Hill Valley

A new differential evolution based on the Hill-Valley technique for multimodal optimization. This technique uses history information to classify individuals on the same peak as one species. Compared to other niching techniques, this technique is insensitive to parameters and does not waste additional computing resources.

Dataset:

Dataset for the project was downloaded from the github repository by YBI Foundation.

[1]:
```
#import libraries
import pandas as pd
```

[2]:
```
import numpy as np
```

[3]:
```
#import dataset
hill= pd.read_csv('Hill Valley Dataset.csv')
```

[4]:
```
hill.head()
```

[4]:
```
        V1       V2       V3       V4       V5       V6       V7  \
0     39.02    36.49  38.20     38.85    39.38    39.74     37.02
1     1.83     1.71   1.77      1.77     1.68     1.78      1.80
2     68177.69 66138.42 72981.88 74304.33 67549.66 69367.34 69169.41
3     44889.06 39191.86 40728.46 38576.36 45876.06 47034.00 46611.43
4     5.70     5.40   5.28      5.38     5.27     5.61      6.00

        V8       V9      V10  …     V92      V93      V94      V95  \
0     39.53    38.81   38.79 …    36.62    36.92    38.80    38.52
1     1.70     1.75   1.78 …      1.80     1.79     1.77     1.74
2     73268.61 74465.84 72503.37…73438.88 71053.35 71112.62 74916.48
3     37668.32 40980.89 38466.15…42625.67 40684.20 46960.73 44546.80
4     5.38     5.34    5.87 …     5.17     5.67     5.60     5.94

       V96      V97      V98      V99      V100      Class
0     38.07    36.73    39.46    37.50    39.10      0
1     1.74     1.80     1.78     1.75     1.69       1
2     72571.58 66348.97 71063.72 67404.27 74920.24   1
3     45410.53 47139.44 43095.68 40888.34 39615.19   0
```

```
4      5.73       5.22    5.30      5.73         5.91       0

[5 rows x 101 columns]
```

[5]:
```
#data describe
hill.describe()
```

[5]:
```
                  V1           V2           V3           V4  \
count   1212.000000  1212.000000  1212.000000  1212.000000
mean    8169.091881  8144.306262  8192.653738  8176.868738
std    17974.950461 17881.049734 18087.938901 17991.903982
min        0.920000     0.900000     0.850000     0.890000
25%       19.602500    19.595000    18.925000    19.277500
50%      301.425000   295.205000   297.260000   299.720000
75%     5358.795000  5417.847500  5393.367500  5388.482500
max    117807.870000 108896.480000 119031.350000 110212.590000

                  V5           V6           V7           V8  \
count   1212.000000  1212.000000  1212.000000  1212.000000
mean    8128.297211  8173.030008  8188.582748  8183.641543
std    17846.757963 17927.114105 18029.562695 18048.582159
min        0.880000     0.860000     0.870000     0.650000
25%       19.210000    19.582500    18.690000    19.062500
50%      295.115000   294.380000   295.935000   290.850000
75%     5321.987500  5328.040000  5443.977500  5283.655000
max    113000.470000 116848.390000 115609.240000 118522.320000

                  V9          V10 …          V92          V93  \
count   1212.000000  1212.000000 …  1212.000000  1212.000000
mean    8154.670066  8120.767574 …  8120.056815  8125.917409
std    17982.390713 17900.798206 … 17773.190621 17758.182403
min        0.650000     0.620000 …     0.870000     0.900000
25%       19.532500    19.285000 …    19.197500    18.895000
50%      294.565000   295.160000 …   297.845000   295.420000
75%     5378.180000  5319.097500 …  5355.355000  5386.037500
max    112895.900000 117798.300000 … 113858.680000
       112948.830000

                 V94          V95          V96          V97  \
count   1212.000000  1212.000000  1212.000000  1212.000000
mean    8158.793812  8140.885421  8213.480611  8185.594002
std    17919.510371 17817.945646 18016.445265 17956.084223
min        0.870000     0.880000     0.890000     0.890000
25%       19.237500    19.385000    19.027500    19.135000
50%      299.155000   293.355000   301.370000   296.960000
75%     5286.385000  5345.797500  5300.890000  5361.047500
```

```
max    112409.570000 112933.730000 112037.220000 115110.420000

                 V98           V99          V100       Class
count 1212.000000   1212.000000  1212.000000 1212.000000
mean   8140.195355   8192.960891  8156.197376    0.500000
std   17768.356106 18064.781479 17829.310973    0.500206
min       0.860000      0.910000      0.890000    0.000000
25%      19.205000     18.812500     19.145000    0.000000
50%     300.925000    299.200000    302.275000    0.500000
75%    5390.850000   5288.712500   5357.847500   1.000000
max   116431.960000 113291.960000 114533.760000 1.000000

[8 rows x 101 columns]
```

