```
pip install pandas-profiling
```

```
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Collecting pandas-profiling
  Downloading pandas_profiling-3.6.6-py2.py3-none-any.whl (324 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 324.4/324.4 kB 11.5 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 345.9/345.9 kB 36.5 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (679 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 679.8/679.8 kB 52.6 MB/s eta
0:00:00
ent already satisfied: seaborn<0.13,>=0.10.1 in
/usr/local/lib/python3.9/dist-packages (from ydata-profiling->pandas-
profiling) (0.12.2)
Requirement already satisfied: pandas!=1.4.0,<1.6,>1.1 in
/usr/local/lib/python3.9/dist-packages (from ydata-profiling->pandas-
profiling) (1.5.3)
Collecting tqdm<4.65,>=4.48.2
  Downloading tqdm-4.64.1-py2.py3-none-any.whl (78 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 78.5/78.5 kB 9.7 MB/s eta
0:00:00
ent already satisfied: pydantic<1.11,>=1.8.1 in
/usr/local/lib/python3.9/dist-packages (from ydata-profiling->pandas-
profiling) (1.10.7)
Collecting imagehash==4.3.1
  Downloading ImageHash-4.3.1-py2.py3-none-any.whl (296 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 296.5/296.5 kB 36.3 MB/s eta
0:00:00
ent already satisfied: numpy<1.24,>=1.16.0 in
/usr/local/lib/python3.9/dist-packages (from ydata-profiling->pandas-
profiling) (1.22.4)
Requirement already satisfied: statsmodels<0.14,>=0.13.2 in
/usr/local/lib/python3.9/dist-packages (from ydata-profiling->pandas-
profiling) (0.13.5)
Requirement already satisfied: requests<2.29,>=2.24.0 in
/usr/local/lib/python3.9/dist-packages (from ydata-profiling->pandas-
profiling) (2.27.1)
Collecting typeguard<2.14,>=2.13.2
  Downloading typeguard-2.13.3-py3-none-any.whl (17 kB)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in
/usr/local/lib/python3.9/dist-packages (from ydata-profiling->pandas-
profiling) (3.1.2)
Collecting scipy<1.10,>=1.4.1
  Downloading scipy-1.9.3-cp39-cp39-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (33.8 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 33.8/33.8 MB 19.5 MB/s eta
0:00:00
age_path]==0.7.5
```

```
  Downloading visions-0.7.5-py3-none-any.whl (102 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 102.7/102.7 kB 12.4 MB/s eta
0:00:00
atplotlib<3.7,>=3.2
  Downloading matplotlib-3.6.3-cp39-cp39-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.8 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.8/11.8 MB 86.6 MB/s eta
0:00:00
ultimethod<1.10,>=1.4
  Downloading multimethod-1.9.1-py3-none-any.whl (10 kB)
Collecting htmlmin==0.1.12
  Downloading htmlmin-0.1.12.tar.gz (19 kB)
  Preparing metadata (setup.py) ... ent already satisfied:
PyYAML<6.1,>=5.0.0 in /usr/local/lib/python3.9/dist-packages (from
ydata-profiling->pandas-profiling) (6.0)
Requirement already satisfied: PyWavelets in
/usr/local/lib/python3.9/dist-packages (from imagehash==4.3.1->ydata-
profiling->pandas-profiling) (1.4.1)
Requirement already satisfied: pillow in
/usr/local/lib/python3.9/dist-packages (from imagehash==4.3.1->ydata-
profiling->pandas-profiling) (8.4.0)
Requirement already satisfied: networkx>=2.4 in
/usr/local/lib/python3.9/dist-packages (from
visions[type_image_path]==0.7.5->ydata-profiling->pandas-profiling)
(3.1)
Collecting tangled-up-in-unicode>=0.0.4
  Downloading tangled_up_in_unicode-0.2.0-py3-none-any.whl (4.7 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.7/4.7 MB 73.3 MB/s eta
0:00:00
ent already satisfied: attrs>=19.3.0 in /usr/local/lib/python3.9/dist-
packages (from visions[type_image_path]==0.7.5->ydata-profiling-
>pandas-profiling) (23.1.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.9/dist-packages (from jinja2<3.2,>=2.11.1-
>ydata-profiling->pandas-profiling) (2.1.2)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.9/dist-packages (from matplotlib<3.7,>=3.2-
>ydata-profiling->pandas-profiling) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in
/usr/local/lib/python3.9/dist-packages (from matplotlib<3.7,>=3.2-
>ydata-profiling->pandas-profiling) (3.0.9)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.9/dist-packages (from matplotlib<3.7,>=3.2-
>ydata-profiling->pandas-profiling) (4.39.3)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.9/dist-packages (from matplotlib<3.7,>=3.2-
>ydata-profiling->pandas-profiling) (0.11.0)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.9/dist-packages (from matplotlib<3.7,>=3.2-
>ydata-profiling->pandas-profiling) (23.1)
```

```
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.9/dist-packages (from matplotlib<3.7,>=3.2-
>ydata-profiling->pandas-profiling) (1.0.7)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.9/dist-packages (from matplotlib<3.7,>=3.2-
>ydata-profiling->pandas-profiling) (1.4.4)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.9/dist-packages (from pandas!=1.4.0,<1.6,>1.1-
>ydata-profiling->pandas-profiling) (2022.7.1)
Requirement already satisfied: joblib>=0.14.1 in
/usr/local/lib/python3.9/dist-packages (from phik<0.13,>=0.11.1-
>ydata-profiling->pandas-profiling) (1.2.0)
Requirement already satisfied: typing-extensions>=4.2.0 in
/usr/local/lib/python3.9/dist-packages (from pydantic<1.11,>=1.8.1-
>ydata-profiling->pandas-profiling) (4.5.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.9/dist-packages (from requests<2.29,>=2.24.0-
>ydata-profiling->pandas-profiling) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.9/dist-packages (from requests<2.29,>=2.24.0-
>ydata-profiling->pandas-profiling) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.9/dist-packages (from requests<2.29,>=2.24.0-
>ydata-profiling->pandas-profiling) (2022.12.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/usr/local/lib/python3.9/dist-packages (from requests<2.29,>=2.24.0-
>ydata-profiling->pandas-profiling) (2.0.12)
Requirement already satisfied: patsy>=0.5.2 in
/usr/local/lib/python3.9/dist-packages (from
statsmodels<0.14,>=0.13.2->ydata-profiling->pandas-profiling) (0.5.3)
Requirement already satisfied: six in /usr/local/lib/python3.9/dist-
packages (from patsy>=0.5.2->statsmodels<0.14,>=0.13.2->ydata-
profiling->pandas-profiling) (1.16.0)
Building wheels for collected packages: htmlmin
  Building wheel for htmlmin (setup.py) ... lmin: filename=htmlmin-
0.1.12-py3-none-any.whl size=27096
sha256=a94b02692dc5b1b0e3514871d9628470911ea1801a8a7fe66699ba0c2b5ea31
3
  Stored in directory:
/root/.cache/pip/wheels/1d/05/04/c6d7d3b66539d9e659ac6dfe81e2d0fd4c1a8
316cc5a403300
Successfully built htmlmin
Installing collected packages: htmlmin, typeguard, tqdm, tangled-up-
in-unicode, scipy, multimethod, matplotlib, imagehash, visions, phik,
ydata-profiling, pandas-profiling
  Attempting uninstall: tqdm
    Found existing installation: tqdm 4.65.0
    Uninstalling tqdm-4.65.0:
      Successfully uninstalled tqdm-4.65.0
  Attempting uninstall: scipy
```

```
    Found existing installation: scipy 1.10.1
    Uninstalling scipy-1.10.1:
      Successfully uninstalled scipy-1.10.1
  Attempting uninstall: matplotlib
    Found existing installation: matplotlib 3.7.1
    Uninstalling matplotlib-3.7.1:
      Successfully uninstalled matplotlib-3.7.1
Successfully installed htmlmin-0.1.12 imagehash-4.3.1 matplotlib-3.6.3
multimethod-1.9.1 pandas-profiling-3.6.6 phik-0.12.3 scipy-1.9.3
tangled-up-in-unicode-0.2.0 tqdm-4.64.1 typeguard-2.13.3 visions-0.7.5
ydata-profiling-4.1.2
```

