```
# Python Interpreter for my system
import sys
print(sys.executable)
```

```
/opt/homebrew/Cellar/jupyterlab/4.2.5_1/libexec/bin/python
```

```
# All the dependencies
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV, train_test_split,
cross_val_score
from sklearn.metrics import roc_auc_score, roc_curve, accuracy_score,
confusion_matrix, classification_report
from sklearn.impute import SimpleImputer

import seaborn as sns
import pickle
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Dataframe as df - Path must be given relative to its location
df = pd.read_csv('diabetese_50000.csv')

# A glimpse of the dataframe
print(df)
```

```
       Has_diabetes  HighBP  HighChol  CholCheck   BMI  Smoker  Stroke
\
0               1.0     1.0       0.0        1.0  30.0     1.0     0.0

1               1.0     1.0       1.0        1.0  27.0     1.0     0.0

2               0.0     0.0       0.0        1.0  22.0     0.0     0.0

3               0.0     0.0       0.0        1.0  38.0     0.0     0.0

4               0.0     0.0       0.0        1.0  32.0     0.0     0.0

...             ...     ...       ...        ...   ...     ...     ...

49995           1.0     1.0       1.0        1.0  33.0     0.0     0.0

49996           0.0     1.0       1.0        1.0  27.0     1.0     0.0

49997           0.0     1.0       1.0        1.0  26.0     0.0     0.0

49998           0.0     0.0       0.0        1.0  27.0     0.0     0.0

49999           0.0     0.0       0.0        1.0  24.0     1.0     0.0
```

```
       HeartDiseaseorAttack  PhysActivity  Fruits  ...  AnyHealthcare
\
0                       0.0           1.0     0.0  ...            1.0
1                       0.0           0.0     0.0  ...            1.0
2                       0.0           1.0     1.0  ...            1.0
3                       0.0           1.0     1.0  ...            1.0
4                       0.0           1.0     1.0  ...            1.0
...                     ...           ...     ...  ...            ...
49995                   0.0           0.0     0.0  ...            1.0
49996                   0.0           1.0     0.0  ...            1.0
49997                   0.0           1.0     1.0  ...            1.0
49998                   0.0           1.0     1.0  ...            1.0
49999                   0.0           1.0     0.0  ...            1.0

       NoDocbcCost  GenHlth  MentHlth  PhysHlth  DiffWalk  Sex
Age  \
0              0.0      3.0       0.0       1.0       0.0  1.0   8.0
1              0.0      3.0       0.0       7.0       0.0  1.0   9.0
2              0.0      2.0       0.0       0.0       0.0  0.0   6.0
3              0.0      3.0       0.0       0.0       0.0  1.0   7.0
4              0.0      2.0       5.0       0.0       0.0  0.0   5.0
...            ...      ...       ...       ...       ...  ...   ...
49995          0.0      3.0       5.0       0.0       1.0  0.0  10.0
49996          0.0      2.0      20.0       5.0       0.0  1.0   9.0
49997          0.0      3.0       0.0       0.0       0.0  1.0   9.0
49998          0.0      1.0       0.0       0.0       0.0  1.0  11.0
49999          0.0      2.0       0.0       0.0       0.0  0.0   3.0
```

```
       Education  Income
0            6.0     8.0
1            4.0     7.0
2            6.0     8.0
3            6.0     8.0
4            5.0     7.0
...          ...     ...
49995        6.0     5.0
49996        4.0     6.0
49997        6.0     6.0
49998        6.0     8.0
49999        6.0     8.0

[50000 rows x 22 columns]
```

```
# Information about the data
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 22 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Has_diabetes          50000 non-null  float64
 1   HighBP                50000 non-null  float64
 2   HighChol              50000 non-null  float64
 3   CholCheck             50000 non-null  float64
 4   BMI                   50000 non-null  float64
 5   Smoker                50000 non-null  float64
 6   Stroke                50000 non-null  float64
 7   HeartDiseaseorAttack  50000 non-null  float64
 8   PhysActivity          50000 non-null  float64
 9   Fruits                50000 non-null  float64
 10  Veggies               50000 non-null  float64
 11  HvyAlcoholConsump     50000 non-null  float64
 12  AnyHealthcare         50000 non-null  float64
 13  NoDocbcCost           50000 non-null  float64
 14  GenHlth               50000 non-null  float64
 15  MentHlth              50000 non-null  float64
 16  PhysHlth              50000 non-null  float64
 17  DiffWalk              50000 non-null  float64
 18  Sex                   50000 non-null  float64
 19  Age                   50000 non-null  float64
 20  Education             50000 non-null  float64
 21  Income                50000 non-null  float64
dtypes: float64(22)
memory usage: 8.4 MB
None
```

```python
# Shape of the data - 50000 rows and 22 columns
print(df.shape)
```

```
(50000, 22)
```

```python
# Counts all the features and their respective values and type(s)
for col in df.columns:
    print(df[col].value_counts())
    print(df[col].dtype)
    print()
```

```
Has_diabetes
0.0    25074
1.0    24926
Name: count, dtype: int64
float64

HighBP
1.0    28149
0.0    21851
Name: count, dtype: int64
float64

HighChol
1.0    26297
0.0    23703
Name: count, dtype: int64
float64

CholCheck
1.0    48758
0.0     1242
Name: count, dtype: int64
float64

BMI
27.0    4498
26.0    3487
28.0    3244
24.0    3102
30.0    3077
        ...
12.0       1
85.0       1
86.0       1
80.0       1
98.0       1
Name: count, Length: 79, dtype: int64
float64
```

```
Smoker
0.0    26292
1.0    23708
Name: count, dtype: int64
float64

Stroke
0.0    46899
1.0     3101
Name: count, dtype: int64
float64

HeartDiseaseorAttack
0.0    42623
1.0     7377
Name: count, dtype: int64
float64

PhysActivity
1.0    35163
0.0    14837
Name: count, dtype: int64
float64

Fruits
1.0    30575
0.0    19425
Name: count, dtype: int64
float64

Veggies
1.0    39380
0.0    10620
Name: count, dtype: int64
float64

HvyAlcoholConsump
0.0    47849
1.0     2151
Name: count, dtype: int64
float64

AnyHealthcare
1.0    47769
0.0     2231
Name: count, dtype: int64
float64

NoDocbcCost
0.0    45324
```

```
1.0     4676
Name: count, dtype: int64
float64

GenHlth
3.0    16572
2.0    14082
4.0     9390
1.0     5866
5.0     4090
Name: count, dtype: int64
float64

MentHlth
0.0     33928
30.0     3060
2.0      2331
5.0      1816
1.0      1461
3.0      1396
10.0     1366
15.0     1240
20.0      809
4.0       705
7.0       566
25.0      309
14.0      247
6.0       199
8.0       136
12.0       95
28.0       72
21.0       57
29.0       40
18.0       31
16.0       20
22.0       17
9.0        16
26.0       14
27.0       13
17.0       13
13.0       11
23.0       11
24.0        9
19.0        6
11.0        6
Name: count, dtype: int64
float64

PhysHlth
0.0     28247
```

```
30.0     5617
2.0      2902
1.0      2031
3.0      1775
5.0      1651
10.0     1386
15.0     1328
4.0       964
7.0       948
20.0      906
14.0      562
25.0      383
6.0       318
8.0       191
21.0      162
12.0      148
28.0      143
29.0       69
16.0       38
9.0        33
18.0       33
22.0       25
27.0       23
17.0       22
23.0       20
13.0       18
24.0       17
26.0       16
11.0       14
19.0       10
Name: count, dtype: int64
float64

DiffWalk
0.0    37347
1.0    12653
Name: count, dtype: int64
float64

Sex
0.0    27126
1.0    22874
Name: count, dtype: int64
float64

Age
10.0    7673
9.0     7222
8.0     6101
11.0    5687
```

```
7.0     4775
12.0    3825
13.0    3799
6.0     3225
5.0     2515
4.0     1992
3.0     1478
2.0      996
1.0      712
Name: count, dtype: int64
float64

Education
6.0    18531
5.0    14136
4.0    13700
3.0     2418
2.0     1163
1.0       52
Name: count, dtype: int64
float64

Income
8.0    14644
7.0     8120
6.0     7171
5.0     5737
4.0     4709
3.0     3880
2.0     3178
1.0     2561
Name: count, dtype: int64
float64


# Identifying numerical and categorical features by count of unique
values.
# I choose 10 to be the threshold to go from categorial to numerical

cat_cols = []
num_cols = []

for col in df.columns:
    if col == 'Has_diabetes':
        continue
    if len(df[col].unique()) < 10:
        cat_cols.append(col)
    else:
        num_cols.append(col)
print("\nNumerical: ")
```

