**This is PYTHON Code for Random Forest Tree**

**(need install pandas, numpy, seaborn, matplotlib, bib and sklearn)**

import pandas as pd

import numpy as np

dataset =pd.read\_csv("H:\data1.csv")

dataset.head()

**Output**

Gender AGE Urea Cr HbA1c Chol TG HDL LDL VLDL BMI CLASS

0 0 50 4.8 62 5.9 5.3 1.3 1.0 3.7 0.6 19.0 Diabetes\_1

1 0 50 4.8 62 5.9 5.3 1.3 1.0 3.7 0.6 19.0 Diabetes\_1

2 0 77 5.0 106 5.4 0.0 2.8 0.8 1.8 0.7 19.0 Diabetes\_0

3 1 33 2.7 47 6.0 4.2 1.4 1.3 2.6 0.7 19.0 Diabetes\_1

4 0 30 6.0 97 5.8 4.2 1.7 1.2 2.2 0.8 19.0 Diabetes\_1

>>> dataset.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1000 entries, 0 to 999

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Gender 1000 non-null int64

1 AGE 1000 non-null int64

2 Urea 1000 non-null float64

3 Cr 1000 non-null int64

4 HbA1c 1000 non-null float64

5 Chol 1000 non-null float64

6 TG 1000 non-null float64

7 HDL 1000 non-null float64

8 LDL 1000 non-null float64

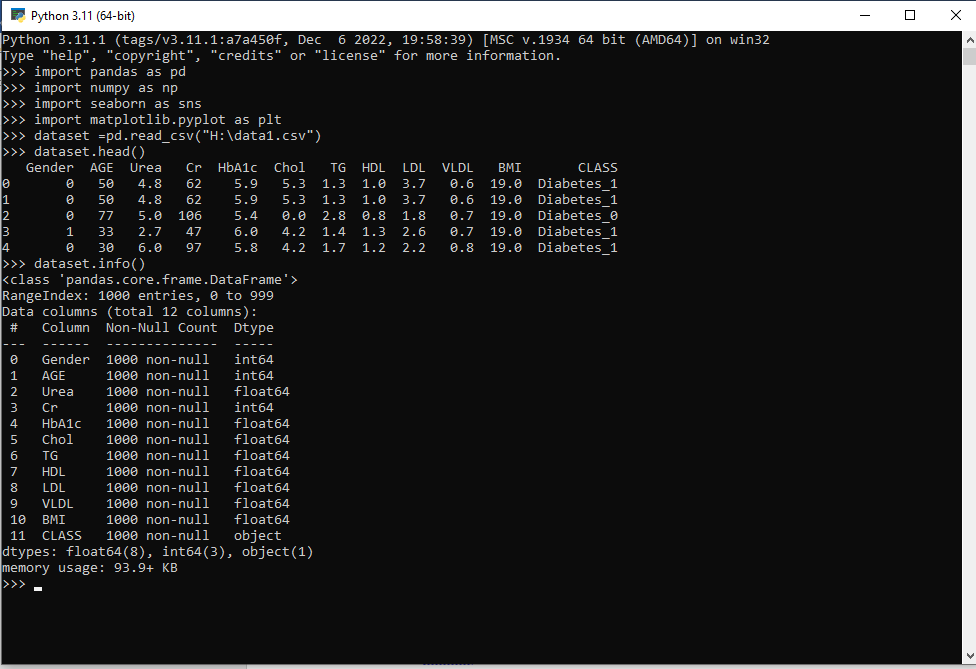
9 VLDL 1000 non-null float64

10 BMI 1000 non-null float64

11 CLASS 1000 non-null object

dtypes: float64(8), int64(3), object(1)

memory usage: 93.9+ KB



import seaborn as sns

dataset.info()