{"pip_warning":{"packages":["matplotlib","mpl_toolkits"]}}

```python
import pandas_profiling
```

```
<ipython-input-2-6a00893fb3e1>:1: DeprecationWarning: `import
pandas_profiling` is going to be deprecated by April 1st. Please use
`import ydata_profiling` instead.
  import pandas_profiling
```

```python
import pandas as pd

import numpy as np

from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import Ridge, Lasso, RidgeCV, LassoCV,
ElasticNet, ElasticNetCV, LogisticRegression


from sklearn.model_selection import train_test_split

from statsmodels.stats.outliers_influence import
variance_inflation_factor

from sklearn.metrics import accuracy_score, confusion_matrix,
roc_curve, roc_auc_score

import matplotlib.pyplot as plt

import seaborn as sns

import pickle

df =
pd.read_csv("https://raw.githubusercontent.com/plotly/datasets/master/
diabetes.csv")

df
```

```
     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]
```

```python
from pandas_profiling import ProfileReport

ProfileReport(df)
```

{"model_id":"0568067ef96f434ea205af8ee9005eae","version_major":2,"version_minor":0}

{"model_id":"30d9d8d1a94846188895959eed53fb06","version_major":2,"version_minor":0}

{"model_id":"29eb2d9770de4d3bb97a247c09348dda","version_major":2,"version_minor":0}

```
<IPython.core.display.HTML object>
```

```
df['BMI'] = df['BMI'].replace(0 , df['BMI'].mean())
```

```
df.columns
```

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

```
df['BloodPressure'] = df['BloodPressure'].replace(0,
df['BloodPressure'].mean())
```

```
df['Glucose'] = df['Glucose'].replace(0, df['Glucose'].mean())
```

