EC9630 Machine Learning - laboratory 01

Task: STASTICAL PATTEN CLASSIFICATION

Name: LIYANAGE L.D.T.N.

RegNo: 2020/E/082

Date: 28th April 2024

Time:

Question 02

(a). The Diabetes Health Indicators Dataset contains healthcare statistics and lifestyle survey information about people in general along with their diagnosis of diabetes. The 35 features consist of some demographics, lab test results, and answers to survey questions for each patient. The target variable for classification is whether a patient has diabetes, is pre-diabetic, or healthy. Dataset Characteristics Tabular, Multivariate

---

Subject Area: Health and Medicine

Associated Tasks: Classification

Feature Type: Categorical, Integer

Instances: 253680

Features: 21

(b). Categorical and Integer

(c). Features: The dataset includes various features like demographics, health history, and personal information that are used to predict the target variable, which is the diabetes status of the individual.

Labels: The target variable in the dataset is the diabetes diagnosis, categorized into three classes: Diabetes, Pre-diabetes, and Healthy. This variable is used for classification purposes to predict the health status of individuals based on the provided features.

(d). 21 Features.

```
import pandas as pd

# Load the dataset
df = pd.read_csv("/content/drive/MyDrive/CSV FILE/diabetes_012_health_indicators_BRF
df.describe()
```

	Diabetes_012	HighBP	HighChol	CholCheck	BMI	
count	253680.000000	253680.000000	253680.000000	253680.000000	253680.000000	2536
mean	0.296921	0.429001	0.424121	0.962670	28.382364	
std	0.698160	0.494934	0.494210	0.189571	6.608694	
min	0.000000	0.000000	0.000000	0.000000	12.000000	
25%	0.000000	0.000000	0.000000	1.000000	24.000000	
50%	0.000000	0.000000	0.000000	1.000000	27.000000	
75%	0.000000	1.000000	1.000000	1.000000	31.000000	
max	2.000000	1.000000	1.000000	1.000000	98.000000	

8 rows × 22 columns

```
# the Variable type informationS
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 253680 entries, 0 to 253679
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   Diabetes_012          253680 non-null  float64
 1   HighBP                253680 non-null  float64
 2   HighChol              253680 non-null  float64
 3   CholCheck             253680 non-null  float64
 4   BMI                   253680 non-null  float64
 5   Smoker                253680 non-null  float64
 6   Stroke                253680 non-null  float64
 7   HeartDiseaseorAttack  253680 non-null  float64
 8   PhysActivity          253680 non-null  float64
 9   Fruits                253680 non-null  float64
 10  Veggies               253680 non-null  float64
 11  HvyAlcoholConsump     253680 non-null  float64
 12  AnyHealthcare         253680 non-null  float64
 13  NoDocbcCost           253680 non-null  float64
 14  GenHlth               253680 non-null  float64
 15  MentHlth              253680 non-null  float64
 16  PhysHlth              253680 non-null  float64
 17  DiffWalk              253680 non-null  float64
 18  Sex                   253680 non-null  float64
 19  Age                   253680 non-null  float64
 20  Education             253680 non-null  float64
 21  Income                253680 non-null  float64
dtypes: float64(22)
memory usage: 42.6 MB
None
```

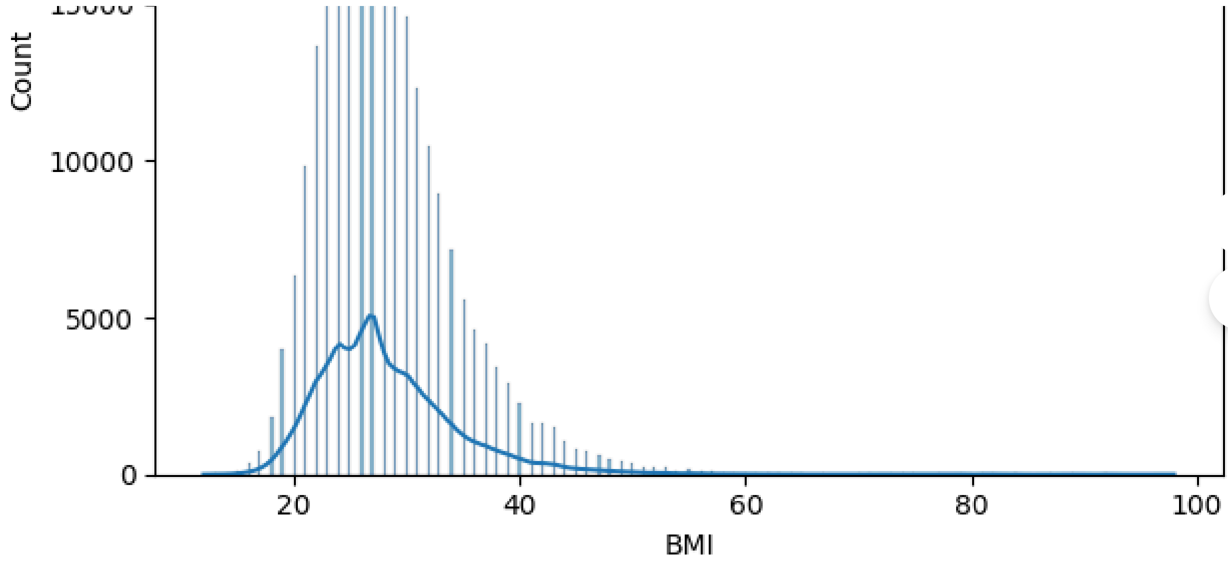
```
# Identify features (all columns except the label column)
features = df.drop(columns=['Diabetes_012'])

labels = df['Diabetes_012']

# Display the list of features
print("Features:")
print(features.columns.tolist())
print("\nLabel:")
print(labels.name)
```

```
Features:
['HighBP', 'HighChol', 'CholCheck', 'BMI', 'Smoker', 'Stroke', 'HeartDiseaseorAttack

Label:
Diabetes_012
```