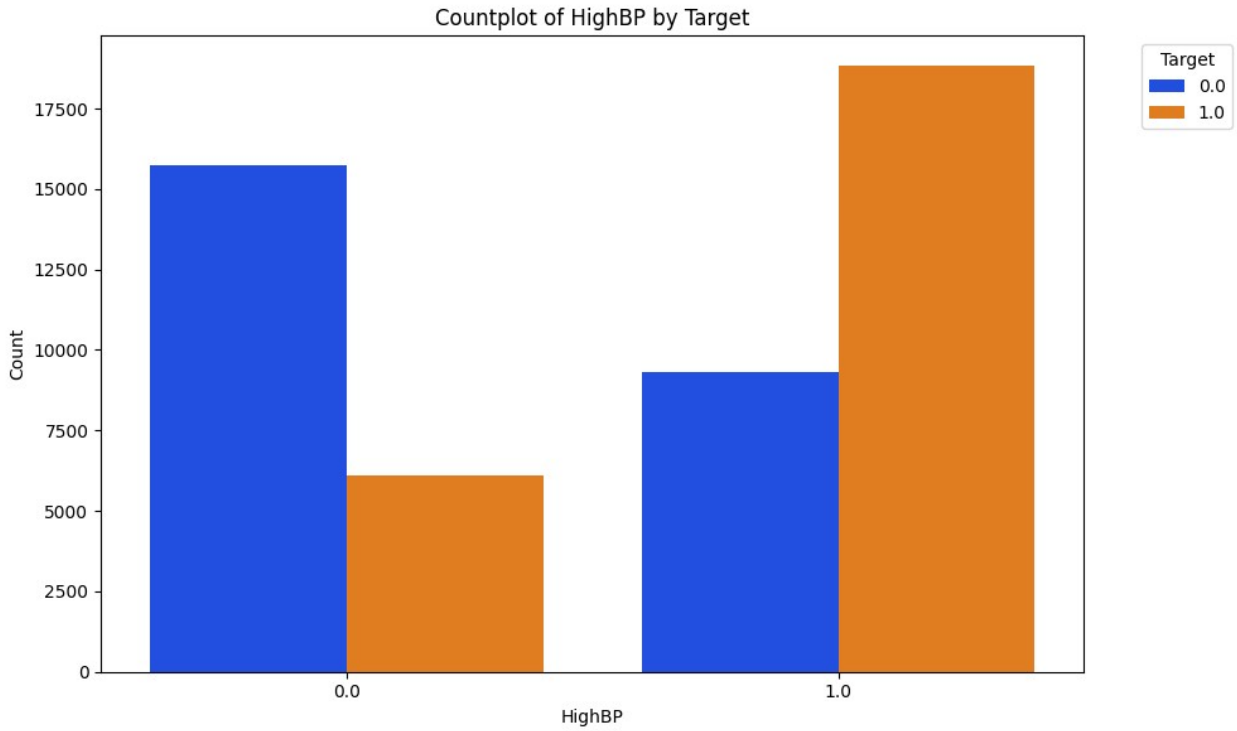
```
print(*num_cols, sep = "\n")

print("\nCategorical: ")
print(*cat_cols, sep = "\n")
```

```
Numerical:
BMI
MentHlth
PhysHlth
Age

Categorical:
HighBP
HighChol
CholCheck
Smoker
Stroke
HeartDiseaseorAttack
PhysActivity
Fruits
Veggies
HvyAlcoholConsump
AnyHealthcare
NoDocbcCost
GenHlth
DiffWalk
Sex
Education
Income
```

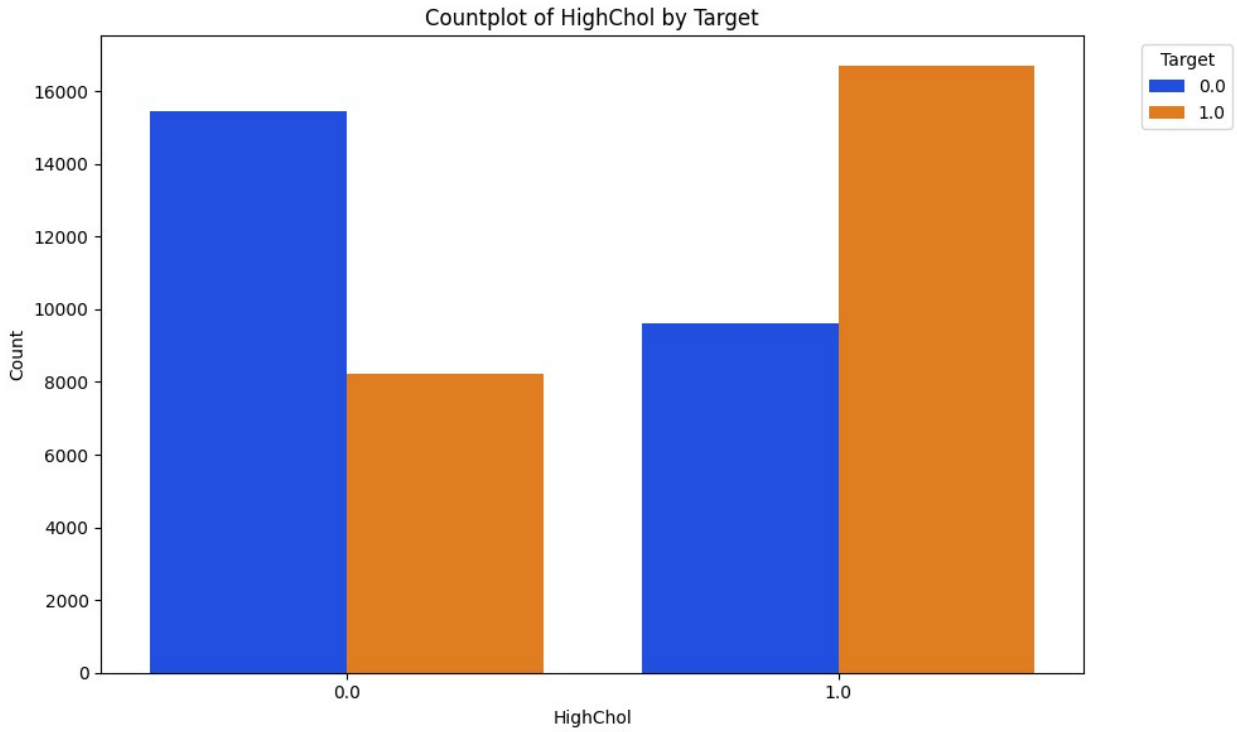
```
# Graphical representation of categories using category columns.

# For each category
for col in cat_cols:
    print(f"Generating countplot for: {col}")

    # Figure size 10 x 6
    plt.figure(figsize=(10, 6))

    # Gerenrate graph(s) using 'bright' color
    sns.countplot(data=df, x=col, hue='Has_diabetes',
palette='bright')

    # Set plot title and labels
    plt.title(f'Countplot of {col} by Target')
    plt.xlabel(col)
    plt.ylabel('Count')

    # Adjustment to the legend(key)
```

```
    plt.legend(title='Target', bbox_to_anchor=(1.05, 1), loc='upper
left')

    # Display the plot using tight layout to adjust padding
    plt.tight_layout()
    plt.show()
```

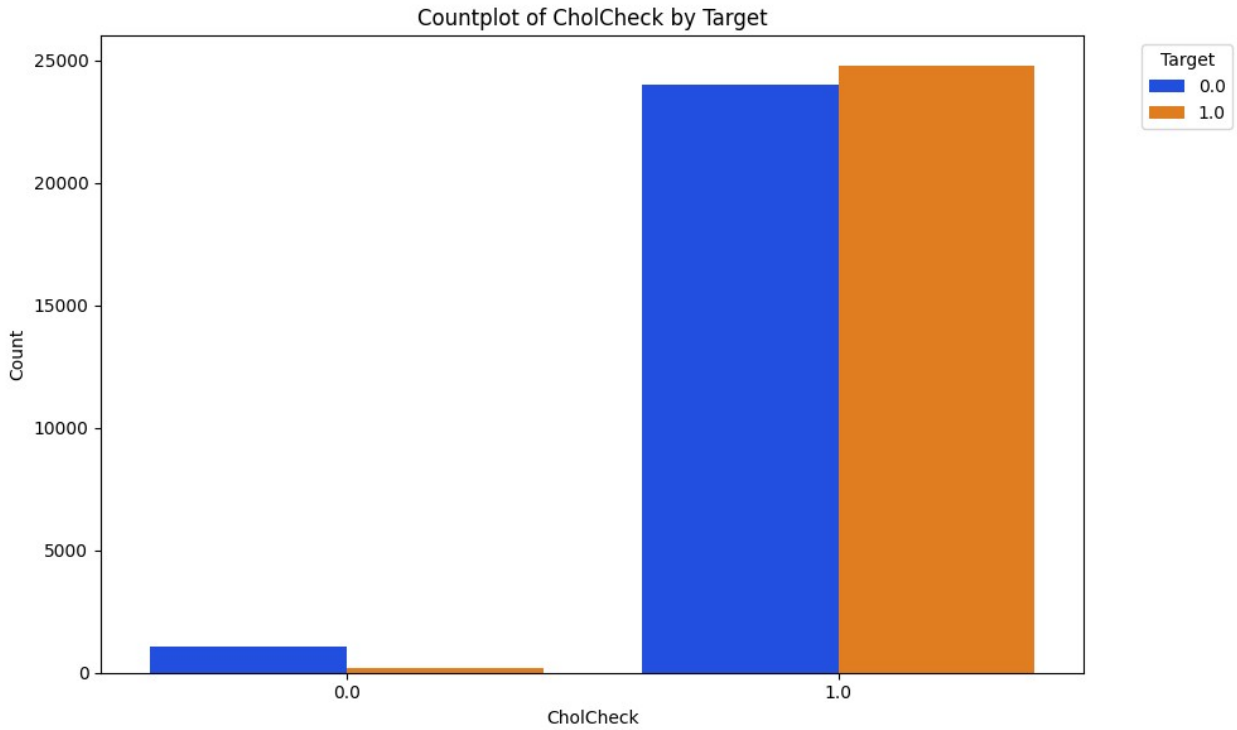
```
Generating countplot for: HighBP
```