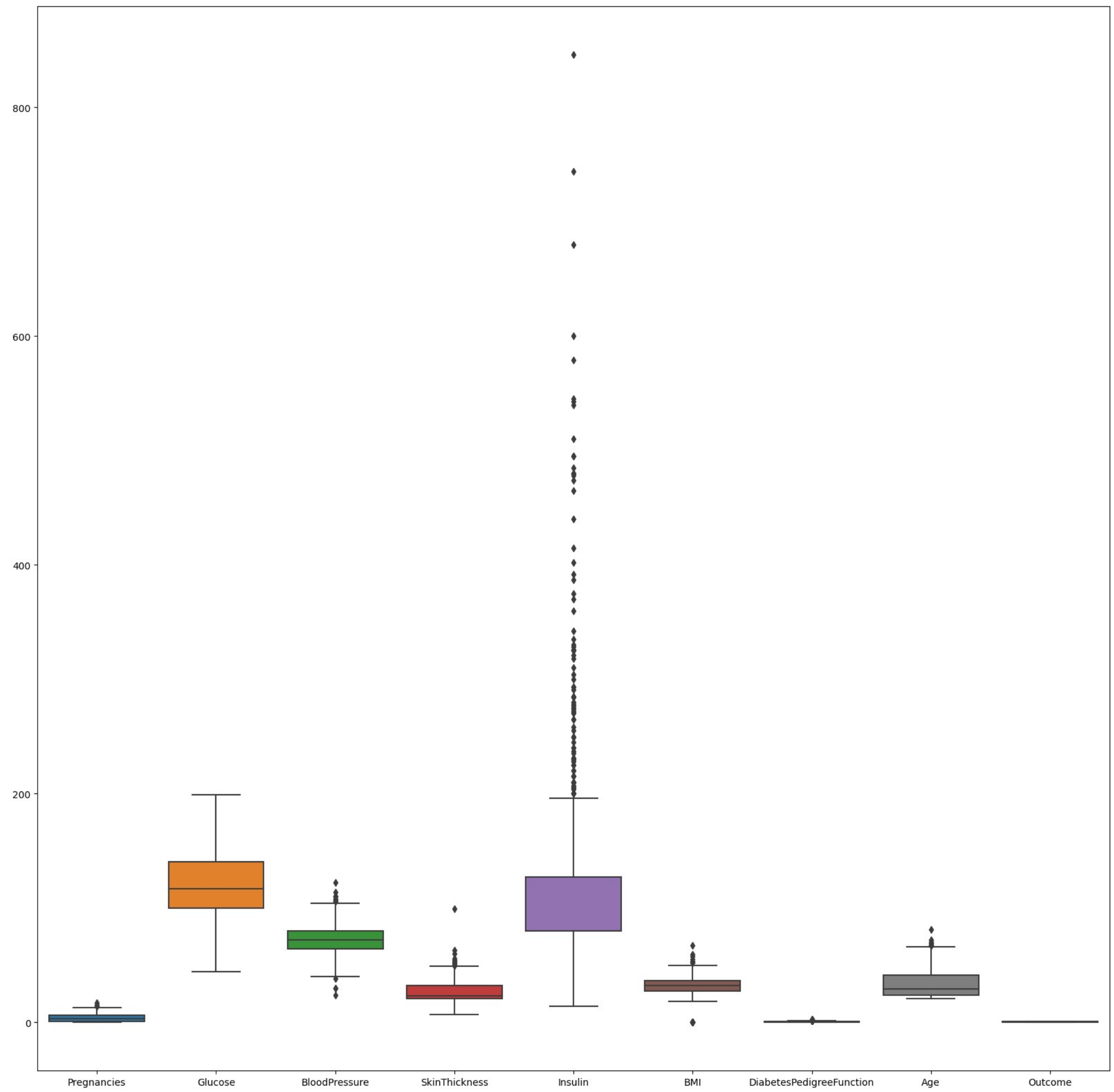
```
df['Insulin'] = df['Insulin'].replace(0, df['Insulin'].mean())
```

```
df['SkinThickness'] = df['SkinThickness'].replace(0,
df['SkinThickness'].mean())
```