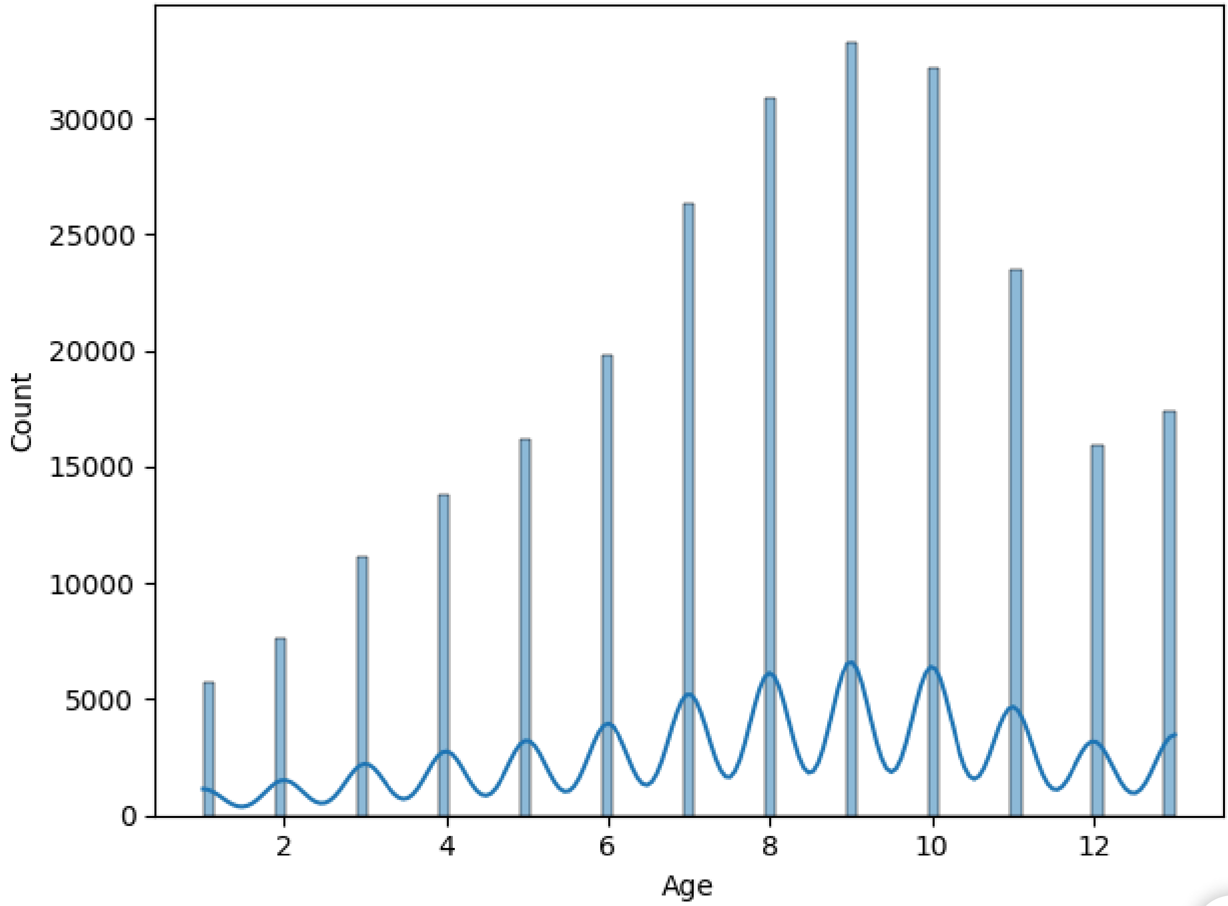
```
import matplotlib.pyplot as plt
import seaborn as sns


# Print the column names
print("Column names:")
print(df.columns)
```

```
Column names:
Index(['Diabetes_012', 'HighBP', 'HighChol', 'CholCheck', 'BMI', 'Smoker',
       'Stroke', 'HeartDiseaseorAttack', 'PhysActivity', 'Fruits', 'Veggies',
       'HvyAlcoholConsump', 'AnyHealthcare', 'NoDocbcCost', 'GenHlth',
       'MentHlth', 'PhysHlth', 'DiffWalk', 'Sex', 'Age', 'Education',
       'Income'],
      dtype='object')
```

```
#  missing values
print("\nMissing values per column:")
print(df.isnull().sum())

# Distributions-Numerical feature
print("\nNumerical feature distributions:")
for col in ['BMI', 'Age', 'Fruits', 'Veggies', 'PhysActivity']:
    print(f"\n{col}:")
    print(df[col].describe())
    plt.figure()
    sns.histplot(data=df, x=col, kde=True)
    plt.tight_layout()
    plt.show()

# Distributions-Categorical feature
print("\nCategorical feature distributions:")
for col in ['Sex', 'HighBP', 'HighChol', 'Smoker']:
    print(f"\n{col}:")
    print(df[col].value_counts())
    plt.figure()
    sns.countplot(data=df, x=col)
    plt.tight_layout()
    plt.show()
```

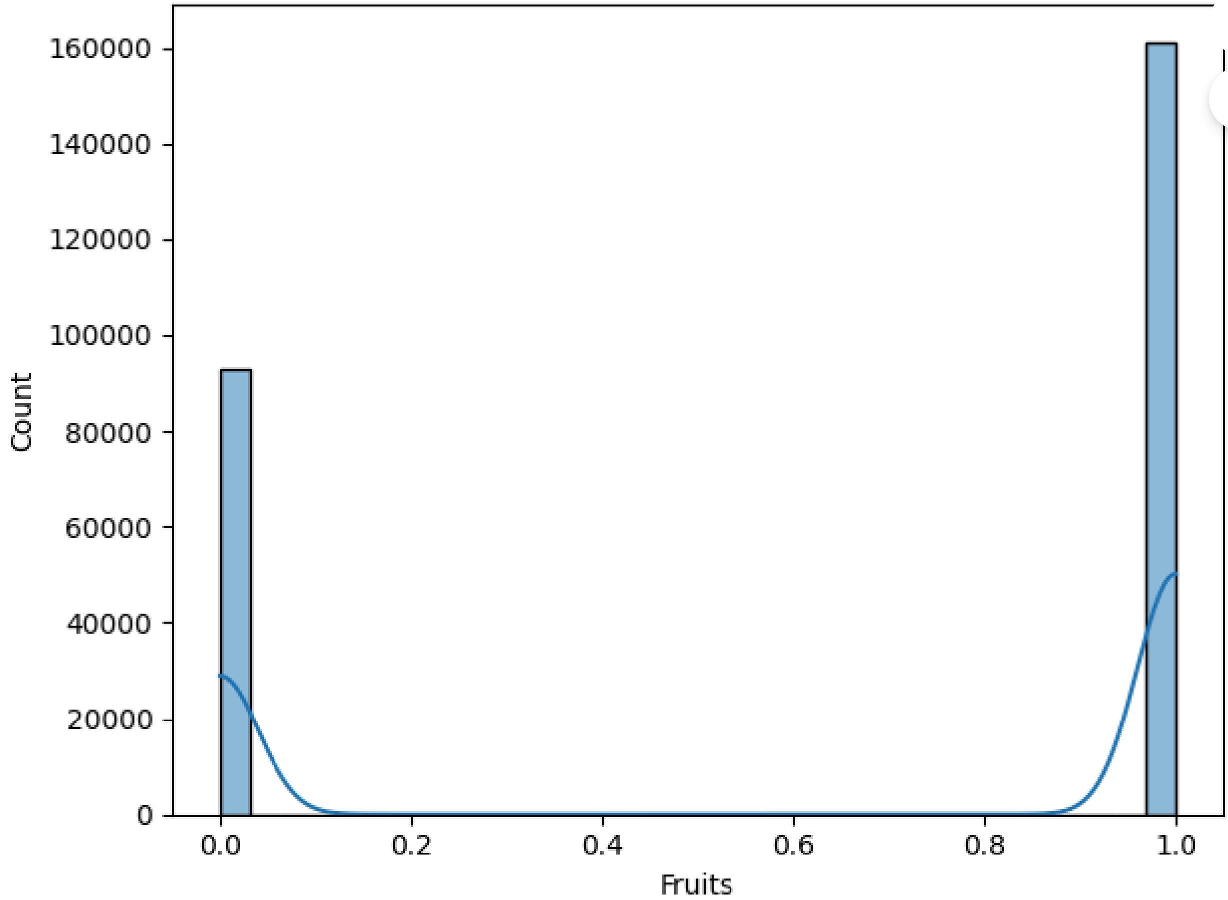



```
Age:
count    253680.000000
mean          8.032119
std           3.054220
min           1.000000
25%           6.000000
50%           8.000000
75%          10.000000
max          13.000000
Name: Age, dtype: float64
```