[6]:
```
#data preprocessing
hill.columns
```

[6]:
```
Index(['V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
       …
 'V92', 'V93', 'V94', 'V95', 'V96', 'V97', 'V98', 'V99', 'V100',
        'Class'],
       dtype='object',
       length=101)
```

[7]:
```
hill['Class'].value_counts()
```

[7]:
```
0    606
1         606
Name: Class, dtype: int64
```

[8]:
```
#define target(y) and feature(X)
y=hill['Class']
```

[9]:
```
y.shape
```

[9]:
```
(1212,)
```

[10]:
```
y
```

[10]:
```
0      0
1      1
2      1
3      0
4      0
```

```
..
1207    1
1208    0
1209    1
1210    1
1211    0
Name: Class, Length: 1212, dtype: int64
```

[11]:
```
X=hill.drop('Class',axis=1)
```

[12]:
```
X.shape
```

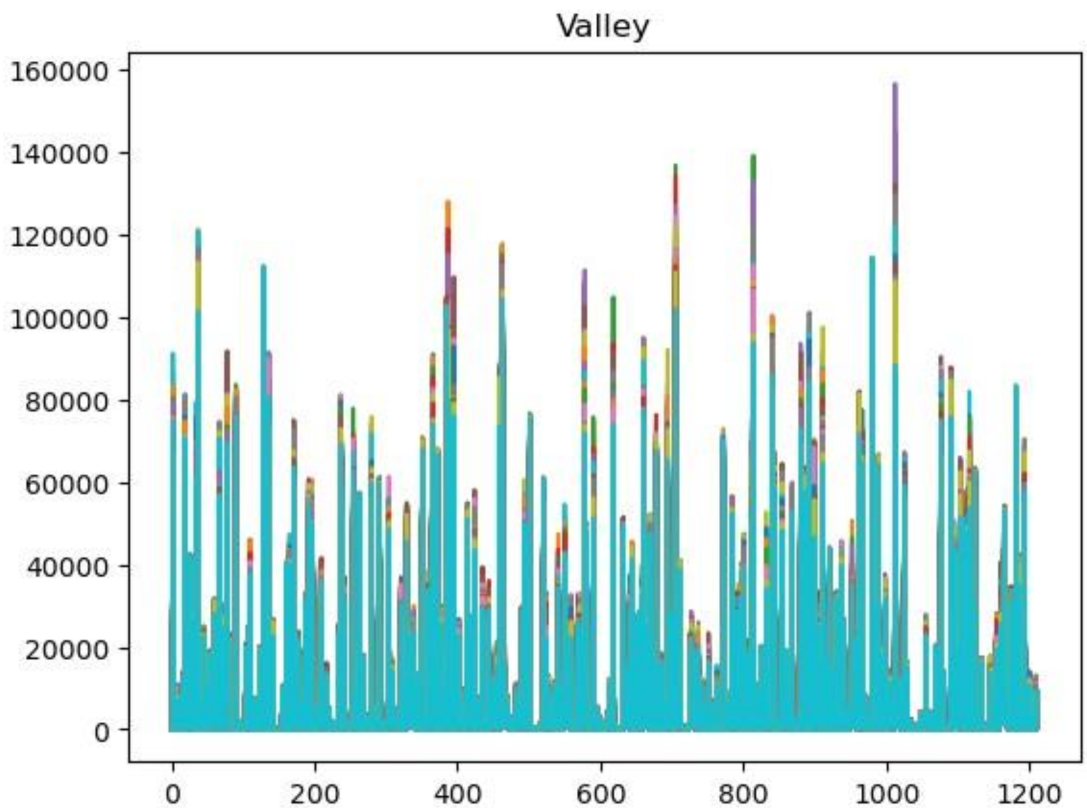
[12]:
```
(1212, 100)
```

[13]:
```
X
```

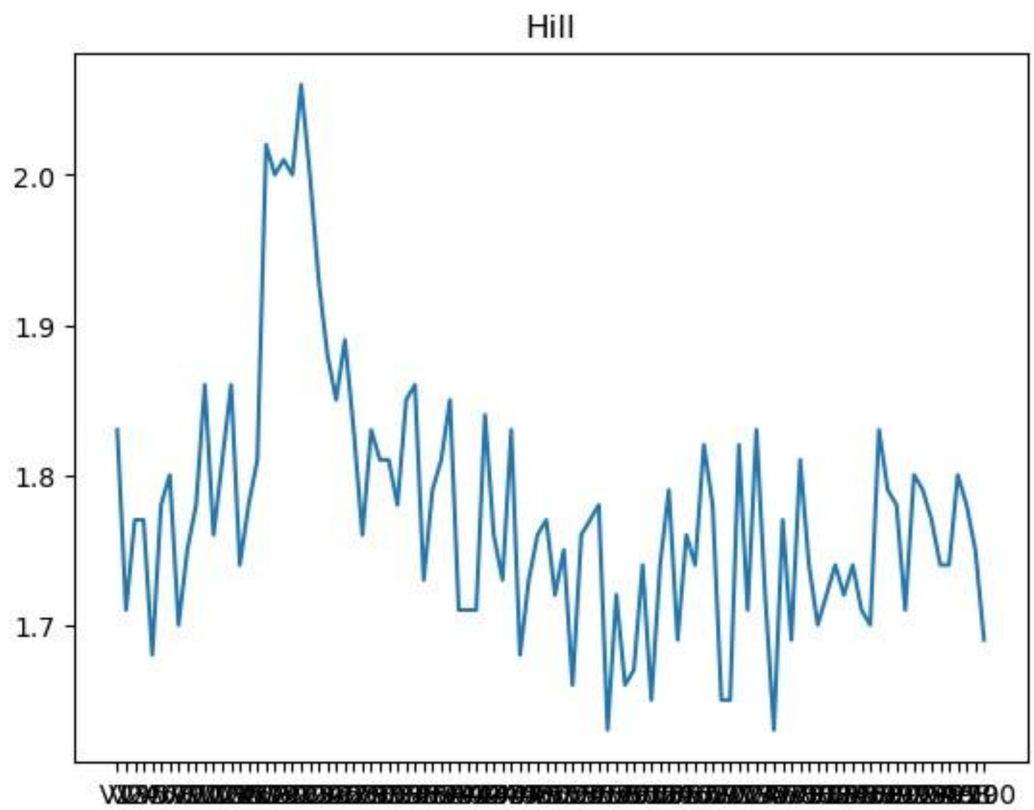
[13]:
```
            V1        V2        V3        V4        V5        V6        V7  \
0        39.02     36.49     38.20     38.85     39.38     39.74     37.02
1         1.83      1.71      1.77      1.77      1.68      1.78      1.80
2     68177.69  66138.42  72981.88  74304.33  67549.66  69367.34  69169.41
3     44889.06  39191.86  40728.46  38576.36  45876.06  47034.00  46611.43
4         5.70      5.40      5.28      5.38      5.27      5.61      6.00
...        ...       ...       ...       ...       ...       ...       ...
1207     13.00     12.87     13.27     13.04     13.19     12.53     14.31
1208     48.66     50.11     48.55     50.43     50.09     49.67     48.95
1209  10160.65   9048.63   8994.94   9514.39   9814.74  10195.24  10031.47
1210     34.81     35.07     34.98     32.37     34.16     34.03     33.31
1211   8489.43   7672.98   9132.14   7985.73   8226.85   8554.28   8838.87