Box Plot to show outliers

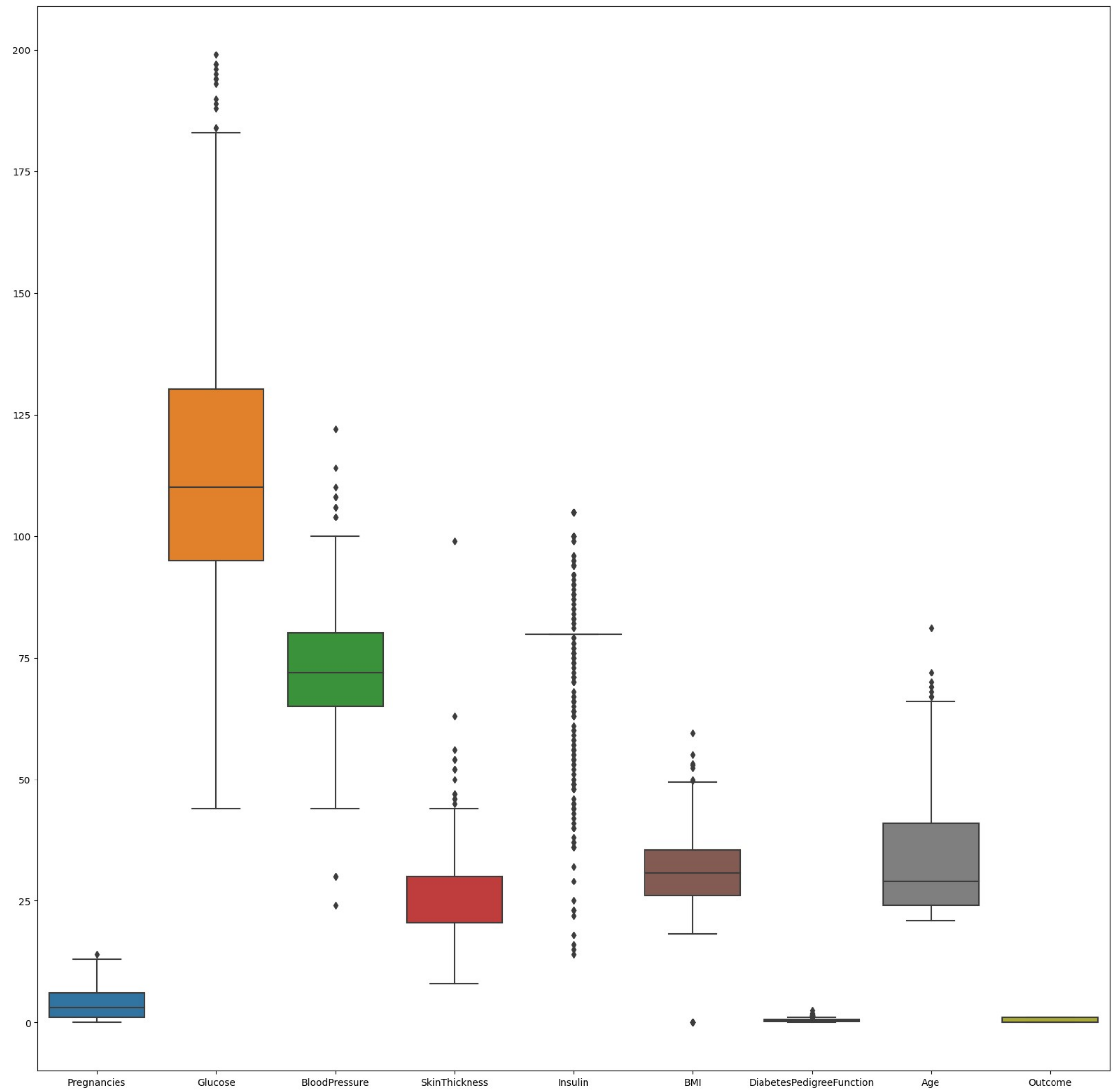
```
fig, ax = plt.subplots(figsize = (20, 20))
sns.boxplot(data = df, ax = ax)
```

```
<Axes: >
```




```
q = df['Insulin'].quantile(.70)
df_new = df[df['Insulin'] < q]

fig, ax = plt.subplots(figsize = (20, 20))
sns.boxplot(data = df_new, ax = ax)
```