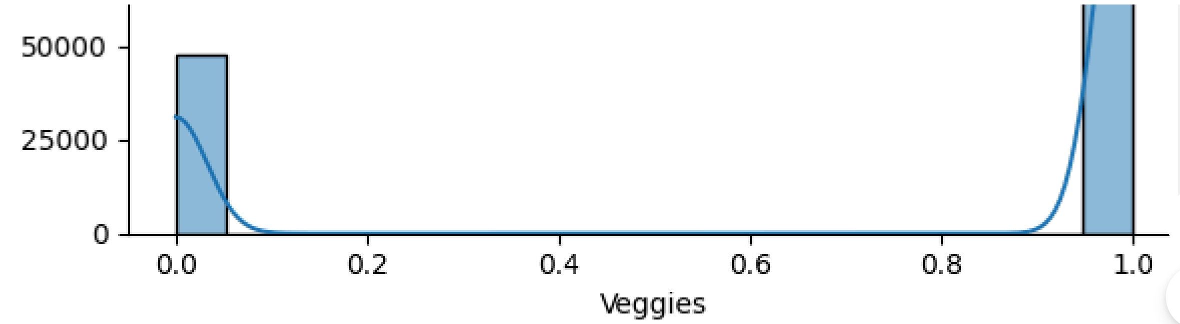



```
Fruits:
count    253680.000000
mean          0.634256
std           0.481639
min           0.000000
```

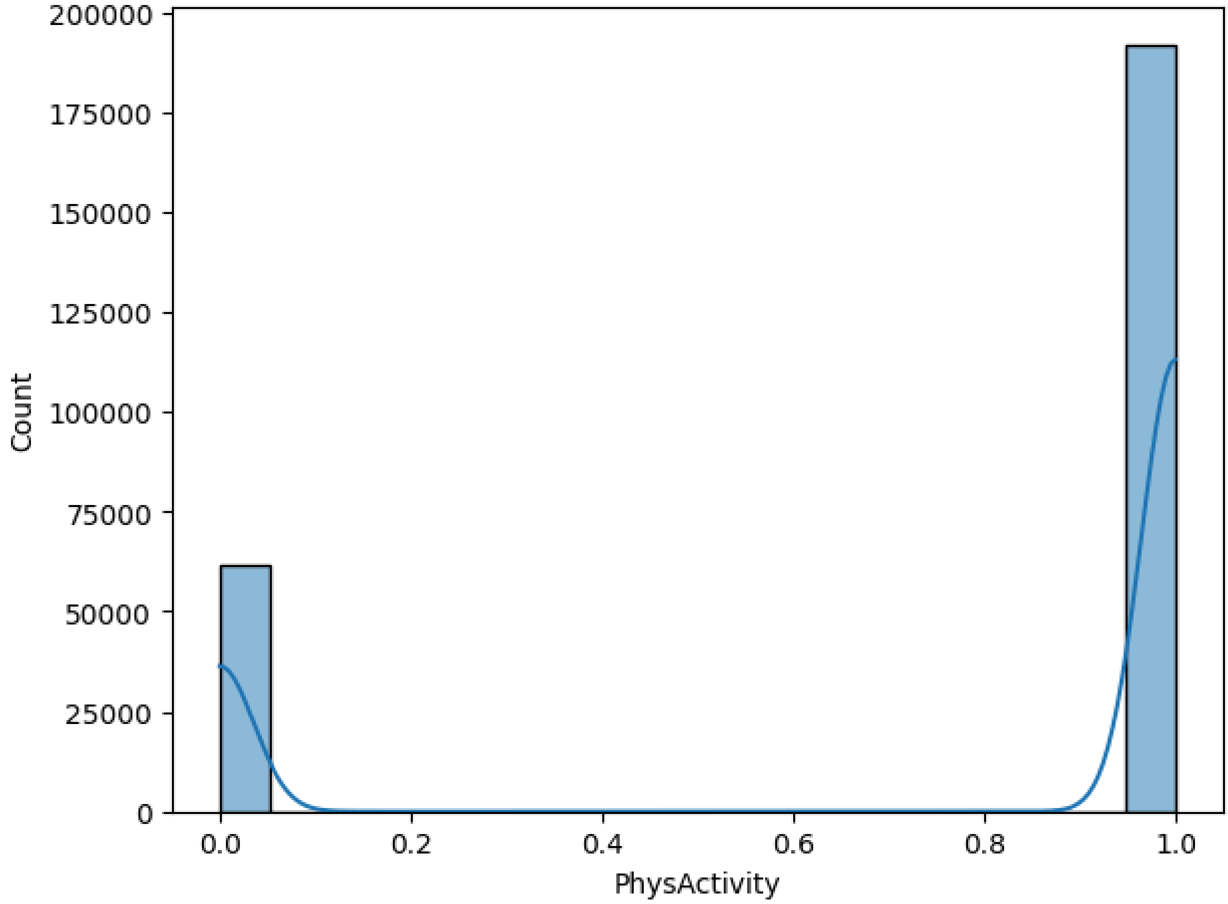
```
25%              0.000000
50%              1.000000
75%              1.000000
max              1.000000
Name: Fruits, dtype: float64
```



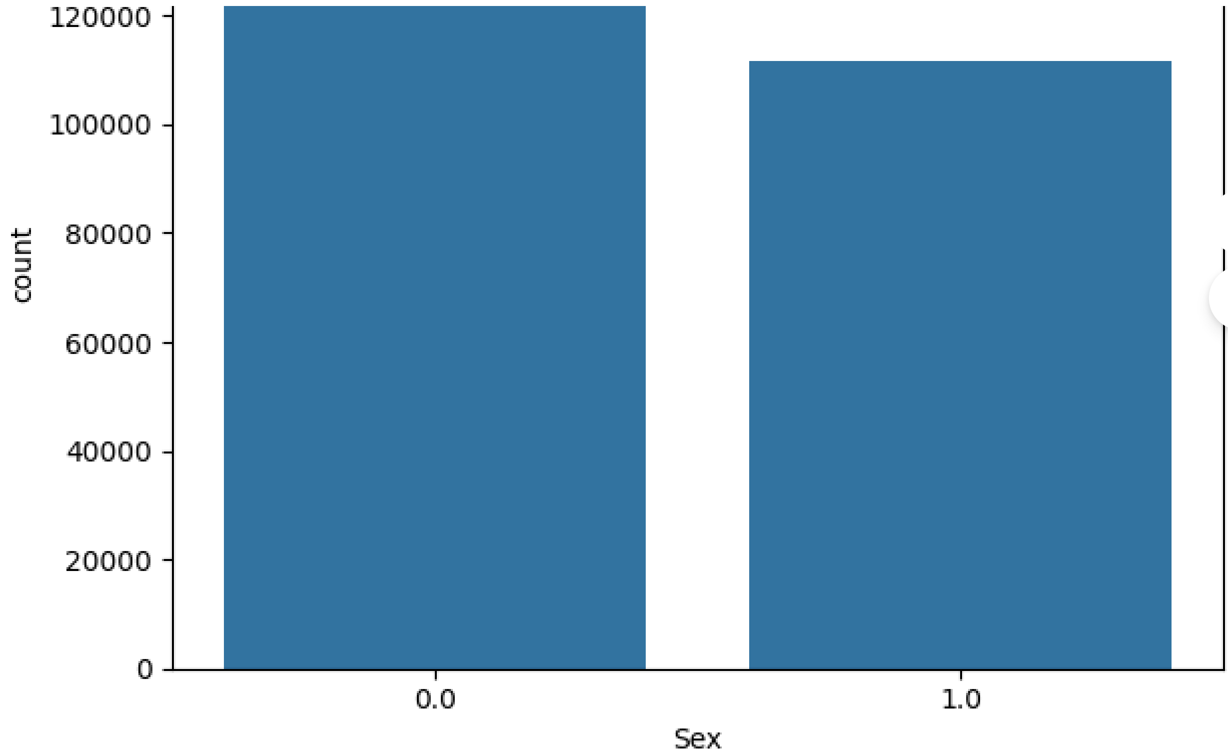

```
Veggies:
count    253680.000000
mean          0.811420
std           0.391175
min           0.000000
25%           1.000000
50%           1.000000
75%           1.000000
max           1.000000
Name: Veggies, dtype: float64
```