            V8        V9       V10  ...       V91       V92       V93  \
0        39.53     38.81     38.79  ...     37.57     36.62     36.92
1         1.70      1.75      1.78  ...      1.71      1.80      1.79
2     73268.61  74465.84  72503.37  ...  69384.71  73438.88  71053.35
3     37668.32  40980.89  38466.15  ...  47653.60  42625.67  40684.20
4         5.38      5.34      5.87  ...      5.52      5.17      5.67
...        ...       ...       ...  ...       ...       ...       ...
1207     13.33     13.63     14.55  ...     12.89     12.48     12.15
1208     48.65     48.63     48.61  ...     47.45     46.93     49.61
1209  10202.28   9152.99   9591.75  ...  10413.41   9068.11   9191.80
1210     32.48     35.63     32.48  ...     33.18     32.76     35.03
1211   8967.24   8635.14   8544.37  ...   7747.70   8609.73   9209.48

           V94       V95       V96       V97       V98       V99      V100
0        38.80     38.52     38.07     36.73     39.46     37.50     39.10
1         1.77      1.74      1.74      1.80      1.78      1.75      1.69
2     71112.62  74916.48  72571.58  66348.97  71063.72  67404.27  74920.24
3     46960.73  44546.80  45410.53  47139.44  43095.68  40888.34  39615.19
4         5.60      5.94      5.73      5.22      5.30      5.73      5.91
...        ...       ...       ...       ...       ...       ...       ...
1207     13.15     12.35     13.58     13.86     12.88     13.87     13.51
1208     47.16     48.17     47.94     49.81     49.89     47.43     47.77
1209   9275.04   9848.18   9074.17   9601.74  10366.24   8997.60   9305.77
1210     32.89     31.91     33.85     35.28     32.49     32.83     34.82
1211   8496.33   8724.01   8219.99   8550.86   8679.43   8389.31   8712.80

[1212 rows x 100 columns]
```

