

Boolean_Classification

February 13, 2022

1 Boolean Classification

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

```
[2]: from pandas import DataFrame
```

```
[3]: df_dataset = pd.read_excel("diabetes.xlsx")
```

```
[4]: df_dataset
```

```
[4]:
```

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | \ |
|-----|-------------|---------|---------------|---------------|---------|------|---|
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 | |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 | |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 | |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 | |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 | |
| .. | ... | ... | ... | ... | ... | ... | |
| 763 | 10 | 101 | 76 | 48 | 180 | 32.9 | |
| 764 | 2 | 122 | 70 | 27 | 0 | 36.8 | |
| 765 | 5 | 121 | 72 | 23 | 112 | 26.2 | |
| 766 | 1 | 126 | 60 | 0 | 0 | 30.1 | |
| 767 | 1 | 93 | 70 | 31 | 0 | 30.4 | |

| | DiabetesPedigreeFunction | Age | Outcome |
|-----|--------------------------|-----|---------|
| 0 | 0.627 | 50 | 1 |
| 1 | 0.351 | 31 | 0 |
| 2 | 0.672 | 32 | 1 |
| 3 | 0.167 | 21 | 0 |
| 4 | 2.288 | 33 | 1 |
| .. | ... | ... | ... |
| 763 | 0.171 | 63 | 0 |
| 764 | 0.340 | 27 | 0 |
| 765 | 0.245 | 30 | 0 |
| 766 | 0.349 | 47 | 1 |
| 767 | 0.315 | 23 | 0 |

```
[768 rows x 9 columns]
```

```
[5]: y = df_dataset["Outcome"]
X = df_dataset.drop(["Outcome"], axis = 1)
```

```
[6]: X
```

```
[6]:      Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0             6      148            72           35         0  33.6
1             1       85            66           29         0  26.6
2             8      183            64            0         0  23.3
3             1       89            66           23        94  28.1
4             0      137            40           35       168  43.1
..          ...      ...            ...           ...         ...  ...
763          10      101            76           48       180  32.9
764           2      122            70           27         0  36.8
765           5      121            72           23       112  26.2
766           1      126            60            0         0  30.1
767           1       93            70           31         0  30.4
```

```
      DiabetesPedigreeFunction  Age
0                0.627    50
1                0.351    31
2                0.672    32
3                0.167    21
4                2.288    33
..                ...    ...
763              0.171    63
764              0.340    27
765              0.245    30
766              0.349    47
767              0.315    23
```

```
[768 rows x 8 columns]
```

```
[7]: y
```

```
[7]: 0      1
1      0
2      1
3      0
4      1
..
763    0
764    0
765    0
766    1
767    0
Name: Outcome, Length: 768, dtype: int64
```

```
[8]: from sklearn.model_selection import train_test_split
```

```
[9]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30,
↳ random_state=26)
```

```
[10]: X_train
```

```
[10]:      Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI   \
233             4      122             68             0           0  35.0
413             1      143             74             22          61  26.2
338             9      152             78             34         171  34.2
349             5           0             80             32           0  41.0
658            11      127            106             0           0  39.0
..            ...      ...             ...             ...      ...
77              5          95             72             33           0  37.7
723             5         117             86             30         105  39.1
102             0         125             96              0           0  22.5
432             1          80             74             11          60  30.0
318             3         115             66             39         140  38.1
```

```
      DiabetesPedigreeFunction  Age
233                0.394      29
413                0.256      21
338                0.893      33
349                0.346      37
658                0.190      51
..                ... ..
77                0.370      27
723                0.251      42
102                0.262      21
432                0.527      22
318                0.150      28
```

```
[537 rows x 8 columns]
```

```
[11]: y_train
```

```
[11]: 233    0
413    0
338    1
349    1
658    0
..
77     0
723    0
102    0
432    0
```

```
318      0
Name: Outcome, Length: 537, dtype: int64
```

```
[12]: X_test
```

```
[12]:      Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI   \
355             9      165             88             0         0  30.4
123             5      132             80             0         0  26.8
551             3       84             68            30        106  31.9
586             8      143             66             0         0  34.9
114             7      160             54            32        175  30.5
..          ...    ...          ...          ...    ...    ...
173             1       79             60            42         48  43.5
644             3      103             72            30        152  27.6
156             2       99             52            15         94  24.6
304             3      150             76             0         0  21.0
597             1       89             24            19         25  27.8

      DiabetesPedigreeFunction  Age
355                      0.302   49
123                      0.186   69
551                      0.591   25
586                      0.129   41
114                      0.588   39
..          ...    ...
173                      0.678   23
644                      0.730   27
156                      0.637   21
304                      0.207   37
597                      0.559   21

[231 rows x 8 columns]
```

```
[13]: y_test
```

```
[13]: 355      1
      123      0
      551      0
      586      1
      114      1
      ..
      173      0
      644      0
      156      0
      304      0
      597      0
Name: Outcome, Length: 231, dtype: int64
```

```
[14]: from sklearn.ensemble import RandomForestClassifier
```

```
[15]: rf_model = RandomForestClassifier().fit(X_train, y_train)
```

```
[16]: y_pred = rf_model.predict(X_test)
```

```
[17]: from sklearn.metrics import accuracy_score
```

```
[18]: accuracy_score(y_test, y_pred)
```

```
[18]: 0.7835497835497836
```