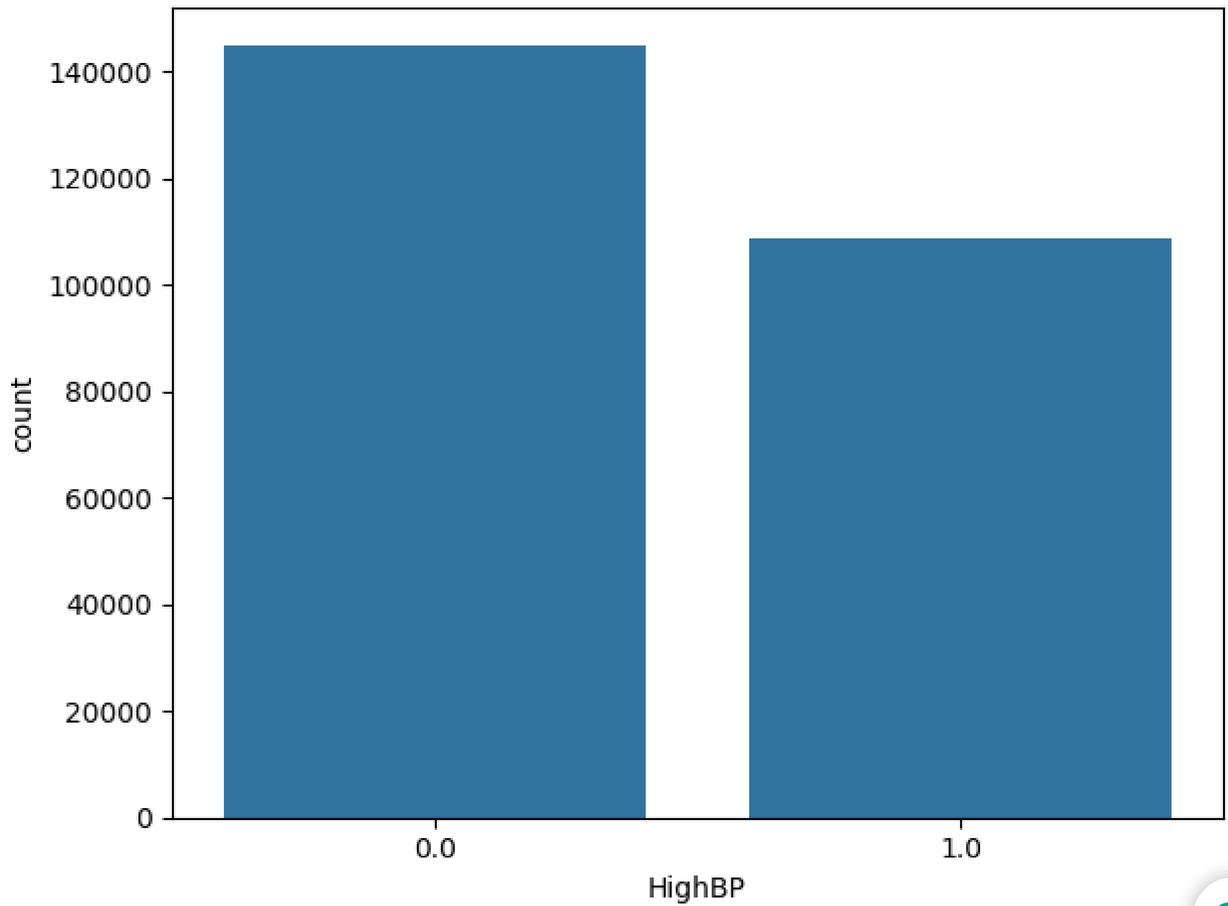

```
PhysActivity:
count    253680.000000
mean          0.756544
std           0.429169
min           0.000000
25%           1.000000
50%           1.000000
75%           1.000000
max           1.000000
Name: PhysActivity, dtype: float64
```




```
Categorical feature distributions:

Sex:
Sex
0.0    141974
1.0    111706
Name: count, dtype: int64
```

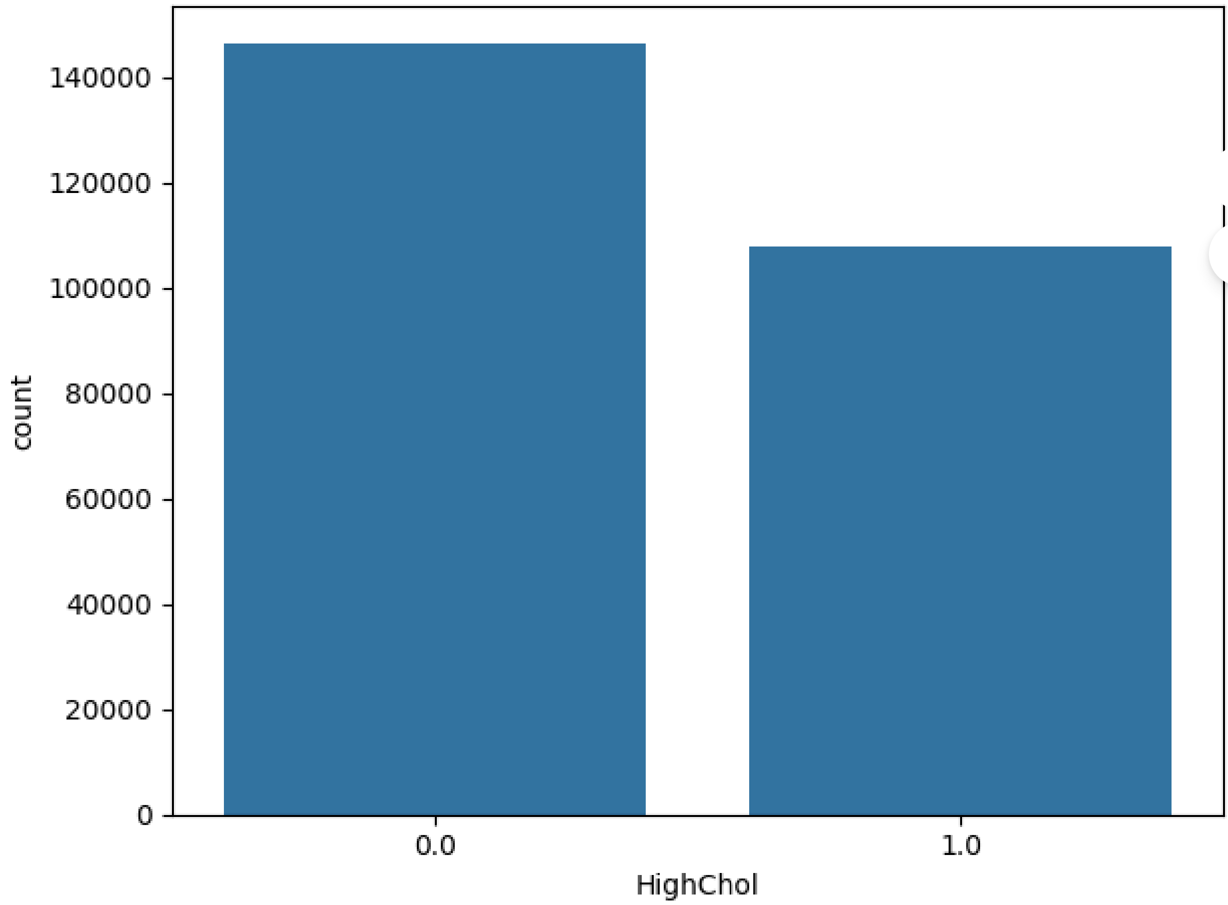



```
HighBP:
HighBP
0.0    144851
1.0    108829
Name: count, dtype: int64
```