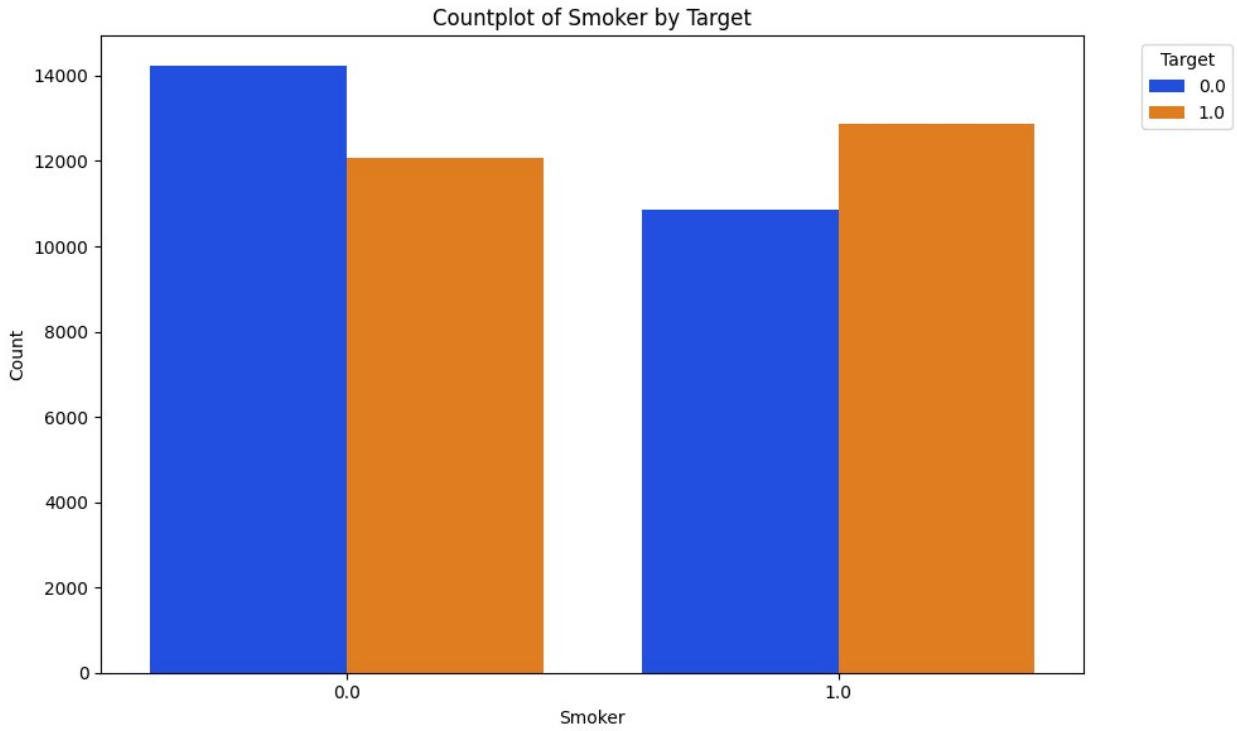

```
Generating countplot for: HighChol
```



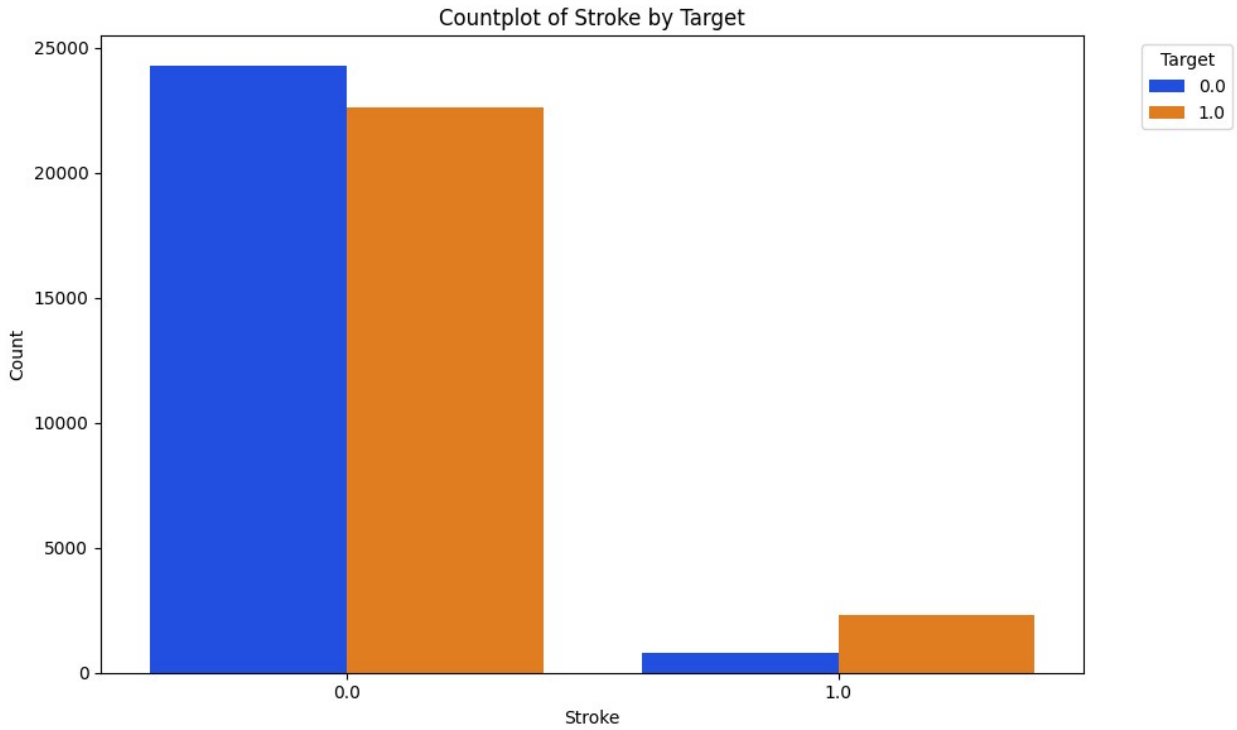

```
Generating countplot for: CholCheck
```



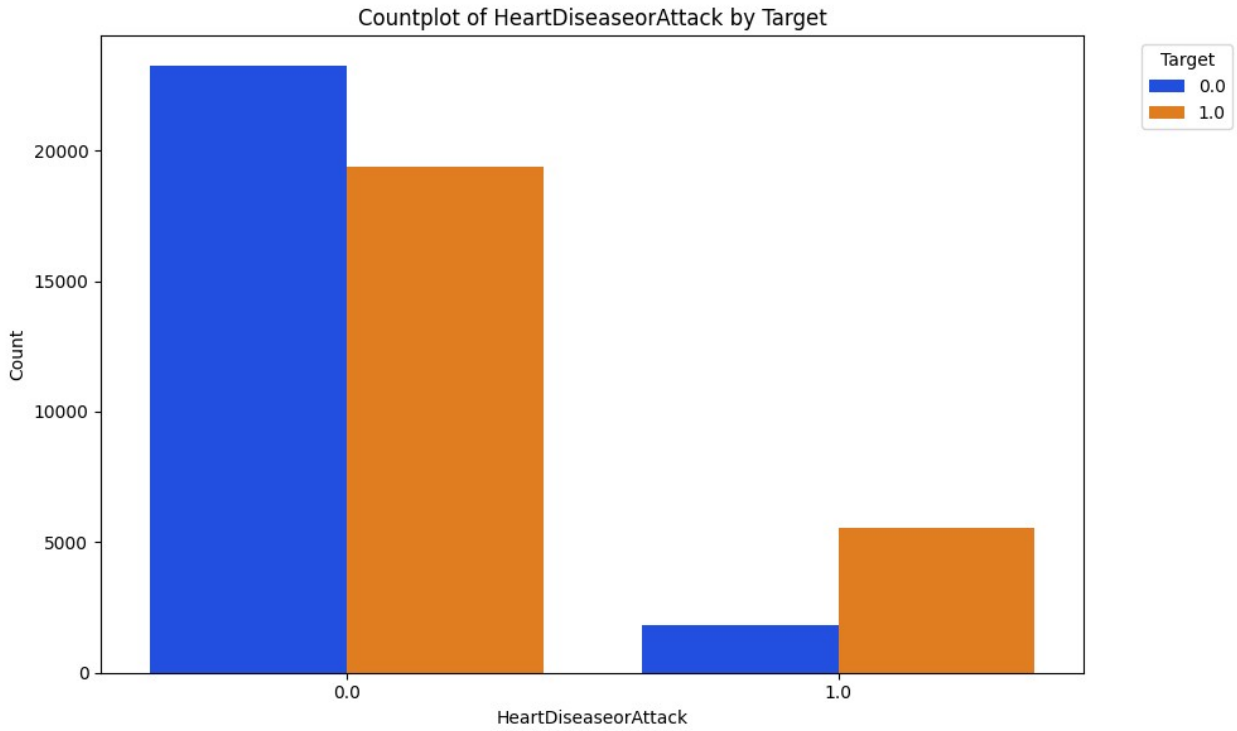

```
Generating countplot for: Smoker
```