<Axes: >

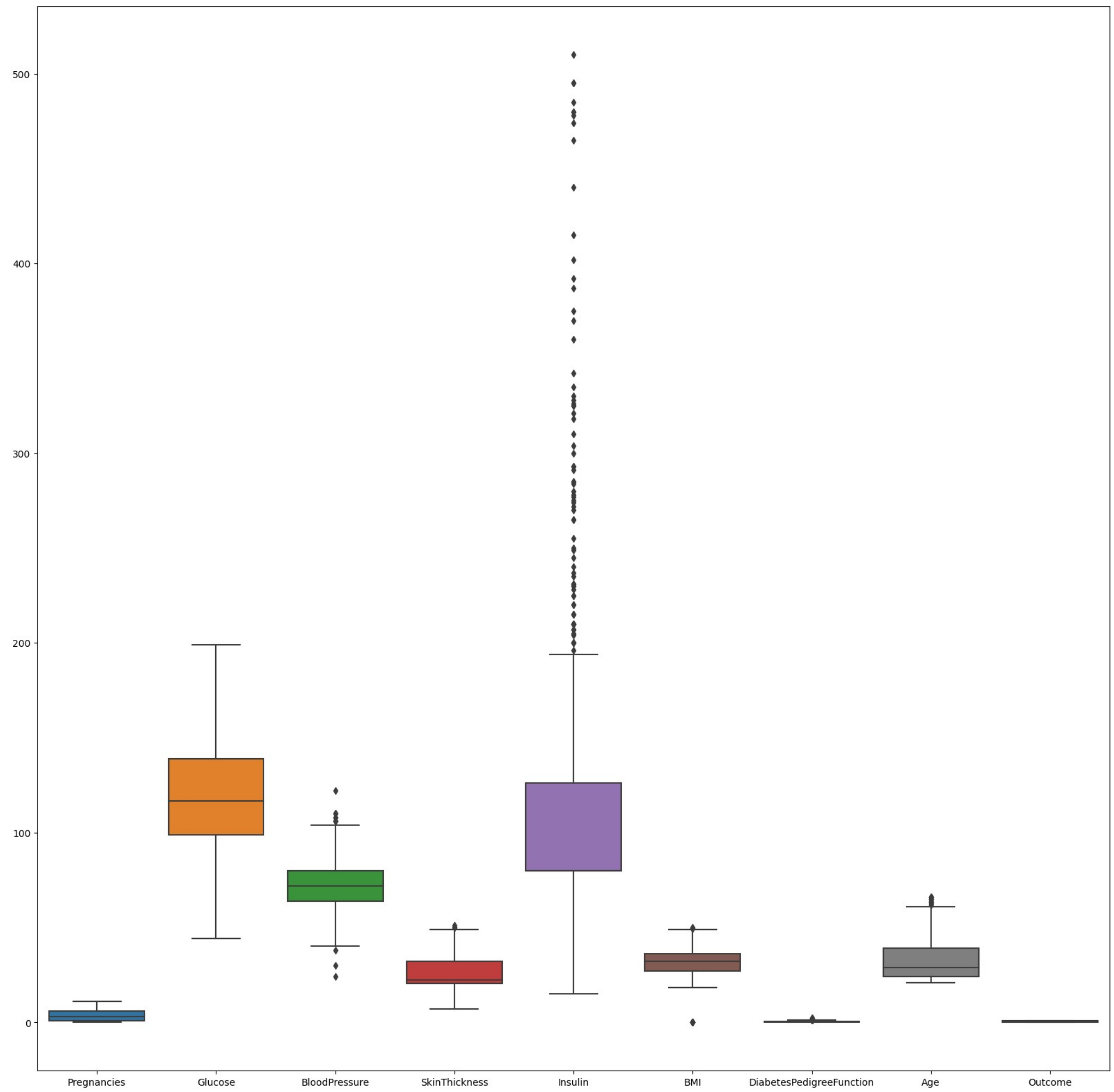



```
q = df['Pregnancies'].quantile(.98)
df_new = df[df['Pregnancies'] < q]

q = df['BMI'].quantile(.99)
df_new =  df_new[df_new['BMI'] < q]


q= df['SkinThickness'].quantile(.99)
df_new = df_new[df_new['SkinThickness'] < q]

q = df['Insulin'].quantile(.99)
df_new = df_new[df_new['Insulin'] < q]

q = df['Age'].quantile(.99)
df_new = df_new[df_new['Age'] < q]

df_new
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness     Insulin
BMI  \
0              6    148.0           72.0      35.000000   79.799479
33.6
1              1     85.0           66.0      29.000000   79.799479
26.6
2              8    183.0           64.0      20.536458   79.799479
23.3
3              1     89.0           66.0      23.000000   94.000000
28.1
4              0    137.0           40.0      35.000000  168.000000
43.1
..           ...      ...            ...            ...         ...
...
763           10    101.0           76.0      48.000000  180.000000
32.9
764            2    122.0           70.0      27.000000   79.799479
36.8
765            5    121.0           72.0      23.000000  112.000000
26.2
766            1    126.0           60.0      20.536458   79.799479
30.1
767            1     93.0           70.0      31.000000   79.799479
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

[716 rows x 9 columns]
```

```
fig, ax = plt.subplots(figsize = (20, 20))
sns.boxplot(data = df_new, ax = ax)
```

```
<Axes: >
```




```
ProfileReport(df_new)
```

{"model_id":"136761633ad04989af90297d87c5c8a2","version_major":2,"version_minor":0}

{"model_id":"a0bd6f0f2002480eab7aba9f415b3d32","version_major":2,"version_minor":0}

{"model_id":"b3c5b957590342aabb18be94efd07a29","version_major":2,"version_minor":0}

<IPython.core.display.HTML object>

```
df_new
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness     Insulin
BMI  \
```

```
0                6    148.0           72.0      35.000000   79.799479
33.6
1                1     85.0           66.0      29.000000   79.799479
26.6
2                8    183.0           64.0      20.536458   79.799479
23.3
3                1     89.0           66.0      23.000000   94.000000
28.1
4                0    137.0           40.0      35.000000  168.000000
43.1
..             ...      ...            ...            ...         ...
...
763             10    101.0           76.0      48.000000  180.000000
32.9
764              2    122.0           70.0      27.000000   79.799479
36.8
765              5    121.0           72.0      23.000000  112.000000
26.2
766              1    126.0           60.0      20.536458   79.799479
30.1
767              1     93.0           70.0      31.000000   79.799479
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

[716 rows x 9 columns]
```

```
y = df_new['Outcome']
```

```
y
```

```
0      1
1      0
2      1
3      0
4      1
      ..
763    0
764    0
765    0
```

```
766    1
767    0
Name: Outcome, Length: 716, dtype: int64
```

```
x = df_new.drop(columns=['Outcome'])
```