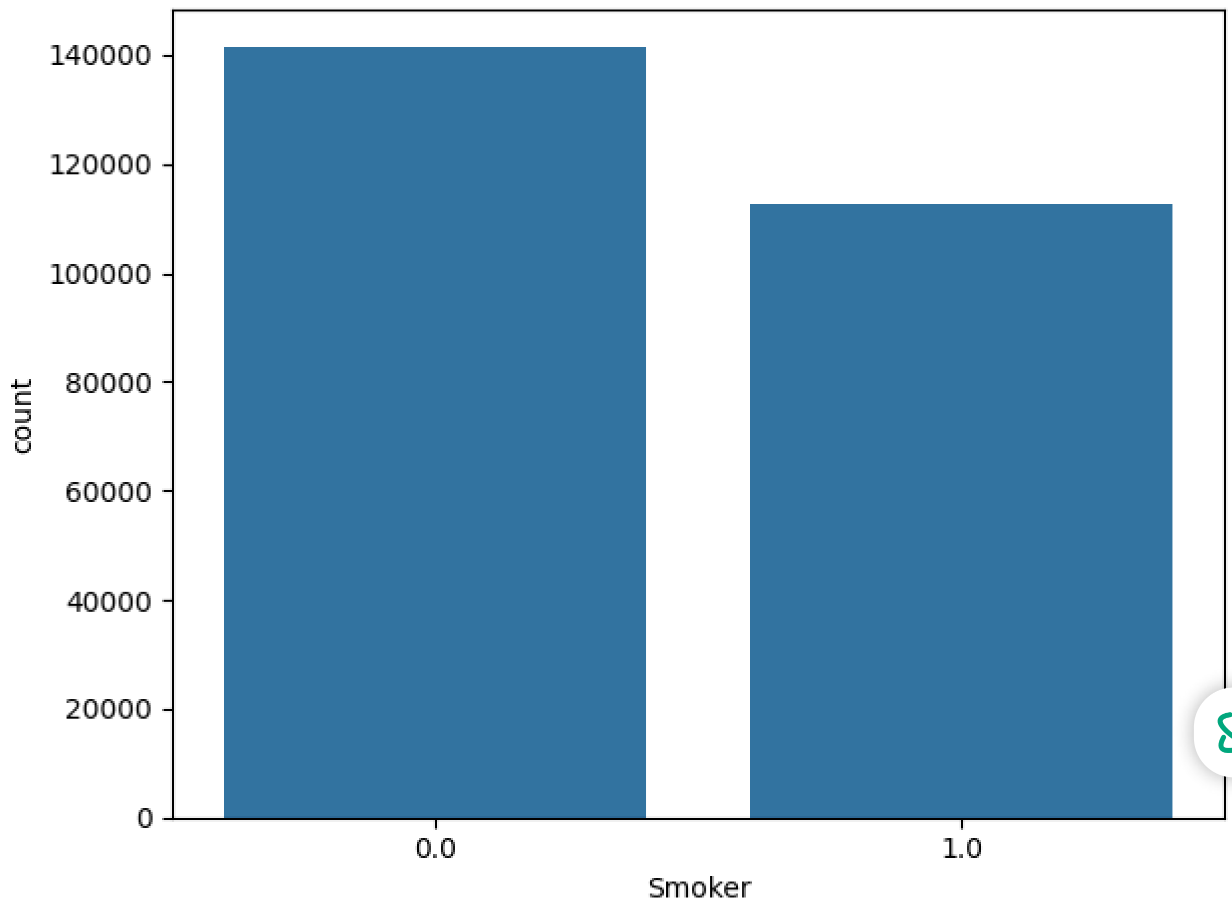



```
HighChol:
HighChol
0.0    146089
1.0    107591
```

```
Name: count, dtype: int64
```




```
Smoker:
Smoker
0.0    141257
1.0    112423
Name: count, dtype: int64
```



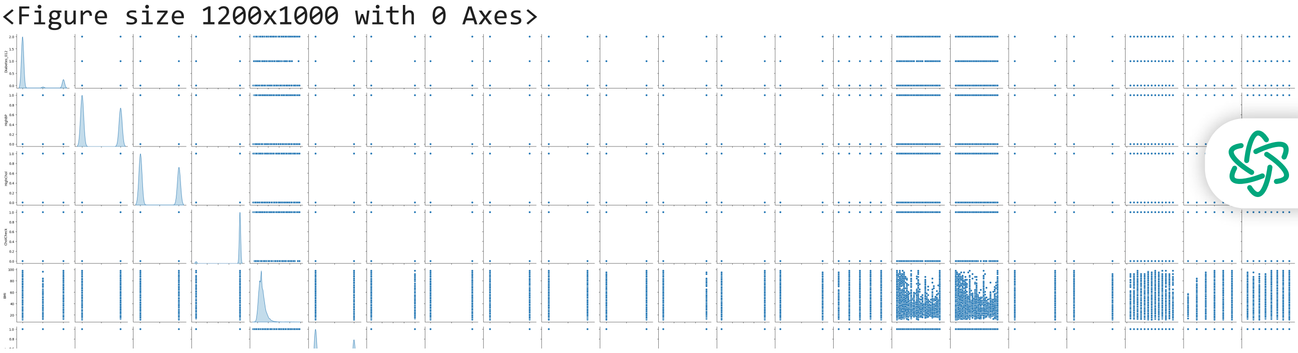
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Display the first few rows of the dataset
print("First few rows of the dataset:")
print(df.head())

# Check for missing values in each column
print("\nMissing values per column:")
print(df.isnull().sum())

# Generate descriptive statistics for numerical columns
print("\nDescriptive statistics:")
print(df.describe())

# Create a correlation matrix for numerical variables
print("\nCorrelation matrix:")
print(df.corr())

# Create a scatter plot matrix for numerical variables
plt.figure(figsize=(12, 10))
sns.pairplot(df, diag_kind="kde")
plt.tight_layout()
plt.show()

# Handling missing values
df = df.dropna()  # Dropping rows with missing values
# Alternatively, fill missing values with the mean: df = df.fillna(df.mean())

# Encoding categorical variables using one-hot encoding
categorical_cols = ['Sex', 'HighBP', 'HighChol', 'Smoker']
df = pd.get_dummies(df, columns=categorical_cols, drop_first=True)