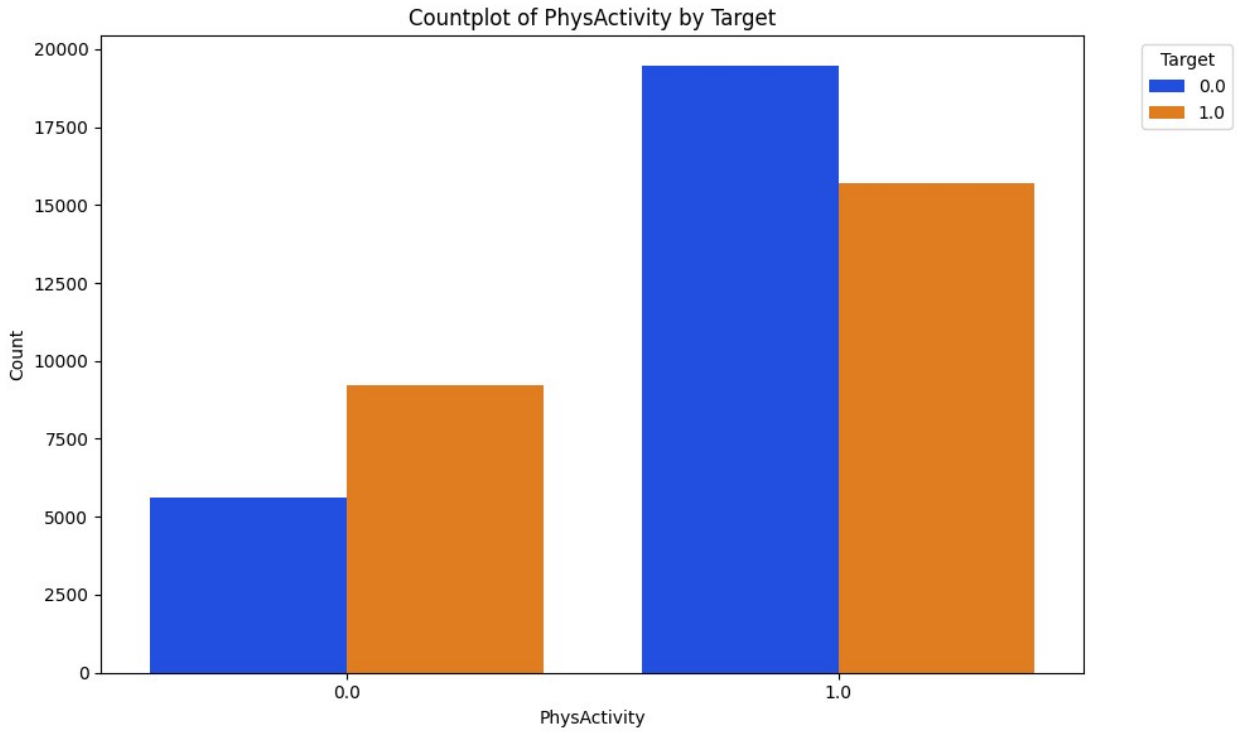

```
Generating countplot for: Stroke
```



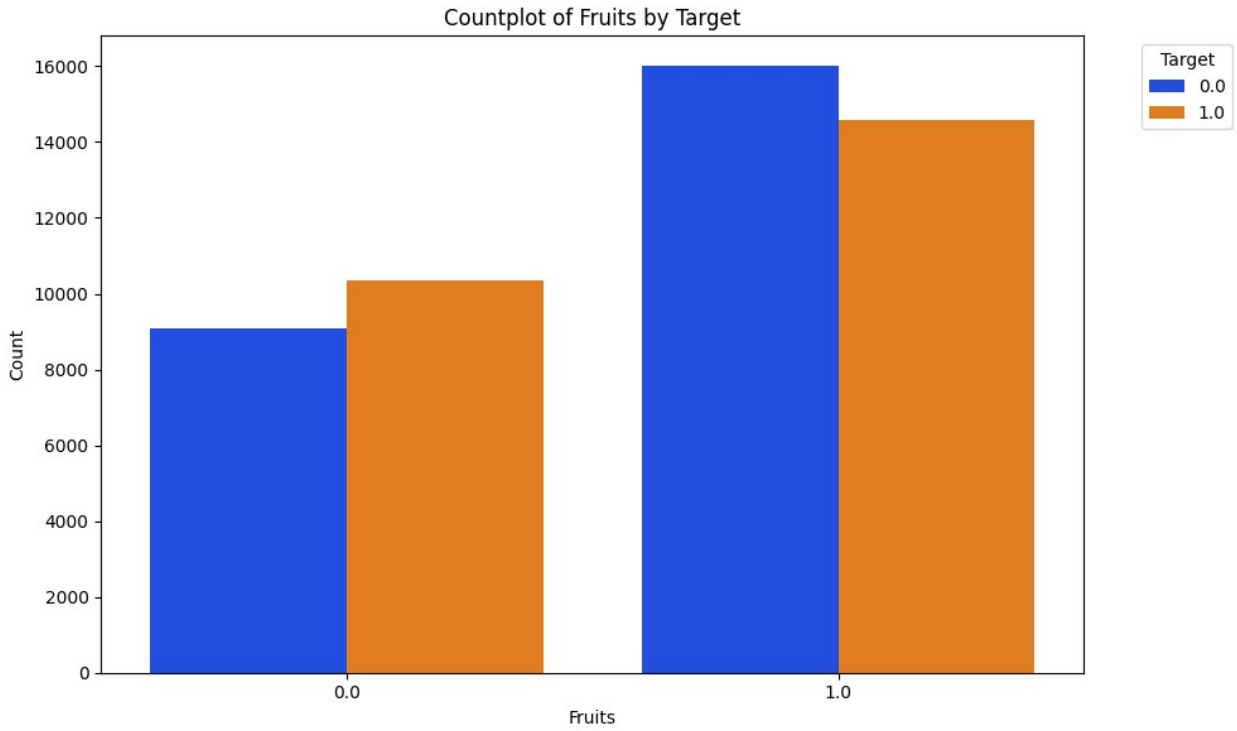

```
Generating countplot for: HeartDiseaseorAttack
```



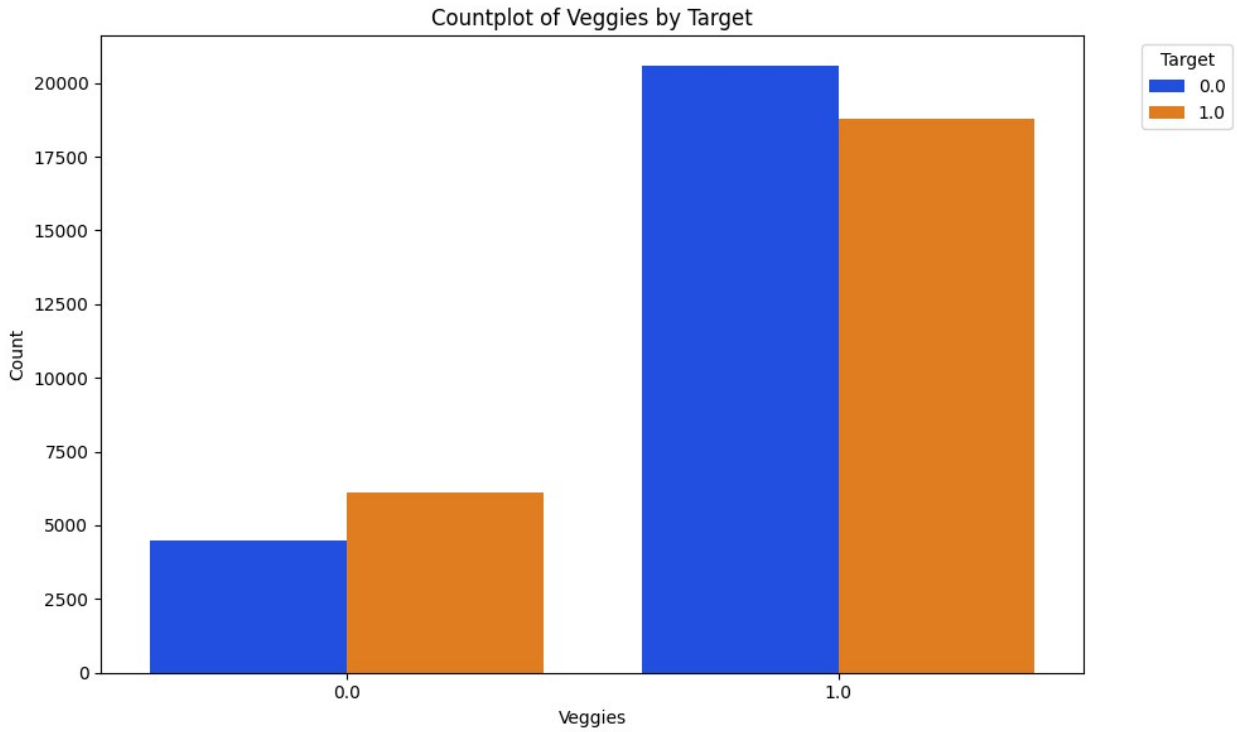

```
Generating countplot for: PhysActivity
```



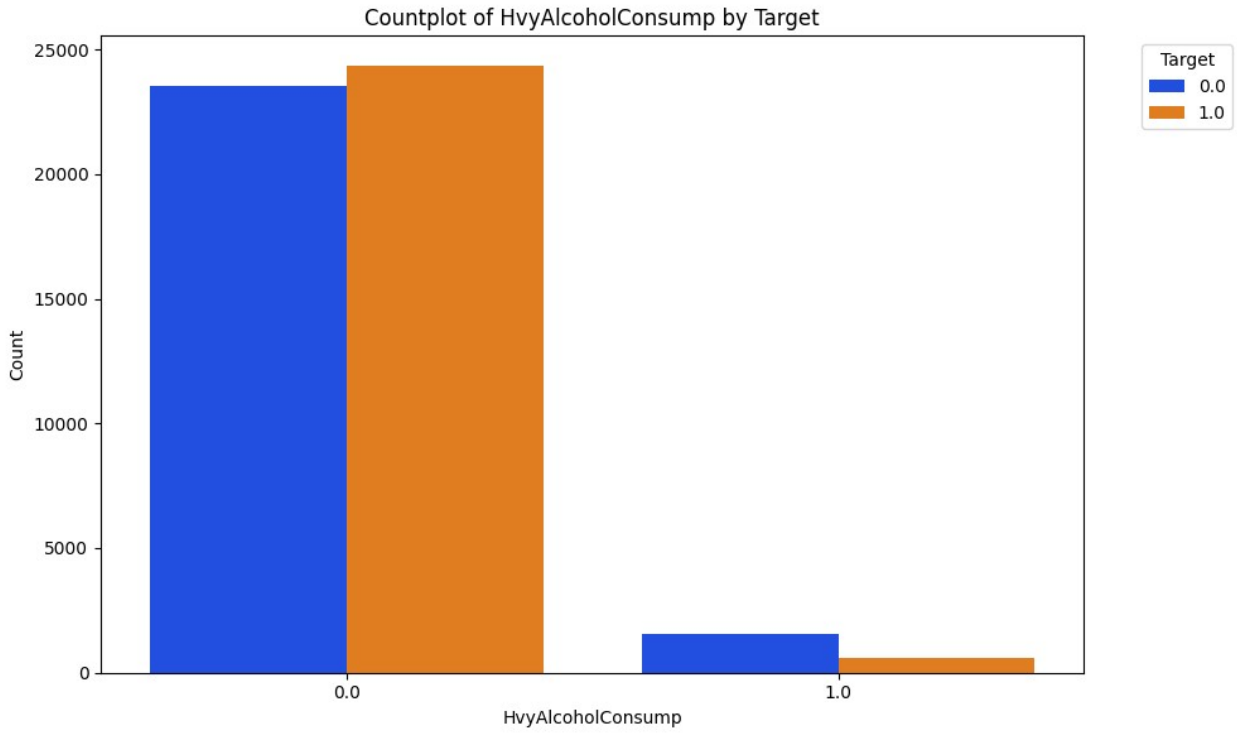

```
Generating countplot for: Fruits
```