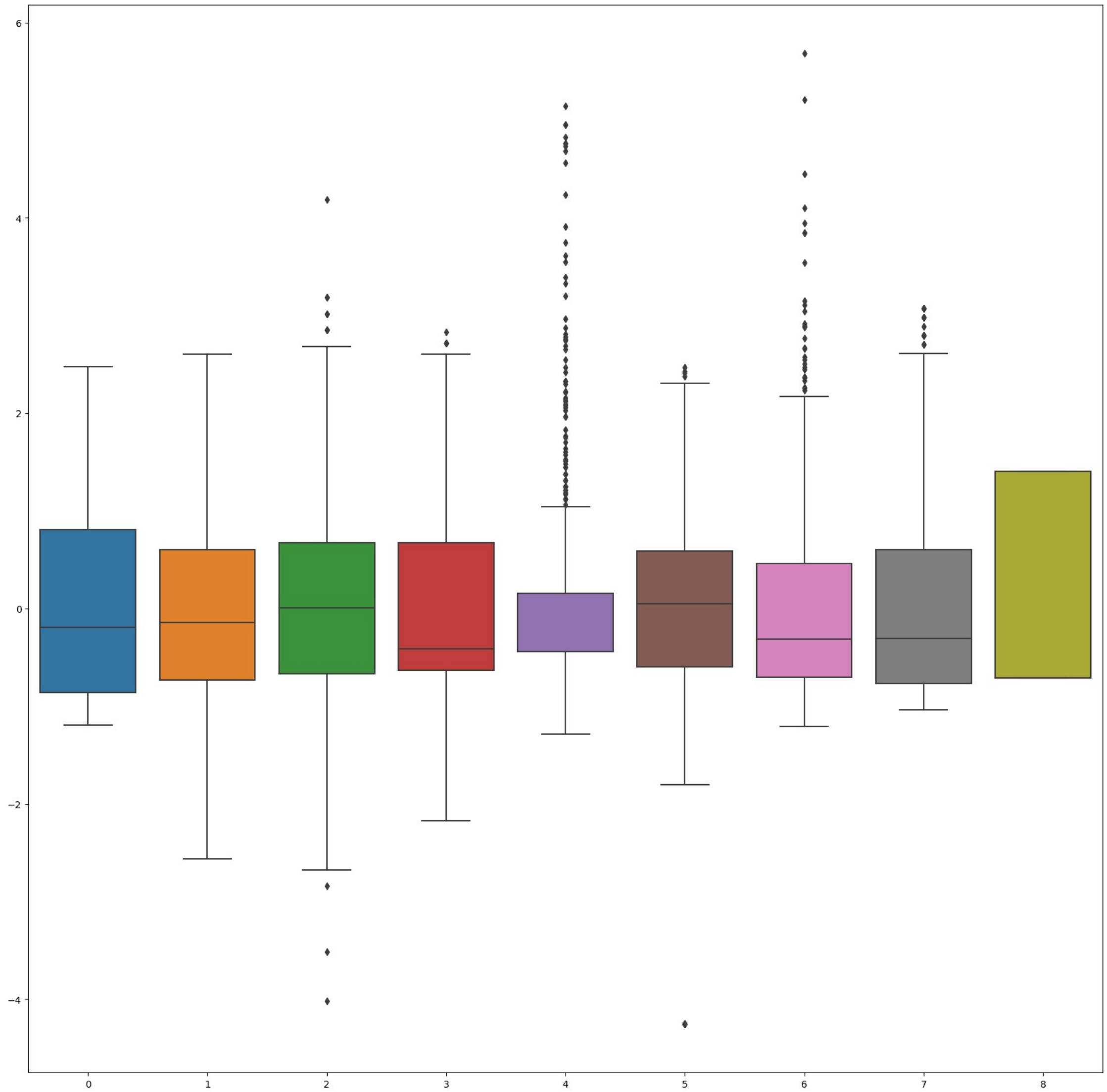
```
x
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness     Insulin
BMI  \
0              6    148.0           72.0      35.000000   79.799479
33.6
1              1     85.0           66.0      29.000000   79.799479
26.6
2              8    183.0           64.0      20.536458   79.799479
23.3
3              1     89.0           66.0      23.000000   94.000000
28.1
4              0    137.0           40.0      35.000000  168.000000
43.1
..           ...      ...            ...            ...         ...
...
763           10    101.0           76.0      48.000000  180.000000
32.9
764            2    122.0           70.0      27.000000   79.799479
36.8
765            5    121.0           72.0      23.000000  112.000000
26.2
766            1    126.0           60.0      20.536458   79.799479
30.1
767            1     93.0           70.0      31.000000   79.799479
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

[716 rows x 8 columns]
```

```
scaler = StandardScaler()
ProfileReport(pd.DataFrame(scaler.fit_transform(x)))
x_scaled = scaler.fit_transform(x)
```

```
df_new_scaler = pd.DataFrame(scaler.fit_transform(df_new))
fig, ax = plt.subplots(figsize = (20, 20))
sns.boxplot(data = df_new_scaler, ax = ax)
```

```
<Axes: >
```