[14]:
```
#data visualization
import matplotlib.pyplot as plt
```

[15]:
```
plt.plot(X.iloc[0:])
plt.title('Valley');
```



[16]:
```
plt.plot(X.iloc[1,:])
plt.title('Hill');
```




```
[17]: #train test split
      from sklearn.model_selection import train_test_split
```

```
[18]: X_train, X_test, y_train, y_test= train_test_split(X,y, test_size=0.3,␣
       ↪stratify=y,random_state=2529)
```

```
[19]: X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
[19]: ((848, 100), (364, 100), (848,), (364,))
```

```
[20]: #modelling
      from sklearn.linear_model import LogisticRegression
```

```
[21]: LR=LogisticRegression()
```

```
[22]: LR.fit(X_train,y_train)
```

[22]: LogisticRegression()

[23]:
```
#model prediction
y_pred=LR.predict(X_test)
```

[24]:
```
y_pred
```

```
[24]: array([0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1,
 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0,
                                                      1, 0, 1, 1,
        0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
        1,
        0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0,
        0,
        1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1,
        1,
        1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1,
        0,
        0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,
        0,
        0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
        1,
        0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0,
        1,
        1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1,
        1,
        0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1,
        1,
        0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1,
        0,
        1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0,
        0,
        1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
        0,
        1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0,
        0,
        0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1,
        0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1], dtype=int64)
```

[25]:
```
y_pred.shape
```

[25]: (364,)

