy': '<NULL>'}
>>> CA ENCODER :{'strategy': 'label_encoding'}
>>> ESTIMATOR :{'strategy': 'LightGBM', 'colsample_bytree': 0.8937539086850688, 'learning_rate': 0.07, 'max_depth': 8, 'n_estimators': 150, 'subsample': 0.8480816768832338, 'boosting_type': 'gbdt', '
MEAN SCORE : accuracy = 0.7487479840421016
VARIANCE : 0.02369042836474475 (fold 1 = 0.7597402597402597, fold 2 = 0.7142857142857143, fold 3 = 0.7337662337662337, fold 4 = 0.7516339869281046, fold 5 = 0.7843137254901961)
CPU time: 0.5842969417572021 seconds
##################################################### testing hyper-parameters... #####################################################
>>> NA ENCODER :{'numerical_strategy': 'mean', 'categorical_strategy': '<NULL>'}
>>> CA ENCODER :{'strategy': 'label_encoding'}
>>> ESTIMATOR :{'strategy': 'LightGBM', 'colsample_bytree': 0.8794905489955854, 'learning_rate': 0.07, 'max_depth': 7, 'n_estimators': 150, 'subsample': 0.9227355751965585, 'boosting_type': 'gbdt', '
MEAN SCORE : accuracy = 0.7383244206773619
VARIANCE : 0.03262708667356312 (fold 1 = 0.7792207792207793, fold 2 = 0.6883116883116883, fold 3 = 0.7142857142857143, fold 4 = 0.7581699346405228, fold 5 = 0.7516339869281046)
CPU time: 0.5579614639282227 seconds
##################################################### testing hyper-parameters... #####################################################
>>> NA ENCODER :{'numerical_strategy': 'mean', 'categorical_strategy': '<NULL>'}
>>> CA ENCODER :{'strategy': 'label_encoding'}
>>> ESTIMATOR :{'strategy': 'LightGBM', 'colsample_bytree': 0.9283293334686264, 'learning_rate': 0.07, 'max_depth': 6, 'n_estimators': 150, 'subsample': 0.9128672971591978, 'boosting_type': 'gbdt', '
MEAN SCORE : accuracy = 0.7279178338001867
VARIANCE : 0.02456588380157412 (fold 1 = 0.7467532467532467, fold 2 = 0.6948051948051948, fold 3 = 0.7012987012987013, fold 4 = 0.7516339869281046, fold 5 = 0.7450980392156863)
CPU time: 0.5391323566436768 seconds
##################################################### testing hyper-parameters... #####################################################
>>> NA ENCODER :{'numerical_strategy': 'mean', 'categorical_strategy': '<NULL>'}
>>> CA ENCODER :{'strategy': 'label_encoding'}
>>> ESTIMATOR :{'strategy': 'LightGBM', 'colsample_bytree': 0.9385757155703601, 'learning_rate': 0.07, 'max_depth': 7, 'n_estimators': 150, 'subsample': 0.944939486699379, 'boosting_type': 'gbdt', 'c
MEAN SCORE : accuracy = 0.750021220609456
VARIANCE : 0.023936681914832822 (fold 1 = 0.7857142857142857, fold 2 = 0.7272727272727273, fold 3 = 0.7207792207792207, fold 4 = 0.7516339869281046, fold 5 = 0.7647058823529411)
CPU time: 0.6363525390625 seconds
##################################################### testing hyper-parameters... #####################################################
>>> NA ENCODER :{'numerical_strategy': 'mean', 'categorical_strategy': '<NULL>'}
>>> CA ENCODER :{'strategy': 'label_encoding'}
>>> ESTIMATOR :{'strategy': 'LightGBM', 'colsample_bytree': 0.9263060042378931, 'learning_rate': 0.07, 'max_depth': 5, 'n_estimators': 150, 'subsample': 0.8396672461967501, 'boosting_type': 'gbdt', '

MEAN SCORE : accuracy = 0.7487649605296663
VARIANCE : 0.0309474733155663 (fold 1 = 0.7727272727272727, fold 2 = 0.7077922077922078, fold 3 = 0.7142857142857143, fold 4 = 0.7777777777777778, fold 5 = 0.7712418300653595)
CPU time: 0.49637937545776367 seconds
##################################################### testing hyper-parameters... #####################################################
>>> NA ENCODER :{'numerical_strategy': 'mean', 'categorical_strategy': '<NULL>'}
>>> CA ENCODER :{'strategy': 'label_encoding'}
>>> ESTIMATOR :{'strategy': 'LightGBM', 'colsample_bytree': 0.9189754100746531, 'learning_rate': 0.07, 'max_depth': 9, 'n_estimators': 150, 'subsample': 0.8809132787215199, 'boosting_type': 'gbdt', '
MEAN SCORE : accuracy = 0.7344198285374757
VARIANCE : 0.035103220164131026 (fold 1 = 0.7792207792207793, fold 2 = 0.6818181818181818, fold 3 = 0.7077922077922078, fold 4 = 0.7450980392156863, fold 5 = 0.7581699346405228)
CPU time: 0.5974433422088623 seconds
100%|██████████| 15/15 [00:09<00:00,  1.62trial/s, best loss: -0.750021220609456]


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ BEST HYPER-PARAMETERS ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

{'est__colsample_bytree': 0.9385757155703601, 'est__learning_rate': 0.07, 'est__max_depth': 7, 'est__n_estimators': 150, 'est__strategy': 'LightGBM', 'est__subsample': 0.944939486699379}
```

```
prd = Predictor()
prd.fit_predict(params, df);
```

```
fitting the pipeline ...
CPU time: 0.0971684455871582 seconds
```



✓ 0s completed at 10:36 AM