Dataset is dispersed equally

```
x_scaled
```

```
array([[ 0.80822764,  0.9059344 ,  0.003933  , ...,  0.26388706,
         0.5060621 ,  1.60917704],
       [-0.86043151, -1.19643634, -0.49855498, ..., -0.67766768,
        -0.35435232, -0.12615744],
       [ 1.4756913 ,  2.07391814, -0.66605097, ..., -1.12154349,
         0.64634705, -0.03482405],
       ...,
       [ 0.47449581,  0.00491836,  0.003933  , ..., -0.73147081,
```

```
        -0.68480133, -0.21749083],
       [-0.86043151,  0.17177319, -1.00104295, ..., -0.20689031,
        -0.3605872 ,  1.33517686],
       [-0.86043151, -0.92946863, -0.16356299, ..., -0.16653796,
        -0.46658028, -0.85682459]])
```

```
y
```

```
0      1
1      0
2      1
3      0
4      1
      ..
763    0
764    0
765    0
766    1
767    0
Name: Outcome, Length: 716, dtype: int64
```

Checking Colinearity

```
def vif_score(x):
  scaler = StandardScaler()
  arr = scaler.fit_transform(x)
  return pd.DataFrame([[x.columns[i],
variance_inflation_factor(arr,i)] for i in range(arr.shape[1])],
columns = ["FEATURE", "VIF_SCORE"])
```

```
vif_score(x)
```

```
                    FEATURE  VIF_SCORE
0               Pregnancies   1.474617
1                   Glucose   1.363812
2             BloodPressure   1.226262
3             SkinThickness   1.420421
4                   Insulin   1.305182
5                       BMI   1.438581
6  DiabetesPedigreeFunction   1.041569
7                       Age   1.681447
```

```
x_train, x_test, y_train, y_test = train_test_split(x_scaled, y,
test_size = .20, random_state = 144)
```

```
x_train
```

```
array([[-0.19296785,  1.64009561, -0.66605097, ...,  0.3849441 ,
        -0.3387651 , -0.21749083],
       [-1.19416334,  1.33975693, -1.83852292, ..., -1.30985444,
        -0.65674434,  2.97917794],
       [ 1.14195947,  2.44099875, -0.33105899, ...,  0.57325505,
```

```
         0.87392043,  0.7871765 ],
       ...,
       [-0.86043151, -1.39666213,  0.67391697, ..., -0.83907707,
         0.36889458, -0.94815798],
       [ 0.14076398,  0.37199897,  1.17640494, ..., -0.48935673,
        -0.14236616,  2.79651116],
       [ 0.47449581,  0.77245054,  0.84141296, ...,  0.04867455,
        -0.03949052,  2.33984419]])
```

```
x_test
```

```
array([[ 0.80822764,  0.77245054,  0.003933  , ...,  0.30423941,
        -0.65362689,  0.6958431 ],
       [ 0.47449581, -1.09632345, -0.49855498, ..., -0.97358489,
        -0.38240931, -0.21749083],
       [-0.86043151, -1.22980731, -0.66605097, ...,  0.70776287,
         0.01974091, -0.40015762],
       ...,
       [-1.19416334, -0.32879128, -0.58230297, ..., -0.94668332,
         0.60893773, -0.12615744],
       [-1.19416334, -0.09519453,  1.00890895, ...,  1.90488247,
         0.26913639, -0.12615744],
       [ 0.47449581, -0.16193646,  0.17142899, ...,  0.08902689,
         0.60893773,  0.23917614]])
```

```
y_train
```

```
132    1
294    0
185    1
643    0
634    0
      ..
693    1
456    0
232    0
479    0
388    1
Name: Outcome, Length: 572, dtype: int64
```

```
y_test
```

```
95     0
52     0
554    0
473    0
690    0
      ..
214    1
474    0
529    0
```

```
16     1
386    1
Name: Outcome, Length: 144, dtype: int64
```

```
x_test[0]
```

```
array([ 0.80822764,  0.77245054,  0.003933  ,  0.10259685,
1.48329699,
        0.30423941, -0.65362689,  0.6958431 ])
```

```
logr_liblinear = LogisticRegression(verbose = 1, solver='liblinear')
```

```
logr.fit(x_train, y_train)
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1
concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s finished
```

```
LogisticRegression(verbose=1)
```

```
logr.predict([x_test[1]])
```

```
array([0])
```

```
logr.predict_proba([x_test[1]])
```

```
array([[0.92571637, 0.07428363]])
```

```
logr.predict_log_proba([x_test[1]])
```

```
array([[-0.07718739, -2.59986461]])
```

```
type(y_test)
```

```
pandas.core.series.Series
```

```
y_test.iloc[0]
```

```
0
```

```
y_test
```

```
95     0
52     0
554    0
473    0
690    0
      ..
214    1
474    0
529    0
16     1
386    1
Name: Outcome, Length: 144, dtype: int64
```

```
logr = LogisticRegression(verbose = 1)
```

```
logr.fit(x_train, y_train)
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1
concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s finished
```

```
LogisticRegression(verbose=1)
```

```
logr
```

```
LogisticRegression(verbose=1)
```

```
logr_liblinear
```

```
LogisticRegression(solver='liblinear', verbose=1)
```

```
logr_liblinear.fit(x_train, y_train)
```

```
[LibLinear]
```

```
LogisticRegression(solver='liblinear', verbose=1)
```

```
y_pred_liblinear = logr_liblinear.predict(x_test)
y_pred_liblinear
```

```
array([1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0,
1,
       1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,
0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
1,
       1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0,
0,
       0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0])
```

```
y_pred_default = logr.predict(x_test)
```

```
y_pred_default
```

```
array([1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0,
1,
       1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,
0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
0,
```

```
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
1,
       1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0,
0,
       0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0])
```

```
confusion_matrix(y_test, y_pred_liblinear)
```

```
array([[87,  7],
       [19, 31]])
```

```
confusion_matrix(y_test, y_pred_default)
```

```
array([[87,  7],
       [19, 31]])
```

```
auc = roc_auc_score(y_test, y_pred_liblinear)
```

```
roc_auc_score(y_test, y_pred_default)
```