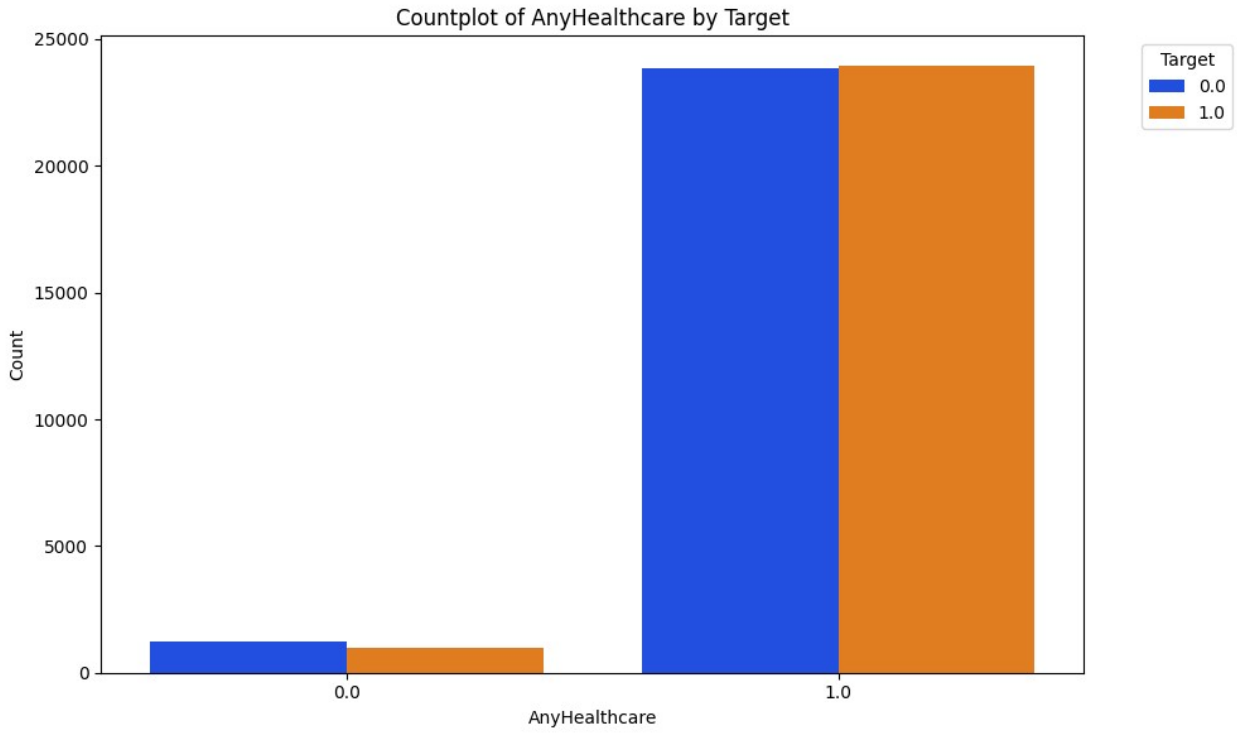

```
Generating countplot for: Veggies
```



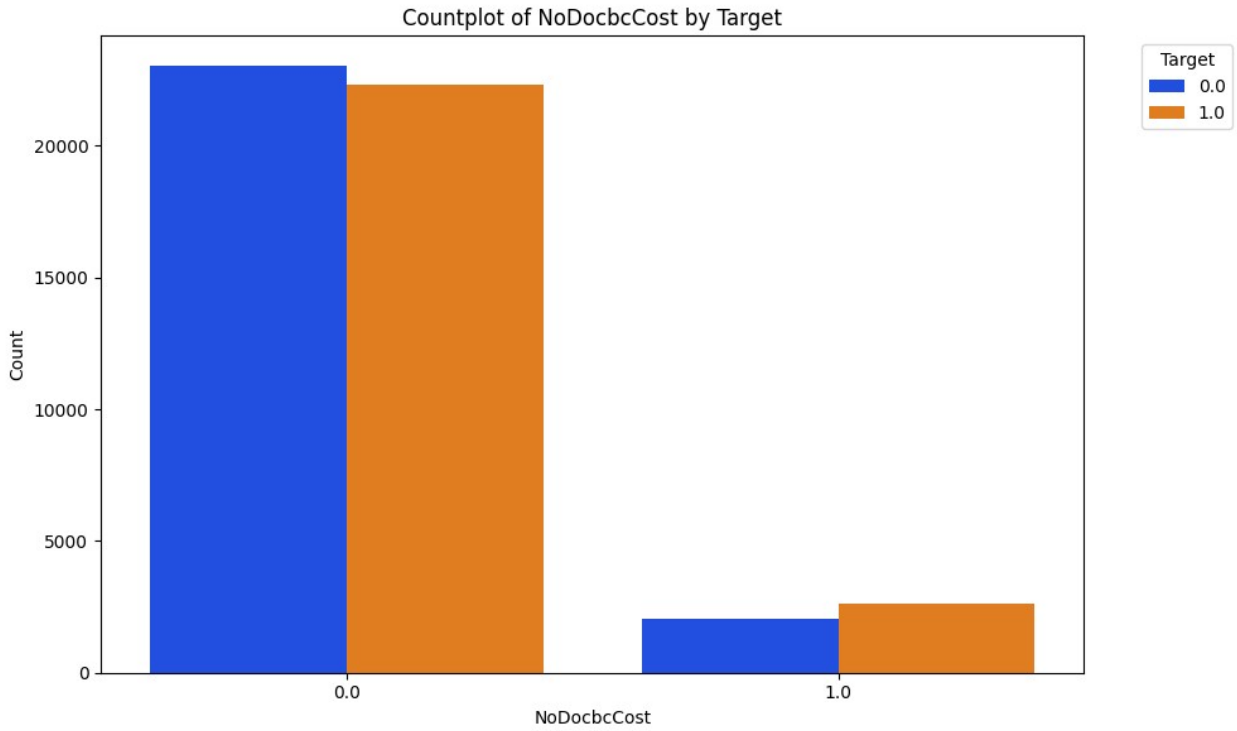

```
Generating countplot for: HvyAlcoholConsump
```



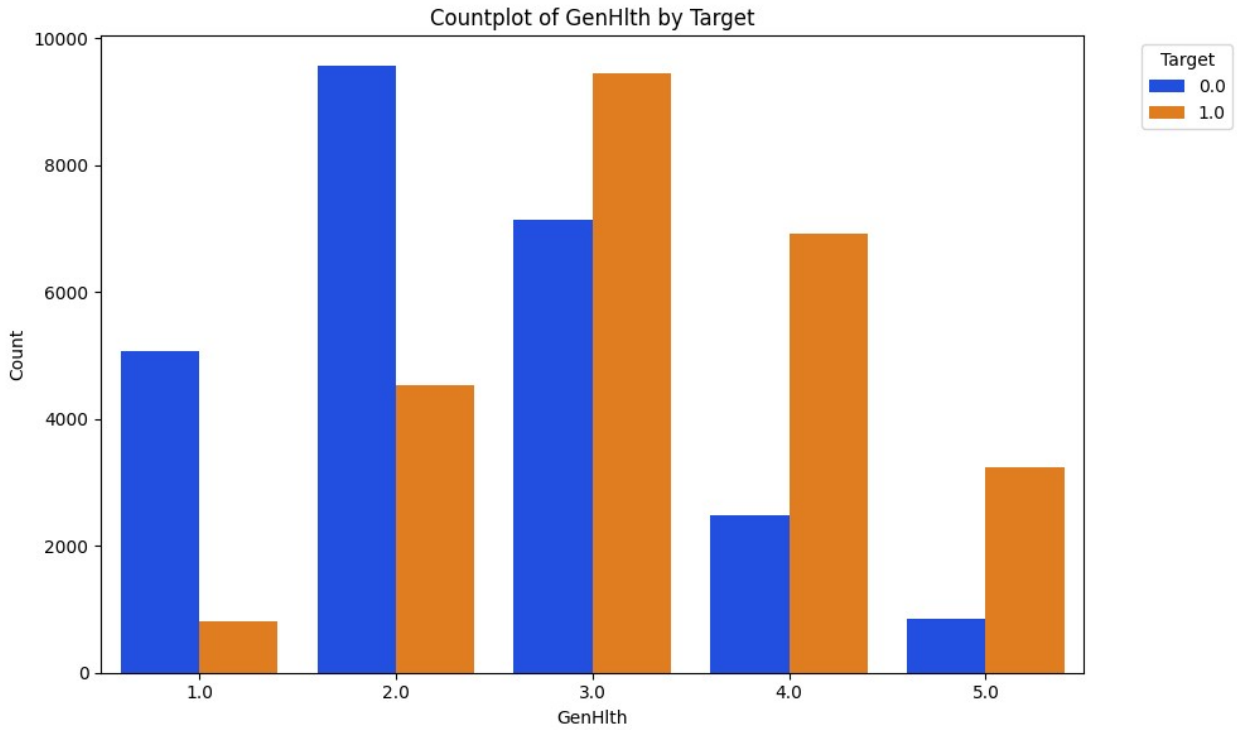

```
Generating countplot for: AnyHealthcare
```