[26]:
```
LR.predict_proba(X_test)
```

```
[26]: array([[1.00000000e+000, 6.96486686e-091],
       [2.48030307e-009, 9.99999998e-001],
```

```
[1.00000000e+000, 5.46284573e-130],
[6.69059698e-004, 9.99330940e-001],
[2.82744567e-002, 9.71725543e-001],
[6.49479408e-001, 3.50520592e-001],
[2.50031782e-001, 7.49968218e-001],
[1.00000000e+000, 0.00000000e+000],
[9.93287657e-001, 6.71234274e-003],
[9.97792248e-001, 2.20775201e-003],
[2.33755160e-002, 9.76624484e-001],
[9.99999997e-001, 3.35332757e-009],
[1.00000000e+000, 0.00000000e+000],
[9.97394705e-001, 2.60529500e-003],
[1.00000000e+000, 2.14507861e-033],
[1.00000000e+000, 0.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[3.21341711e-002, 9.67865829e-001],
[3.87365239e-001, 6.12634761e-001],
[5.30917007e-001, 4.69082993e-001],
[0.00000000e+000, 1.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[5.30758692e-001, 4.69241308e-001],
[0.00000000e+000, 1.00000000e+000],
[6.58984028e-001, 3.41015972e-001],
[2.80173324e-003, 9.97198267e-001],
[7.21355833e-001, 2.78644167e-001],
[8.55841396e-008, 9.99999914e-001],
[4.80944001e-001, 5.19055999e-001],
[6.42363138e-003, 9.93576369e-001],
[0.00000000e+000, 1.00000000e+000],
[1.00000000e+000, 3.31526659e-014],
[0.00000000e+000, 1.00000000e+000],
[1.00000000e+000, 0.00000000e+000],
[5.26841874e-001, 4.73158126e-001],
[5.01970776e-001, 4.98029224e-001],
[5.22703908e-001, 4.77296092e-001],
[3.92097426e-001, 6.07902574e-001],
[9.99999999e-001, 1.23393821e-009],
[1.00000000e+000, 1.13558927e-097],
[4.70480094e-001, 5.29519906e-001],
[1.00000000e+000, 3.49558188e-028],
[0.00000000e+000, 1.00000000e+000],
[3.78835857e-003, 9.96211641e-001],
[1.00000000e+000, 2.11330701e-034],
[0.00000000e+000, 1.00000000e+000],
[2.13538467e-001, 7.86461533e-001],
```

```
[1.38686495e-006, 9.99998613e-001],
[1.25512933e-011, 1.00000000e+000],
[6.56645535e-001, 3.43354465e-001],
[5.13450454e-001, 4.86549546e-001],
[1.00000000e+000, 1.27963682e-058],
[4.01860248e-001, 5.98139752e-001],
[4.94022955e-003, 9.95059770e-001],
[1.00000000e+000, 3.01608679e-014],
[4.37641136e-001, 5.62358864e-001],
[4.93181381e-001, 5.06818619e-001],
[3.93556076e-001, 6.06443924e-001],
[1.00000000e+000, 8.97249563e-147],
[0.00000000e+000, 1.00000000e+000],
[1.03513831e-007, 9.99999896e-001],
[4.72629224e-001, 5.27370776e-001],
[4.84177056e-001, 5.15822944e-001],
[0.00000000e+000, 1.00000000e+000],
[9.84817518e-001, 1.51824823e-002],
[4.31352705e-001, 5.68647295e-001],
[1.00000000e+000, 6.78159043e-096],
[0.00000000e+000, 1.00000000e+000],
[5.54839113e-001, 4.45160887e-001],
[0.00000000e+000, 1.00000000e+000],
[8.11405128e-001, 1.88594872e-001],
[4.96822868e-001, 5.03177132e-001],
[9.99939221e-001, 6.07792501e-005],
[2.31290100e-001, 7.68709900e-001],
[9.52120879e-001, 4.78791208e-002],
[1.00000000e+000, 1.31496431e-288],
[0.00000000e+000, 1.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[1.00000000e+000, 0.00000000e+000],
[1.00000000e+000, 1.82830241e-066],
[4.57699939e-001, 5.42300061e-001],
[5.10479577e-001, 4.89520423e-001],
[1.01109786e-001, 8.98890214e-001],
[0.00000000e+000, 1.00000000e+000],
[4.52984234e-001, 5.47015766e-001],
[3.96226270e-008, 9.99999960e-001],
[5.16517026e-001, 4.83482974e-001],
[9.01193555e-001, 9.88064453e-002],
[4.58148004e-001, 5.41851996e-001],
[4.31409863e-004, 9.99568590e-001],
[1.00000000e+000, 0.00000000e+000],
[5.16593100e-001, 4.83406900e-001],
```

```
[9.98715917e-001, 1.28408313e-003],
[1.00000000e+000, 6.47677010e-035],