# Splitting features and labels
X = df.drop('Diabetes_012', axis=1)
y = df['Diabetes_012']

# Display the shapes of features and labels
print("\nShapes of features and labels:")
print("Features shape:", X.shape)
print("Labels shape:", y.shape)
```

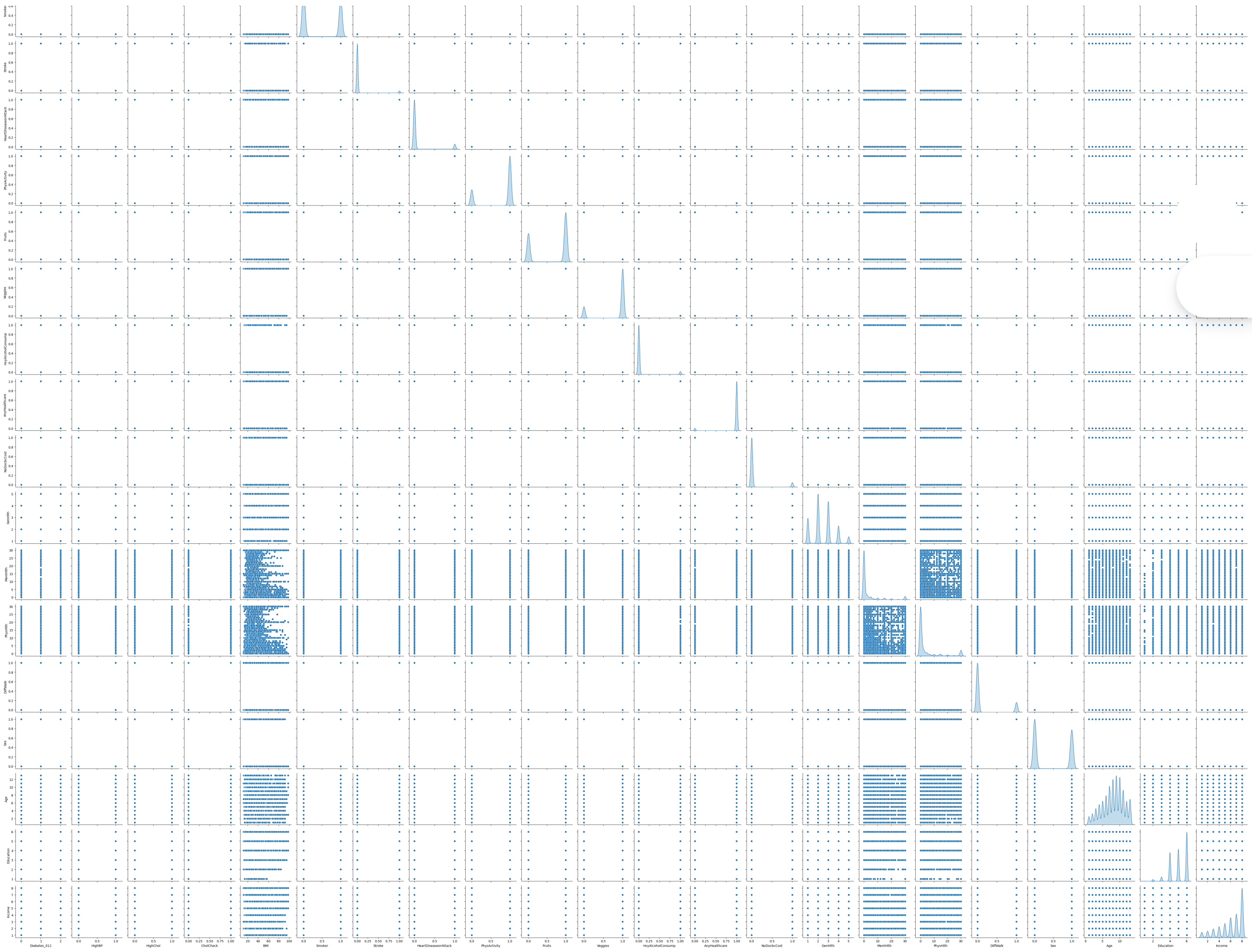
```
First few rows of the dataset:
   Diabetes_012  HighBP  HighChol  CholCheck   BMI  Smoker  Stroke  \
0           0.0     1.0       1.0        1.0  40.0     1.0     0.0
1           0.0     0.0       0.0        0.0  25.0     1.0     0.0
2           0.0     1.0       1.0        1.0  28.0     0.0     0.0
3           0.0     1.0       0.0        1.0  27.0     0.0     0.0
4           0.0     1.0       1.0        1.0  24.0     0.0     0.0

   HeartDiseaseorAttack  PhysActivity  Fruits  ...  AnyHealthcare  \
0                   0.0           0.0     0.0  ...            1.0
1                   0.0           1.0     0.0  ...            0.0
2                   0.0           0.0     1.0  ...            1.0
3                   0.0           1.0     1.0  ...            1.0
4                   0.0           1.0     1.0  ...            1.0

   NoDocbcCost  GenHlth  MentHlth  PhysHlth  DiffWalk  Sex   Age  Education  \
0          0.0      5.0      18.0      15.0       1.0  0.0   9.0        4.0
1          1.0      3.0       0.0       0.0       0.0  0.0   7.0        6.0
2          1.0      5.0      30.0      30.0       1.0  0.0   9.0        4.0
3          0.0      2.0       0.0       0.0       0.0  0.0  11.0        3.0
4          0.0      2.0       3.0       0.0       0.0  0.0  11.0        5.0

   Income
0     3.0
1     1.0
2     8.0
3     6.0
4     4.0

[5 rows x 22 columns]

Missing values per column:
Diabetes_012            0
HighBP                  0
HighChol                0
CholCheck               0
BMI                     0
Smoker                  0
Stroke                  0
HeartDiseaseorAttack    0
PhysActivity            0
Fruits                  0
Veggies                 0
HvyAlcoholConsump       0
AnyHealthcare           0
NoDocbcCost             0
GenHlth                 0
MentHlth                0
PhysHlth                0
DiffWalk                0
Sex                     0
Age                     0
Education               0
Income                  0
dtype: int64

Descriptive statistics:
        Diabetes_012         HighBP       HighChol      CholCheck  \
count  253680.000000  253680.000000  253680.000000  253680.000000
mean        0.296921       0.429001       0.424121       0.962670
```

```
std        0.698160       0.494934       0.494210       0.189571
min        0.000000       0.000000       0.000000       0.000000
25%        0.000000       0.000000       0.000000       1.000000
50%        0.000000       0.000000       0.000000       1.000000
75%        0.000000       1.000000       1.000000       1.000000
max        2.000000       1.000000       1.000000       1.000000

                 BMI         Smoker         Stroke  HeartDiseaseorAttack  \
count  253680.000000  253680.000000  253680.000000         253680.000000
mean       28.382364       0.443169       0.040571              0.094186
std         6.608694       0.496761       0.197294              0.292087
min        12.000000       0.000000       0.000000              0.000000
25%        24.000000       0.000000       0.000000              0.000000
50%        27.000000       0.000000       0.000000              0.000000
75%        31.000000       1.000000       0.000000              0.000000
max        98.000000       1.000000       1.000000              1.000000