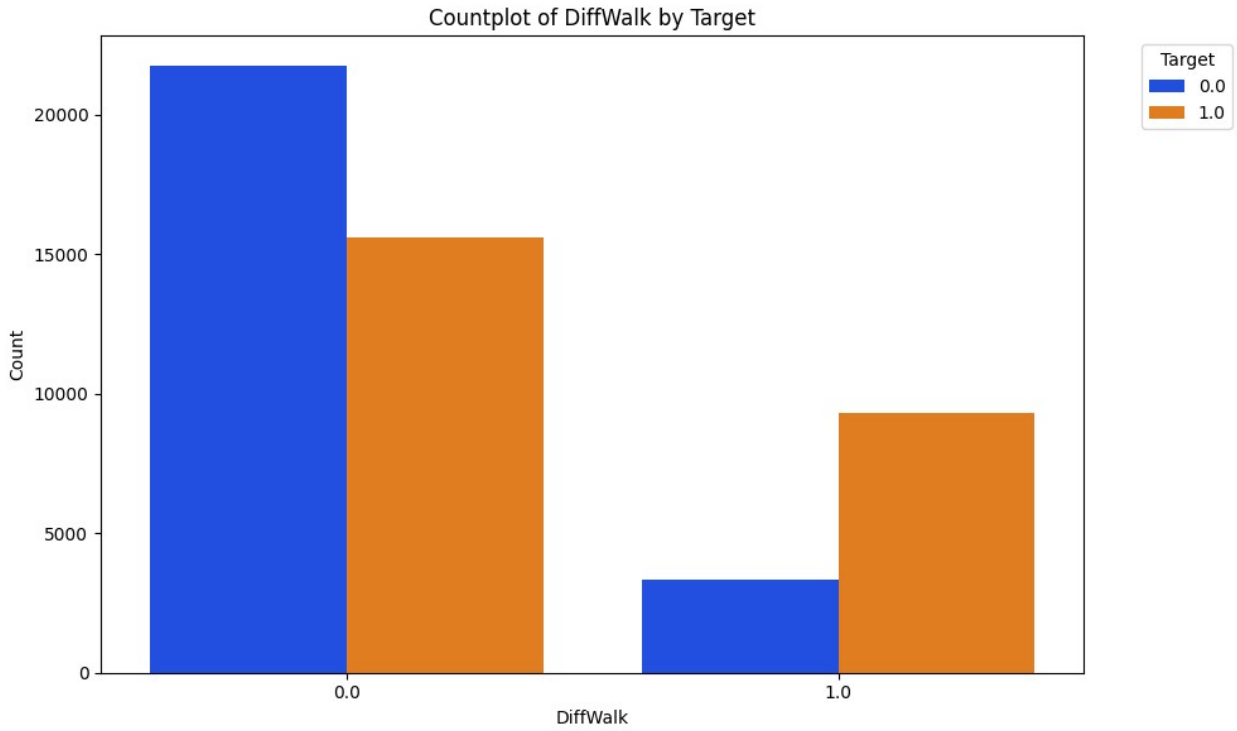

```
Generating countplot for: NoDocbcCost
```



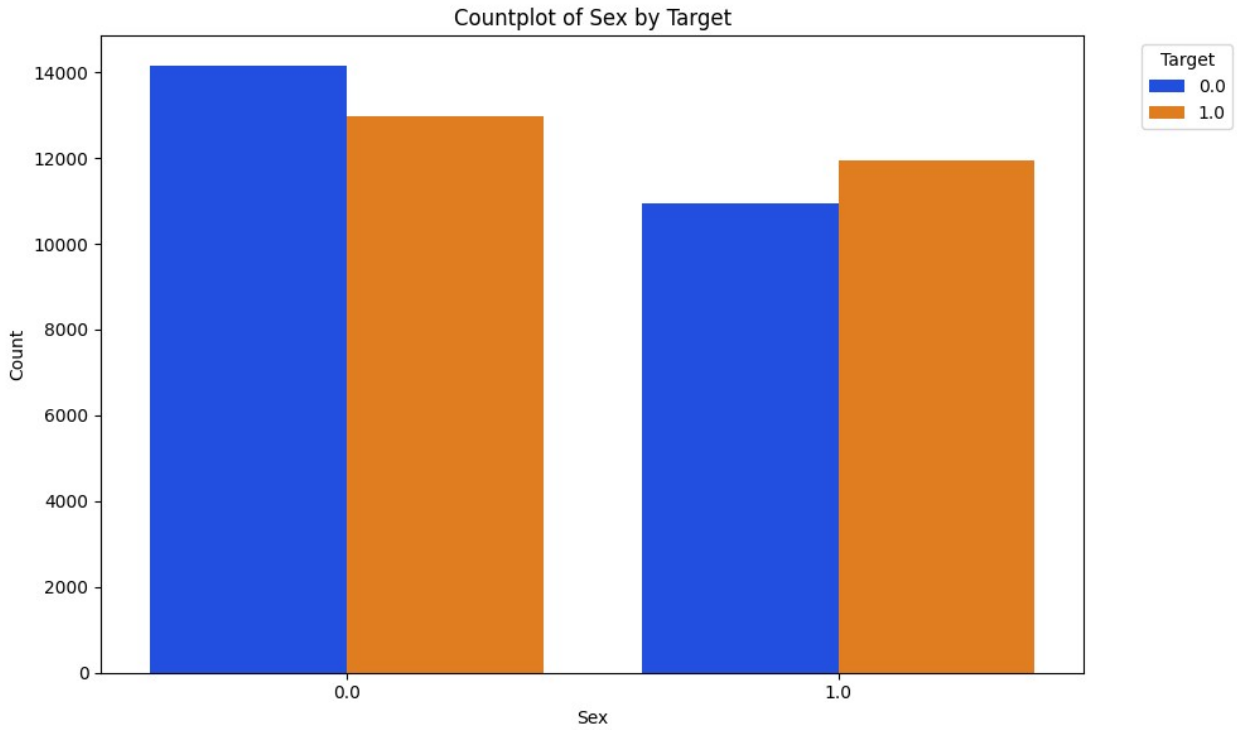

```
Generating countplot for: GenHlth
```



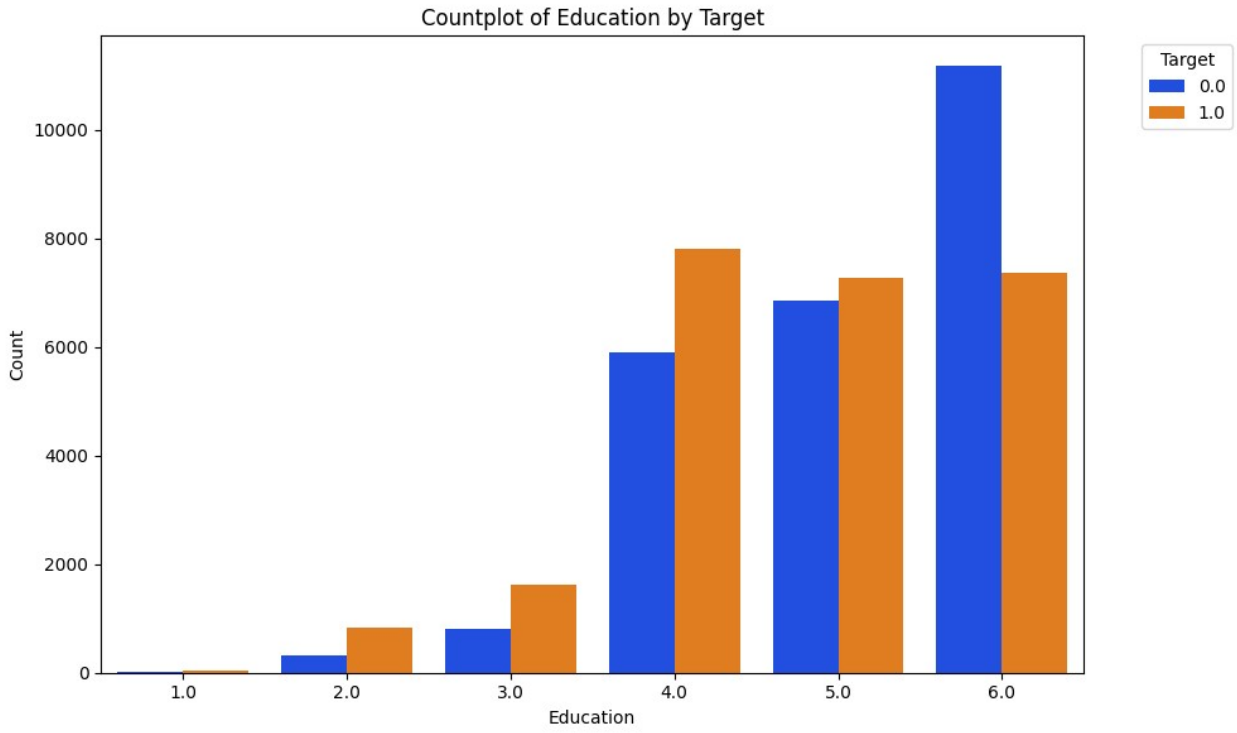

```
Generating countplot for: DiffWalk
```