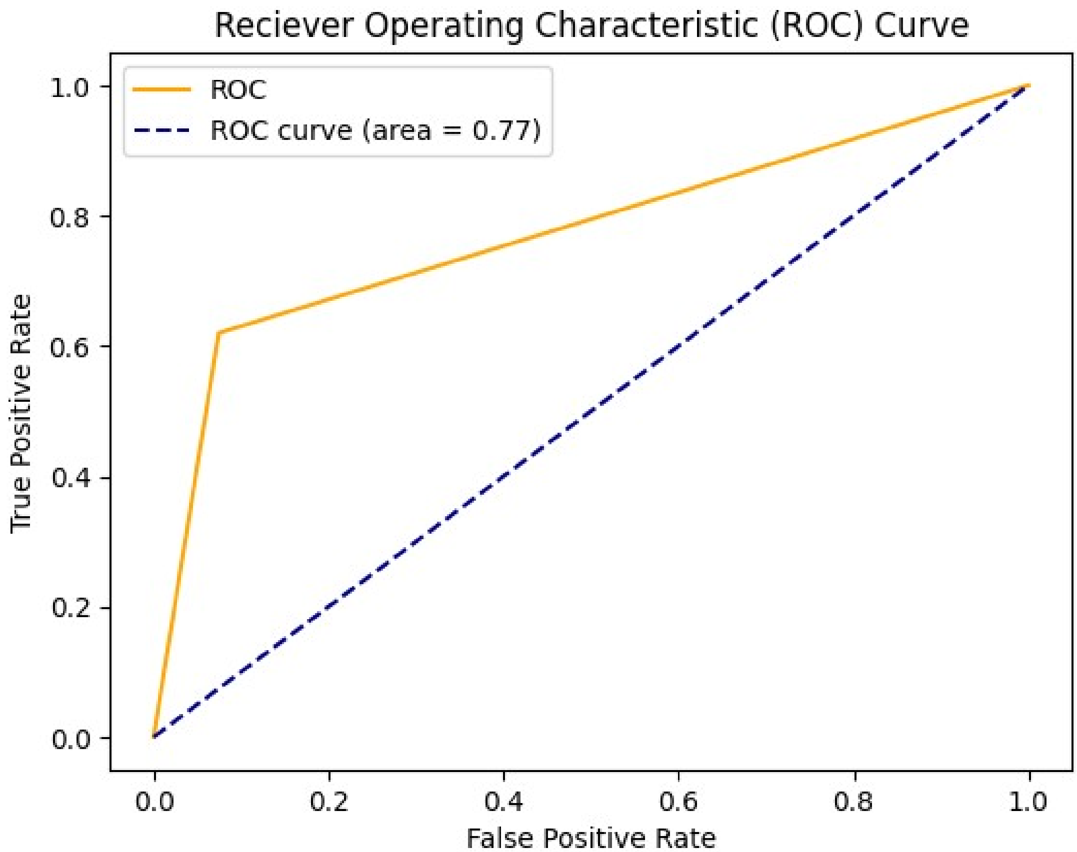
```
0.7727659574468084
```

```
fpr, tpr, thresholds =   roc_curve(y_test, y_pred_liblinear)
```

```
  roc_curve(y_test, y_pred_liblinear)
```

```
(array([0.        , 0.07446809, 1.        ]),
 array([0.  , 0.62, 1.  ]),
 array([2, 1, 0]))
```

```
plt.plot(fpr, tpr, color = 'orange', label='ROC')
plt.plot([0, 1], [0,1], color='darkblue', linestyle='--', label='ROC
curve (area = %0.2f)' % auc)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Reciever Operating Characteristic (ROC) Curve')
plt.legend()
plt.show()
```


Reciever Operating Characteristic (ROC) Curve
ROC
ROC curve (area = 0.77)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0