[4.87577325e-001, 5.12422675e-001],
[0.00000000e+000, 1.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[1.00000000e+000, 1.11582721e-221],
[5.50006406e-001, 4.49993594e-001],
[1.00000000e+000, 4.53194168e-100],
[1.00000000e+000, 4.96124638e-021],
[0.00000000e+000, 1.00000000e+000],
[1.00000000e+000, 1.94352582e-099],
[8.39361007e-001, 1.60638993e-001],
[8.32151375e-007, 9.99999168e-001],
[1.00000000e+000, 0.00000000e+000],
[3.28348104e-001, 6.71651896e-001],
[4.39279724e-011, 1.00000000e+000],
[4.88595746e-001, 5.11404254e-001],
[7.78925147e-002, 9.22107485e-001],
[0.00000000e+000, 1.00000000e+000],
[3.07234537e-001, 6.92765463e-001],
[9.85778383e-001, 1.42216171e-002],
[5.55626720e-001, 4.44373280e-001],
[1.00000000e+000, 7.49354032e-188],
[0.00000000e+000, 1.00000000e+000],
[1.97997386e-001, 8.02002614e-001],
[2.70444400e-009, 9.99999997e-001],
[1.00000000e+000, 0.00000000e+000],
[1.00000000e+000, 8.17425816e-017],
[9.97412089e-001, 2.58791134e-003],
[0.00000000e+000, 1.00000000e+000],
[5.39057016e-001, 4.60942984e-001],
[0.00000000e+000, 1.00000000e+000],
[5.71240324e-001, 4.28759676e-001],
[5.66713027e-001, 4.33286973e-001],
[2.63837472e-002, 9.73616253e-001],
[9.99999809e-001, 1.90615221e-007],
[9.79348400e-001, 2.06516004e-002],
[1.00000000e+000, 1.03730932e-034],
[7.61207660e-006, 9.99992388e-001],
[1.00000000e+000, 5.51367971e-064],
[8.62075122e-001, 1.37924878e-001],
[0.00000000e+000, 1.00000000e+000],
[1.00000000e+000, 3.54741032e-078],
[4.06829123e-001, 5.93170877e-001],
[1.00000000e+000, 7.99146553e-012],
```

```
[4.45582004e-001, 5.54417996e-001],
[0.00000000e+000, 1.00000000e+000],
[2.82229545e-002, 9.71777046e-001],
[4.70436157e-001, 5.29563843e-001],
[1.00000000e+000, 4.60496138e-288],
[6.16100455e-001, 3.83899545e-001],
[9.94763696e-001, 5.23630378e-003],
[7.14106206e-001, 2.85893794e-001],
[1.00000000e+000, 1.16085768e-059],
[4.95214594e-001, 5.04785406e-001],
[1.00000000e+000, 0.00000000e+000],
[6.28230865e-001, 3.71769135e-001],
[5.33369605e-001, 4.66630395e-001],
[4.40855214e-001, 5.59144786e-001],
[5.27321641e-001, 4.72678359e-001],
[1.00000000e+000, 0.00000000e+000],
[5.32636530e-001, 4.67363470e-001],
[5.19438920e-001, 4.80561080e-001],
[0.00000000e+000, 1.00000000e+000],
[5.21477925e-001, 4.78522075e-001],
[0.00000000e+000, 1.00000000e+000],
[1.00000000e+000, 1.99318885e-067],
[1.00000000e+000, 8.60713830e-011],
[5.44757954e-001, 4.55242046e-001],
[1.00000000e+000, 0.00000000e+000],
[9.99927425e-001, 7.25748034e-005],
[9.42951160e-001, 5.70488400e-002],
[3.56559316e-001, 6.43440684e-001],
[1.00000000e+000, 5.42669958e-238],
[1.00000000e+000, 6.10350817e-122],
[0.00000000e+000, 1.00000000e+000],
[1.00000000e+000, 1.77957283e-019],
[9.98123960e-001, 1.87604024e-003],
[9.99999461e-001, 5.38839266e-007],
[5.83780040e-001, 4.16219960e-001],
[1.00000000e+000, 5.30020341e-274],
[9.17128412e-001, 8.28715878e-002],
[1.00000000e+000, 1.08523549e-079],
[4.63629135e-013, 1.00000000e+000],
[1.00000000e+000, 0.00000000e+000],
[5.19055862e-001, 4.80944138e-001],
[1.00000000e+000, 0.00000000e+000],
[1.00000000e+000, 1.84279107e-099],
[8.36575264e-001, 1.63424736e-001],
[6.09710342e-001, 3.90289658e-001],
```

```
[4.57818951e-001, 5.42181049e-001],
[1.00000000e+000, 0.00000000e+000],
[4.97544625e-001, 5.02455375e-001],
[8.03322086e-001, 1.96677914e-001],
[0.00000000e+000, 1.00000000e+000],
[9.99999999e-001, 1.40528443e-009],
[4.85156250e-004, 9.99514844e-001],
[0.00000000e+000, 1.00000000e+000],