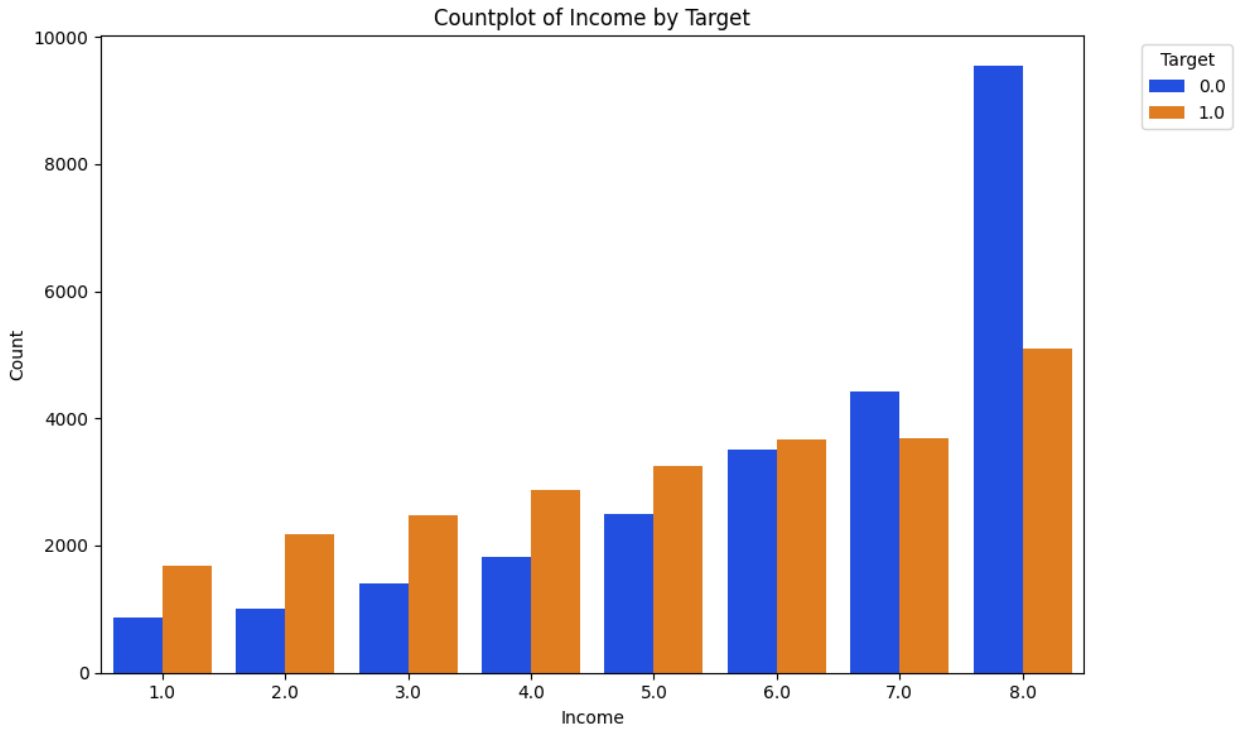

```
Generating countplot for: Sex
```




```
Generating countplot for: Education
```

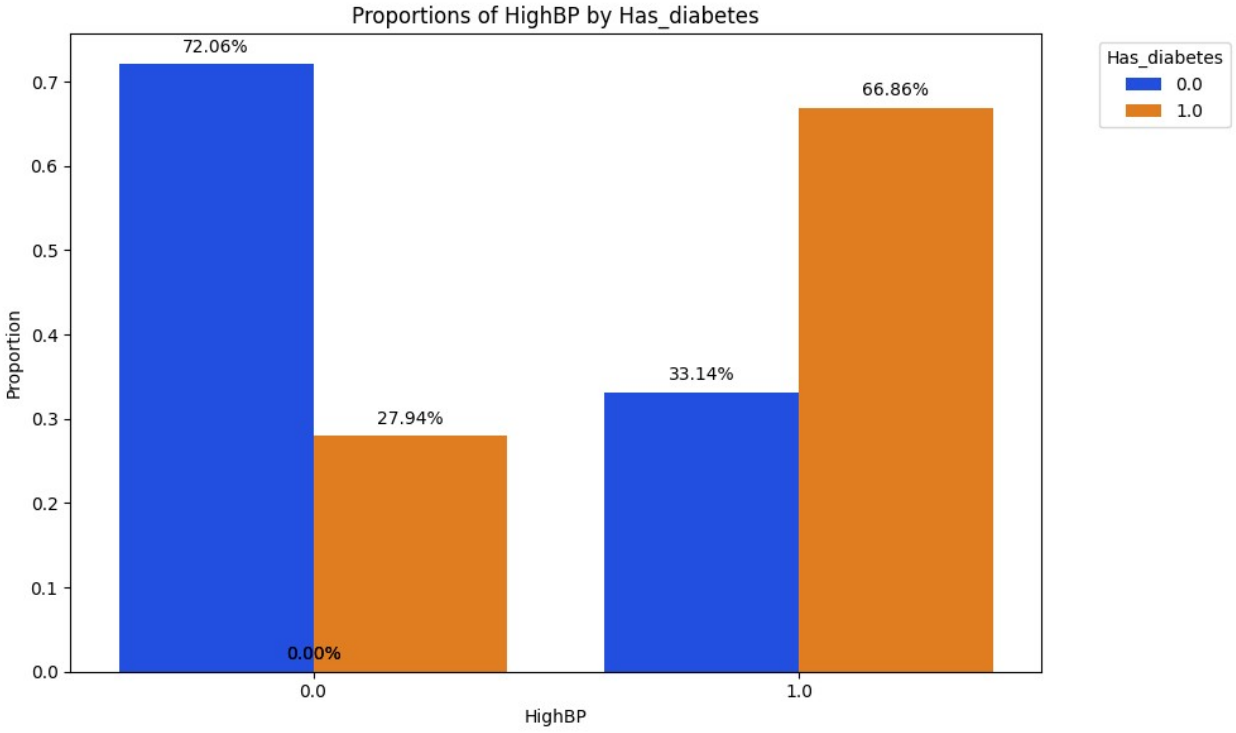


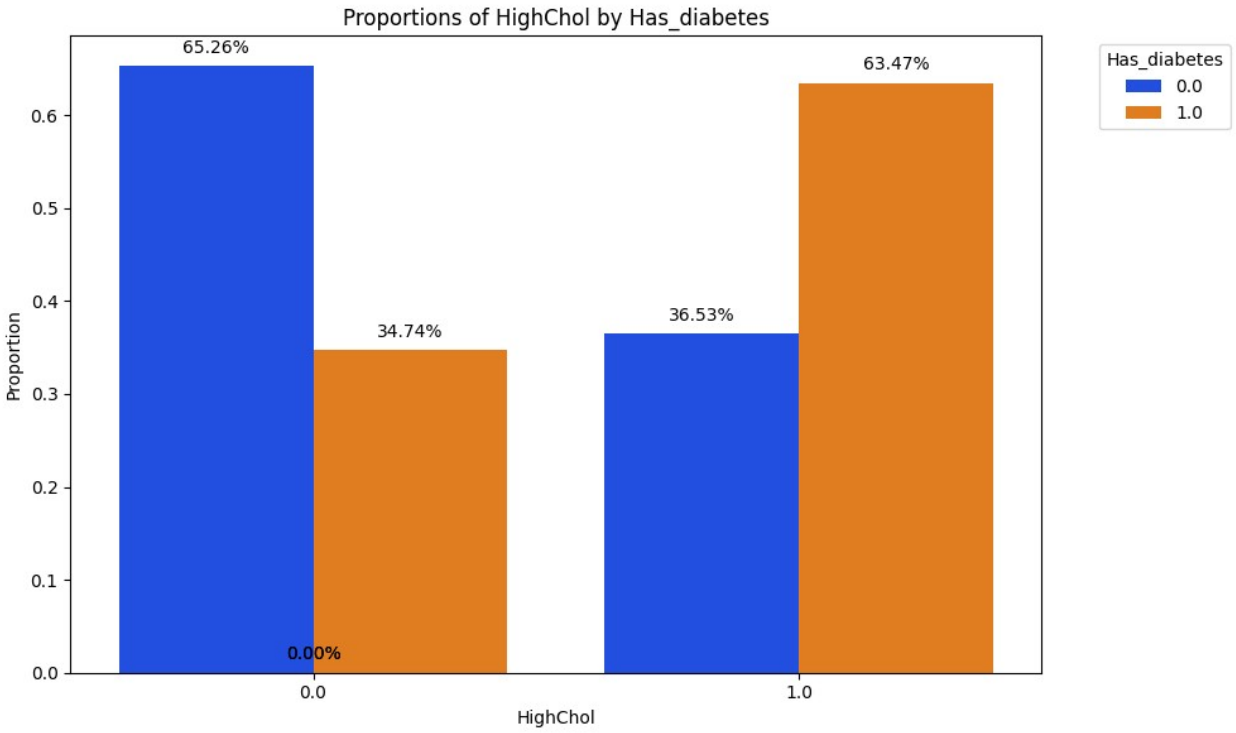

```
Generating countplot for: Income
```