dataset.describe().T # T transposes the table

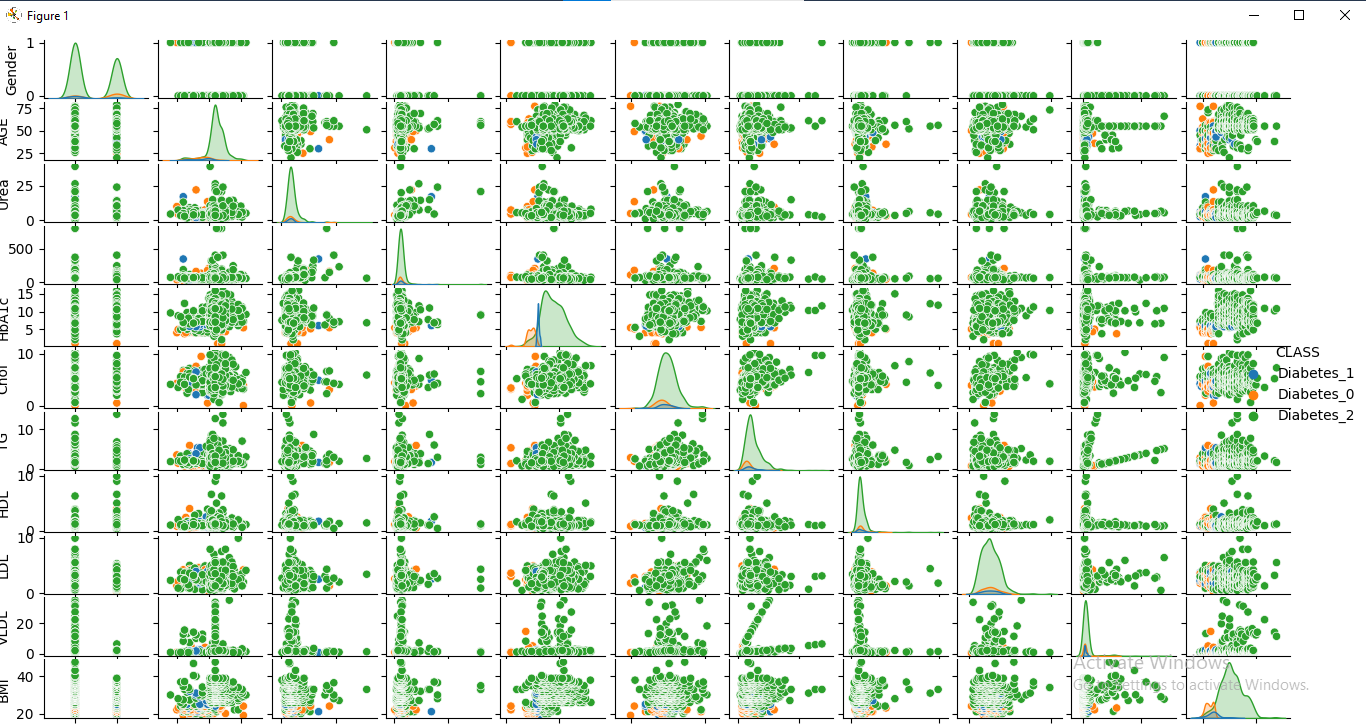
import matplotlib.pyplot as plt

g = sns.pairplot(dataset, hue='CLASS')

g.fig.suptitle("Scatterplot and histogram of pairs of variables ", fontsize = 14, y=1.05);

plt.show()

**OUTPUT**



dataset['CLASS'].unique()