        PhysActivity         Fruits  ...  AnyHealthcare    NoDocbcCost  \
count  253680.000000  253680.000000  ...  253680.000000  253680.000000
mean        0.756544       0.634256  ...       0.951053       0.084177
std         0.429169       0.481639  ...       0.215759       0.277654
min         0.000000       0.000000  ...       0.000000       0.000000
25%         1.000000       0.000000  ...       1.000000       0.000000
50%         1.000000       1.000000  ...       1.000000       0.000000
75%         1.000000       1.000000  ...       1.000000       0.000000
max         1.000000       1.000000  ...       1.000000       1.000000

             GenHlth       MentHlth       PhysHlth       DiffWalk  \
count  253680.000000  253680.000000  253680.000000  253680.000000
mean        2.511392       3.184772       4.242081       0.168224
std         1.068477       7.412847       8.717951       0.374066
min         1.000000       0.000000       0.000000       0.000000
25%         2.000000       0.000000       0.000000       0.000000
50%         2.000000       0.000000       0.000000       0.000000
75%         3.000000       2.000000       3.000000       0.000000
max         5.000000      30.000000      30.000000       1.000000

                 Sex            Age      Education         Income
count  253680.000000  253680.000000  253680.000000  253680.000000
mean        0.440342       8.032119       5.050434       6.053875
std         0.496429       3.054220       0.985774       2.071148
min         0.000000       1.000000       1.000000       1.000000
25%         0.000000       6.000000       4.000000       5.000000
50%         0.000000       8.000000       5.000000       7.000000
75%         1.000000      10.000000       6.000000       8.000000
max         1.000000      13.000000       6.000000       8.000000

[8 rows x 22 columns]

Correlation matrix:
                      Diabetes_012    HighBP  HighChol  CholCheck       BMI  \
Diabetes_012              1.000000  0.271596  0.209085   0.067546  0.224379
HighBP                    0.271596  1.000000  0.298199   0.098508  0.213748
HighChol                  0.209085  0.298199  1.000000   0.085642  0.106722
CholCheck                 0.067546  0.098508  0.085642   1.000000  0.034495
BMI                       0.224379  0.213748  0.106722   0.034495  1.000000
Smoker                    0.062914  0.096991  0.091299  -0.009929  0.013804
Stroke                    0.107179  0.129575  0.092620   0.024158  0.020153
HeartDiseaseorAttack      0.180272  0.209361  0.180765   0.044206  0.052904
PhysActivity             -0.121947 -0.125267 -0.078046   0.004190 -0.147294
```

```
Fruits                -0.042192 -0.040555 -0.040859  0.023849 -0.087518
Veggies               -0.058972 -0.061266 -0.039874  0.006121 -0.062275
HvyAlcoholConsump     -0.057882 -0.003972 -0.011543 -0.023730 -0.048736
AnyHealthcare          0.015410  0.038425  0.042230  0.117626 -0.018471
NoDocbcCost            0.035436  0.017358  0.013310 -0.058255  0.058206
GenHlth                0.302587  0.300530  0.208426  0.046589  0.239185
MentHlth               0.073507  0.056456  0.062069 -0.008366  0.085310
PhysHlth               0.176287  0.161212  0.121751  0.031775  0.121141
DiffWalk               0.224239  0.223618  0.144672  0.040585  0.197078
Sex                    0.031040  0.052207  0.031205 -0.022115  0.042950
Age                    0.185026  0.344452  0.272318  0.090321 -0.036618
Education             -0.130517 -0.141358 -0.070802  0.001510 -0.103932
Income                -0.171483 -0.171235 -0.085459  0.014259 -0.100069

                        Smoker    Stroke  HeartDiseaseorAttack  PhysActivity  \
Diabetes_012          0.062914  0.107179              0.180272     -0.121947
HighBP                0.096991  0.129575              0.209361     -0.125267
HighChol              0.091299  0.092620              0.180765     -0.078046
CholCheck            -0.009929  0.024158              0.044206      0.004190
BMI                   0.013804  0.020153              0.052904     -0.147294
Smoker                1.000000  0.061173              0.114441     -0.087401
Stroke                0.061173  1.000000              0.203002     -0.069151
HeartDiseaseorAttack  0.114441  0.203002              1.000000     -0.087299
PhysActivity         -0.087401 -0.069151             -0.087299      1.000000
Fruits               -0.077666 -0.013389             -0.019790      0.142756
Veggies              -0.030678 -0.041124             -0.039167      0.153150
HvyAlcoholConsump     0.101619 -0.016950             -0.028991      0.012392
AnyHealthcare        -0.023251  0.008776              0.018734      0.035505
NoDocbcCost           0.048946  0.034804              0.031000     -0.061638
GenHlth               0.163143  0.177942              0.258383     -0.266186
MentHlth              0.092196  0.070172              0.064621     -0.125587
PhysHlth              0.116460  0.148944              0.181698     -0.219230
DiffWalk              0.122463  0.176567              0.212709     -0.253174
Sex                   0.093662  0.002978              0.086096      0.032482
Age                   0.120641  0.126974              0.221618     -0.092511
Education            -0.161955 -0.076009             -0.099600      0.199658
Income               -0.123937 -0.128599             -0.141011      0.198539

                        Fruits  ...  AnyHealthcare  NoDocbcCost   GenHlth  \
Diabetes_012         -0.042192  ...       0.015410     0.035436  0.302587
HighBP               -0.040555  ...       0.038425     0.017358  0.300530
HighChol             -0.040859  ...       0.042230     0.013310  0.208426
CholCheck             0.023849  ...       0.117626    -0.058255  0.046589
BMI                  -0.087518  ...      -0.018471     0.058206  0.239185
Smoker               -0.077666  ...      -0.023251     0.048946  0.163143
Stroke               -0.013389  ...       0.008776     0.034804  0.177942
HeartDiseaseorAttack -0.019790  ...       0.018734     0.031000  0.258383
PhysActivity          0.142756  ...       0.035505    -0.061638 -0.266186
Fruits                1.000000  ...       0.031544    -0.044243 -0.103854
Veggies               0.254342  ...       0.029584    -0.032232 -0.123066
HvyAlcoholConsump    -0.035288  ...      -0.010488     0.004684 -0.036724
AnyHealthcare         0.031544  ...       1.000000    -0.232532 -0.040817
NoDocbcCost          -0.044243  ...      -0.232532     1.000000  0.166397
GenHlth              -0.103854  ...      -0.040817     0.166397  1.000000
MentHlth             -0.068217  ...      -0.052707     0.192107  0.301674
PhysHlth             -0.044633  ...      -0.008276     0.148998  0.524364
DiffWalk             -0.048352  ...       0.007074     0.118447  0.456920
Sex                  -0.091175  ...      -0.019405    -0.044931 -0.006091
Age                   0.064547  ...       0.138046    -0.119777  0.152450
Education             0.110187  ...       0.122514    -0.100701 -0.284912
Income                0.079929  ...       0.157999    -0.203182 -0.370014
```