[5.11007529e-001, 4.88992471e-001],
[0.00000000e+000, 1.00000000e+000],
[4.57411886e-014, 1.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[4.53440432e-001, 5.46559568e-001],
[8.34649138e-001, 1.65350862e-001],
[1.00000000e+000, 3.33901168e-017],
[2.58520627e-006, 9.99997415e-001],
[7.31378313e-009, 9.99999993e-001],
[2.00488627e-004, 9.99799511e-001],
[5.24710859e-001, 4.75289141e-001],
[1.00000000e+000, 1.83443204e-032],
[1.00000000e+000, 2.47169714e-023],
[0.00000000e+000, 1.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[7.58571318e-002, 9.24142868e-001],
[4.90775023e-001, 5.09224977e-001],
[3.10559048e-001, 6.89440952e-001],
[4.63377521e-001, 5.36622479e-001],
[1.00000000e+000, 1.25643252e-081],
[1.00000000e+000, 3.32125665e-161],
[3.75360050e-001, 6.24639950e-001],
[0.00000000e+000, 1.00000000e+000],
[1.00000000e+000, 8.84388796e-209],
[0.00000000e+000, 1.00000000e+000],
[5.20740539e-001, 4.79259461e-001],
[3.30213743e-001, 6.69786257e-001],
[5.18133858e-001, 4.81866142e-001],
[0.00000000e+000, 1.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[5.71177575e-001, 4.28822425e-001],
[1.24979599e-008, 9.99999988e-001],
[1.00000000e+000, 0.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[5.06281906e-001, 4.93718094e-001],
[0.00000000e+000, 1.00000000e+000],
[4.68875319e-001, 5.31124681e-001],
```

```
[1.54985501e-001, 8.45014499e-001],
[0.00000000e+000, 1.00000000e+000],
[5.15755088e-001, 4.84244912e-001],
[0.00000000e+000, 1.00000000e+000],
[4.39712140e-001, 5.60287860e-001],
[0.00000000e+000, 1.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[8.65262938e-001, 1.34737062e-001],
[1.00000000e+000, 4.10886234e-032],
[0.00000000e+000, 1.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[2.60093524e-001, 7.39906476e-001],
[8.10192955e-001, 1.89807045e-001],
[0.00000000e+000, 1.00000000e+000],
[1.38668484e-003, 9.98613315e-001],
[1.00000000e+000, 0.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[4.17509089e-001, 5.82490911e-001],
[9.99794090e-001, 2.05909536e-004],
[1.00000000e+000, 0.00000000e+000],
[9.76835309e-004, 9.99023165e-001],
[2.05677670e-008, 9.99999979e-001],
[5.15967664e-001, 4.84032336e-001],
[5.35275140e-001, 4.64724860e-001],
[0.00000000e+000, 1.00000000e+000],
[3.94914847e-001, 6.05085153e-001],
[0.00000000e+000, 1.00000000e+000],
[5.06684683e-001, 4.93315317e-001],
[0.00000000e+000, 1.00000000e+000],
[1.00000000e+000, 5.10696921e-064],
[5.10456968e-001, 4.89543032e-001],
[4.79371316e-001, 5.20628684e-001],
[8.05766832e-001, 1.94233168e-001],
[0.00000000e+000, 1.00000000e+000],
[1.00000000e+000, 4.30617601e-060],
[6.66133815e-016, 1.00000000e+000],
[5.18248914e-001, 4.81751086e-001],
[7.34823285e-004, 9.99265177e-001],
[1.30017836e-002, 9.86998216e-001],
[4.18348380e-001, 5.81651620e-001],
[5.05307777e-001, 4.94692223e-001],
[1.00000000e+000, 2.11834274e-212],
[4.92538693e-001, 5.07461307e-001],
[1.87292590e-001, 8.12707410e-001],
[3.64353661e-001, 6.35646339e-001],
```

```
[9.99999545e-001, 4.54517253e-007],
[5.53272389e-001, 4.46727611e-001],
[1.50149920e-003, 9.98498501e-001],
[0.00000000e+000, 1.00000000e+000],
[4.48984742e-001, 5.51015258e-001],
[1.00000000e+000, 0.00000000e+000],
[4.81946630e-001, 5.18053370e-001],
[4.31394453e-001, 5.68605547e-001],
[4.59887587e-001, 5.40112413e-001],
[4.79039705e-001, 5.20960295e-001],
[1.00000000e+000, 6.59497127e-014],
[9.51601601e-001, 4.83983987e-002],
[9.40395879e-001, 5.96041210e-002],
[9.99999195e-001, 8.05254117e-007],