**OUTPUT (to convert names into numbers)**

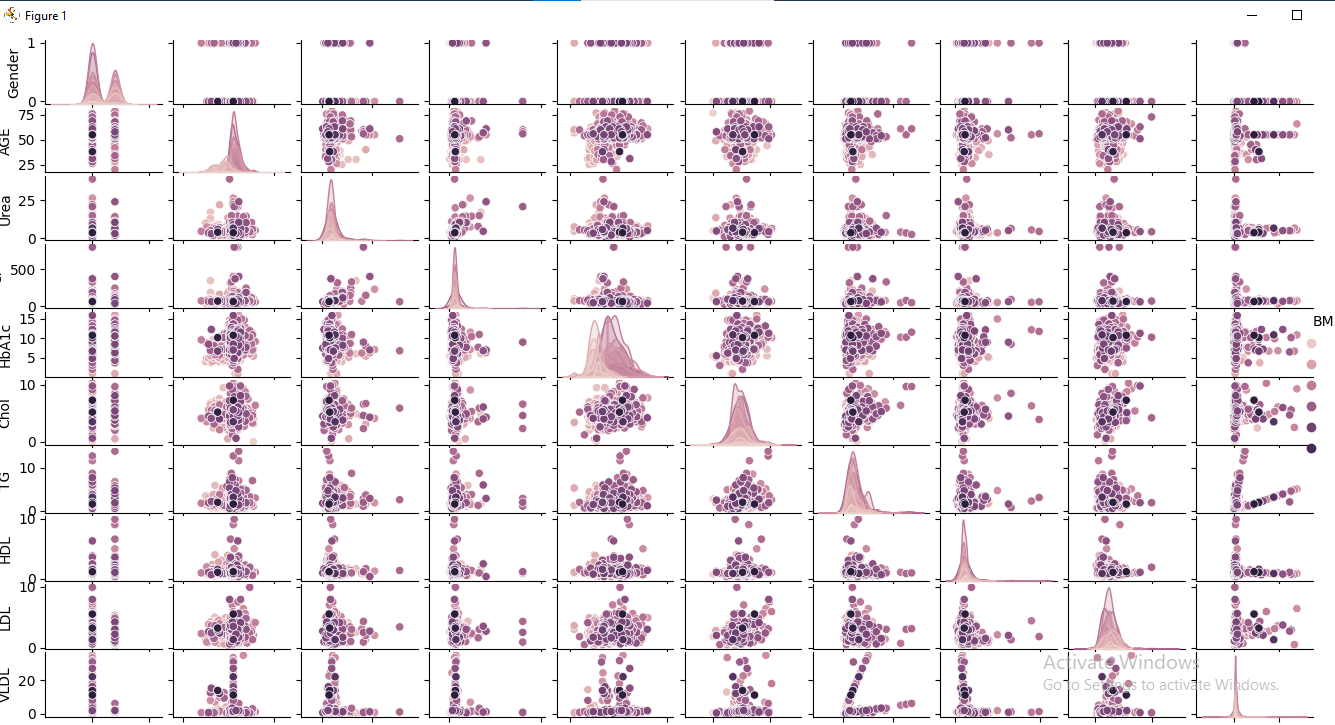
array(['Diabetes\_1', 'Diabetes\_0', 'Diabetes\_2'], dtype=object)

g = sns.pairplot(dataset, hue='BMI')

g.fig.suptitle("Scatterplot and histogram of pairs of variables ", fontsize = 14, y=1.05);

plt.show()

**OUTPUT**

****

**(replacing strings with numbers)**

dataset[' CLASS '] = dataset[' CLASS '].replace(' Diabetes\_1 ', 0);

dataset[' CLASS '] = dataset[' CLASS '].replace('Diabetes\_0', 1);

dataset[' CLASS '] = dataset[' CLASS '].replace('Diabetes\_2', 2);

y = dataset['CLASS']

X = dataset.drop(['CLASS'], axis=1)

**(Splitting data into training and testing)**

from sklearn.model\_selection import train\_test\_split

SEED = 42

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=SEED)

**(selecting 20% of data to test)**

from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier(n\_estimators=3, max\_depth=2, random\_state=SEED)

# Fit RandomForestClassifier

rfc.fit(X\_train, y\_train)

# Predict the test set labels

**(calculating the accuracy of Diabetic Dataset)**

y\_pred = rfc.predict(X\_test)

**(Calculate the Confusion matrix)**

from sklearn.metrics import classification\_report, confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

sns.heatmap(cm, annot=True, fmt='d').set\_title(The confusion matrix (0 = Diabetes\_1, 1 = Diabetes\_0, 2 = Diabetes\_2)')

print(classification\_report(y\_test,y\_pred))

features\_df = pd.DataFrame({'features': rfc.feature\_names\_in\_, 'importances': rfc.feature\_importances\_ })

To cite this please refer to

**Sahar J. Mohammed, Ali S. Ahmed and Mohammed S. Mohammed, “Feature Minimization for Diabetic Disorders High Performances prediction system based on Random Forest Tree”. 2023**