```
                      MentHlth  PhysHlth  DiffWalk       Sex       Age  \
Diabetes_012          0.073507  0.176287  0.224239  0.031040  0.185026
HighBP                0.056456  0.161212  0.223618  0.052207  0.344452
HighChol              0.062069  0.121751  0.144672  0.031205  0.272318
CholCheck            -0.008366  0.031775  0.040585 -0.022115  0.090321
BMI                   0.085310  0.121141  0.197078  0.042950 -0.036618
Smoker                0.092196  0.116460  0.122463  0.093662  0.120641
Stroke                0.070172  0.148944  0.176567  0.002978  0.126974
HeartDiseaseorAttack  0.064621  0.181698  0.212709  0.086096  0.221618
PhysActivity         -0.125587 -0.219230 -0.253174  0.032482 -0.092511
Fruits               -0.068217 -0.044633 -0.048352 -0.091175  0.064547
Veggies              -0.058884 -0.064290 -0.080506 -0.064765 -0.009771
HvyAlcoholConsump     0.024716 -0.026415 -0.037668  0.005740 -0.034578
AnyHealthcare        -0.052707 -0.008276  0.007074 -0.019405  0.138046
NoDocbcCost           0.192107  0.148998  0.118447 -0.044931 -0.119777
GenHlth               0.301674  0.524364  0.456920 -0.006091  0.152450
MentHlth              1.000000  0.353619  0.233688 -0.080705 -0.092068
PhysHlth              0.353619  1.000000  0.478417 -0.043137  0.099130
DiffWalk              0.233688  0.478417  1.000000 -0.070299  0.204450
Sex                  -0.080705 -0.043137 -0.070299  1.000000 -0.027340
Age                  -0.092068  0.099130  0.204450 -0.027340  1.000000
Education            -0.101830 -0.155093 -0.192642  0.019480 -0.101901
Income               -0.209806 -0.266799 -0.320124  0.127141 -0.127775

                      Education    Income
Diabetes_012          -0.130517 -0.171483
HighBP                -0.141358 -0.171235
HighChol              -0.070802 -0.085459
CholCheck              0.001510  0.014259
BMI                   -0.103932 -0.100069
Smoker                -0.161955 -0.123937
Stroke                -0.076009 -0.128599
HeartDiseaseorAttack  -0.099600 -0.141011
PhysActivity           0.199658  0.198539
Fruits                 0.110187  0.079929
Veggies                0.154329  0.151087
HvyAlcoholConsump      0.023997  0.053619
AnyHealthcare          0.122514  0.157999
NoDocbcCost           -0.100701 -0.203182
GenHlth               -0.284912 -0.370014
MentHlth              -0.101830 -0.209806
PhysHlth              -0.155093 -0.266799
DiffWalk              -0.192642 -0.320124
Sex                    0.019480  0.127141
Age                   -0.101901 -0.127775
Education              1.000000  0.449106
Income                 0.449106  1.000000

[22 rows x 22 columns]
<Figure size 1200x1000 with 0 Axes>
```





```
Shapes of features and labels:
Features shape: (253680, 21)
Labels shape: (253680,)
```

```
# Task 4: Separating features and labels
X = df.drop('Diabetes_012', axis=1)  # Features
y = df['Diabetes_012']  # Labels

# Print the shapes of features and labels
print("Shape of features (X):", X.shape)
print("Shape of labels (y):", y.shape)
```

```
Shape of features (X): (253680, 21)
Shape of labels (y): (253680,)
```

```
# Task 5: Calculating Information Gain
from sklearn.metrics import mutual_info_score

# Separate features and labels
X = df.drop('Diabetes_012', axis=1)
y = df['Diabetes_012']

# Calculate information gain for each feature
information_gain = []
for feature in X.columns:
    info_gain = mutual_info_score(X[feature], y)
    information_gain.append(info_gain)

# Create a DataFrame to display feature information gain
feature_info_gain = pd.DataFrame({'Feature': X.columns, 'Information Gain': information_
feature_info_gain = feature_info_gain.sort_values(by='Information Gain', ascending=False
print(feature_info_gain)
```

```
                 Feature  Information Gain
10               GenHlth          0.047067
18            HighBP_1.0          0.037435
1                    BMI          0.030067
14                   Age          0.022087
19          HighChol_1.0          0.022049
13              DiffWalk          0.021242
16                Income          0.015080
12              PhysHlth          0.014923
3   HeartDiseaseorAttack          0.013050
15             Education          0.008582
4           PhysActivity          0.006936
2                 Stroke          0.004573
0              CholCheck          0.003242
11              MentHlth          0.003085
7      HvyAlcoholConsump          0.002043
20            Smoker_1.0          0.001980
6                Veggies          0.001671
5                 Fruits          0.000884
9            NoDocbcCost          0.000722
17               Sex_1.0          0.000492
8          AnyHealthcare          0.000143
```

```
# Task 6: Splitting Data into Train, Validation, and Test Sets
from sklearn.model_selection import train_test_split

# Separate features and labels
X = df.drop('Diabetes_012', axis=1)
y = df['Diabetes_012']

# Split the data into train, validation, and test sets
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3, random_state=42
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.25, random_s

# Print the shapes of the data splits
print("Shape of X_train:", X_train.shape)
print("Shape of y_train:", y_train.shape)
print("Shape of X_val:", X_val.shape)
print("Shape of y_val:", y_val.shape)
print("Shape of X_test:", X_test.shape)
print("Shape of y_test:", y_test.shape)
```

```
Shape of X_train: (177576, 21)
Shape of y_train: (177576,)
Shape of X_val: (57078, 21)
Shape of y_val: (57078,)
Shape of X_test: (19026, 21)
Shape of y_test: (19026,)
```

```
# Task 7: Training a Decision Tree Model
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# Train the decision tree model
dtc = DecisionTreeClassifier(random_state=42)
dtc.fit(X_train, y_train)

# Make predictions on the test set
y_pred = dtc.predict(X_test)

# Calculate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

```
Accuracy: 0.77
```

In Task 7,

1. Imports the DecisionTreeClassifier from scikit-learn's tree module and the accuracy_score function from the metrics module.
2. Creates an instance of the DecisionTreeClassifier with random_state=42 for reproducibility.
3. Trains the decision tree model on the training data X_train and y_train using the fit method.

4. Makes predictions on the test data X_test using the predict method and stores the predictions in y_pred.
5. Calculates the accuracy of the model by comparing the predicted labels y_pred with the true labels y_test using the accuracy_score function.
6. Prints the accuracy score.

Hyperparameters of the decision tree model and their significance:

1. criterion (default='gini'): This parameter specifies the function to measure the quality of a split. The two options are 'gini' for the Gini impurity and 'entropy' for the information gain (entropy). According to the scikit-learn documentation , the Gini impurity is a measure of node impurity, and it is the default criterion for classification tasks.
2. max_depth (default=None): This parameter sets the maximum depth of the tree. A higher value allows the tree to grow deeper, which can lead to overfitting on the training data. Setting it to None (default) means that the tree can grow without any depth restriction.
3. min_samples_split (default=2): This parameter specifies the minimum number of samples required to split an internal node. A higher value can prevent overfitting by not allowing splits in nodes with too few samples.
4. min_samples_leaf (default=1): This parameter sets the minimum number of samples required to be at a leaf node. Increasing this value can prevent overfitting by pruning the tree and reducing the number of leaf nodes.
5. max_features (default=None): This parameter limits the number of features to consider when looking for the best split. Setting it to None (default) means that all features are considered for splitting.
6. random_state (default=None): This parameter controls the randomness of the decision tree's behavior when using random subsets of features for splitting. Setting a fixed value ensures reproducibility of results.

```
# Task 8: Evaluating Training Accuracy
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# Train the decision tree model
dtc = DecisionTreeClassifier(random_state=42)
dtc.fit(X_train, y_train)

# Make predictions on the train set
y_train_pred = dtc.predict(X_train)

# Calculate the training accuracy
train_accuracy = accuracy_score(y_train, y_train_pred)
print(f"Training Accuracy: {train_accuracy:.2f}")
```

```
Training Accuracy: 0.99
```