[4.84969593e-001, 5.15030407e-001],
[4.82408623e-001, 5.17591377e-001],
[0.00000000e+000, 1.00000000e+000],
[5.19770178e-001, 4.80229822e-001],
[9.90922363e-001, 9.07763710e-003],
[9.23151390e-001, 7.68486103e-002],
[0.00000000e+000, 1.00000000e+000],
[5.05193928e-001, 4.94806072e-001],
[0.00000000e+000, 1.00000000e+000],
[1.00000000e+000, 4.80977680e-036],
[0.00000000e+000, 1.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[4.27810668e-001, 5.72189332e-001],
[0.00000000e+000, 1.00000000e+000],
[2.81383580e-001, 7.18616420e-001],
[5.01106913e-001, 4.98893087e-001],
[4.92286023e-001, 5.07713977e-001],
[3.32632810e-010, 1.00000000e+000],
[2.01969439e-003, 9.97980306e-001],
[0.00000000e+000, 1.00000000e+000],
[5.17078688e-001, 4.82921312e-001],
[0.00000000e+000, 1.00000000e+000],
[6.77487063e-002, 9.32251294e-001],
[5.05549168e-001, 4.94450832e-001],
[6.43967404e-001, 3.56032596e-001],
[9.62545553e-006, 9.99990375e-001],
[0.00000000e+000, 1.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[4.95528090e-001, 5.04471910e-001],
[9.99740531e-001, 2.59469481e-004],
```

```
[1.00000000e+000, 0.00000000e+000],
[6.37103885e-001, 3.62896115e-001],
[6.28201781e-001, 3.71798219e-001],
[6.85642984e-001, 3.14357016e-001],
[0.00000000e+000, 1.00000000e+000],
[4.84618516e-001, 5.15381484e-001],
[0.00000000e+000, 1.00000000e+000],
[7.62148324e-001, 2.37851676e-001],
[0.00000000e+000, 1.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[2.22044605e-016, 1.00000000e+000],
[1.00000000e+000, 0.00000000e+000],
[7.23549778e-001, 2.76450222e-001],
[1.00000000e+000, 9.67482463e-309],
[7.95488616e-001, 2.04511384e-001],
[0.00000000e+000, 1.00000000e+000],
[0.00000000e+000, 1.00000000e+000],
[9.99981014e-001, 1.89861253e-005],
[0.00000000e+000, 1.00000000e+000],
[2.55906477e-001, 7.44093523e-001],
[1.00000000e+000, 1.95026509e-016],
[9.63248637e-001, 3.67513635e-002],
[1.00000000e+000, 0.00000000e+000],
[6.10731510e-001, 3.89268490e-001],
[1.00000000e+000, 5.77527956e-022],
[1.00000000e+000, 6.52708450e-301],
[3.59196605e-001, 6.40803395e-001],
[0.00000000e+000, 1.00000000e+000],
[1.00000000e+000, 0.00000000e+000],
[9.99999763e-001, 2.37043258e-007],
[9.99531036e-001, 4.68964240e-004],
[1.11386166e-005, 9.99988861e-001],
[8.52669416e-001, 1.47330584e-001],
[6.40045840e-001, 3.59954160e-001],
[8.13197971e-001, 1.86802029e-001],
[1.00000000e+000, 7.23276100e-079],
[2.22044605e-016, 1.00000000e+000],
[5.64748811e-001, 4.35251189e-001],
[0.00000000e+000, 1.00000000e+000],
[5.57096716e-004, 9.99442903e-001],
[5.16827104e-001, 4.83172896e-001],
[1.00000000e+000, 0.00000000e+000],
[5.55255851e-001, 4.44744149e-001],
[4.52972869e-001, 5.47027131e-001],
[0.00000000e+000, 1.00000000e+000],
```

```
        [2.40286649e-001, 7.59713351e-001],
        [0.00000000e+000, 1.00000000e+000]])
```

```
[27]: #model evaluation
      from sklearn.metrics import confusion_matrix, classification_report
```

```
[28]: print(confusion_matrix(y_test,y_pred))
```

```
[[176   6]
 [  4 178]]
```

```
[29]: print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.98      0.97      0.97       182
           1       0.97      0.98      0.97       182

    accuracy                           0.97       364
   macro avg       0.97      0.97      0.97       364
Weighted avg       0.97      0.97      0.97       364
```

Conclusion:
Logistic regression algorithm was applied on the imported dataset and the model is been predicted and evaluated based on the attributes.