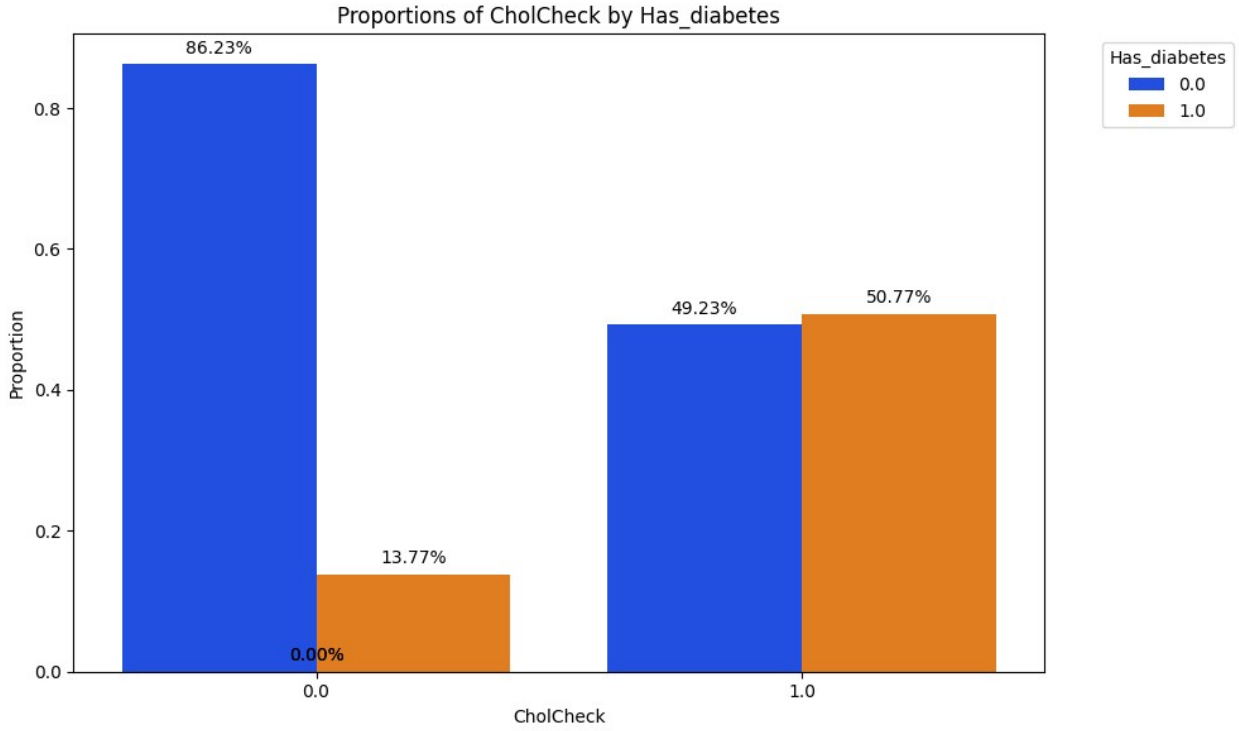


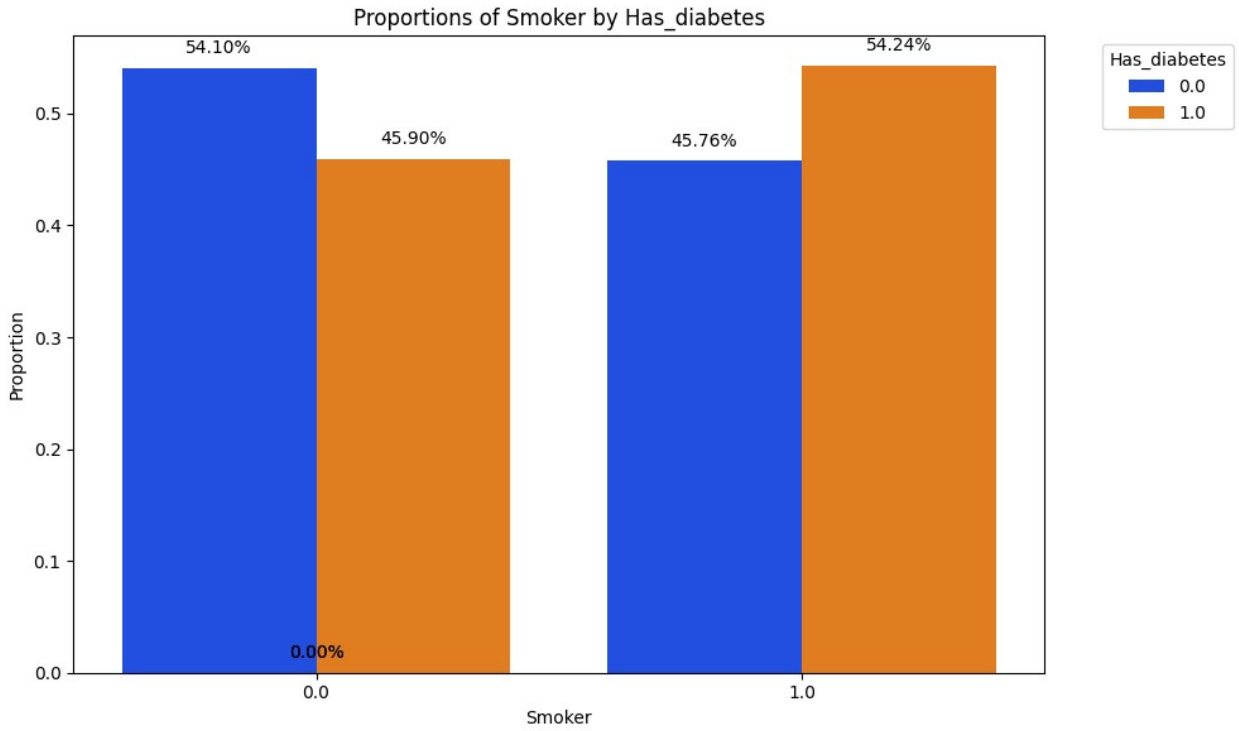

```
# Function to plot proportions of categorical variables
def plot_proportions(data, x, hue):
```

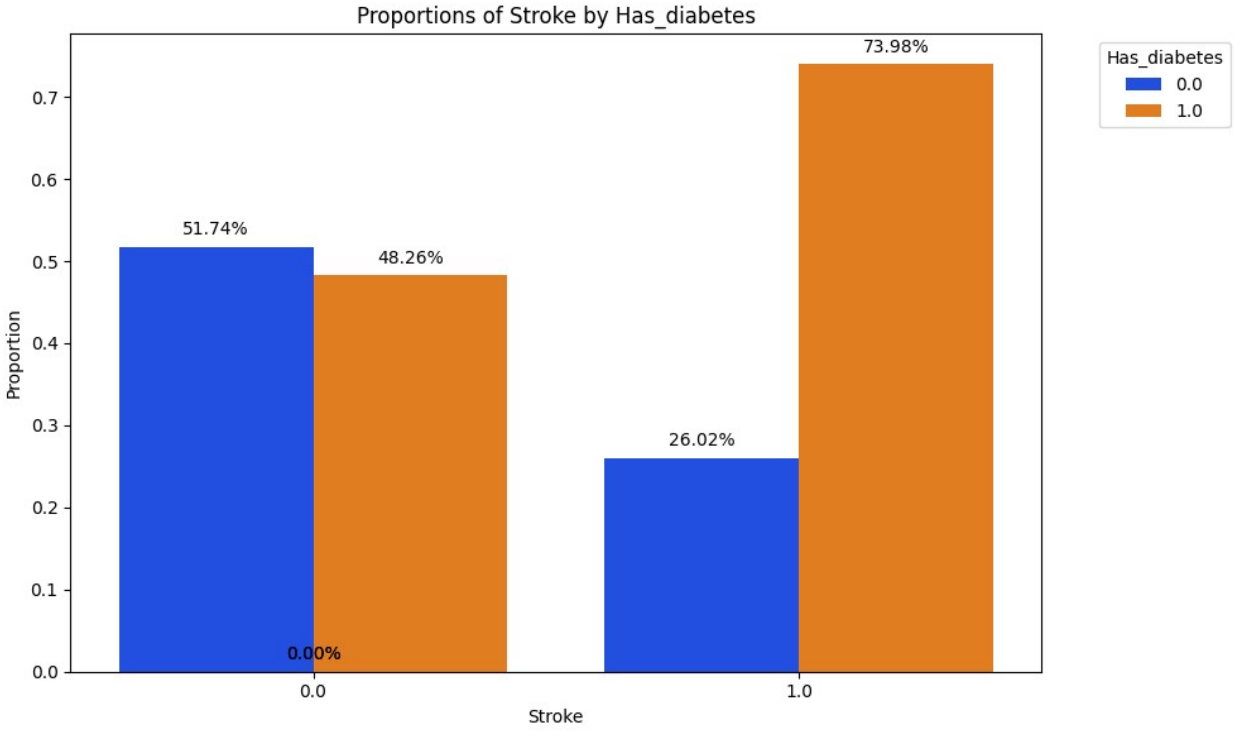
```
    # Calculate proportions
    counts = data.groupby([x, hue]).size().reset_index(name='count')
    total_counts = data.groupby(x).size().reset_index(name='total')

    # Merge counts with total_counts to calculate proportions
    merged = pd.merge(counts, total_counts, on=x)

    # Proportion is calculated within a group having the same target
variable label (0 or 1)
    merged['proportion'] = merged['count'] / merged['total']

    # Plot proportions
    plt.figure(figsize=(10, 6))

    sns.barplot(data=merged, x=x, y='proportion', hue=hue,
palette='bright')

    # Annotate with proportion values using the current axes
    for p in plt.gca().patches:
        height = p.get_height()
        plt.text(p.get_x() + p.get_width() / 2, height + 0.01,
f'{height:.2%}',
                 ha='center', va='bottom')

    # Proportions of catagories that has diabetes
    plt.title(f'Proportions of {x} by {hue}')
    plt.xlabel(x)
    plt.ylabel('Proportion')

    # Adjustment to the legend(key)
    plt.legend(title=hue, bbox_to_anchor=(1.05, 1), loc='upper left')

    # Display the plot using tight layout to adjust padding
    plt.tight_layout()
    plt.show()

# Apply function to each categorical column
for col in cat_cols:
    plot_proportions(df, col, 'Has_diabetes')
```

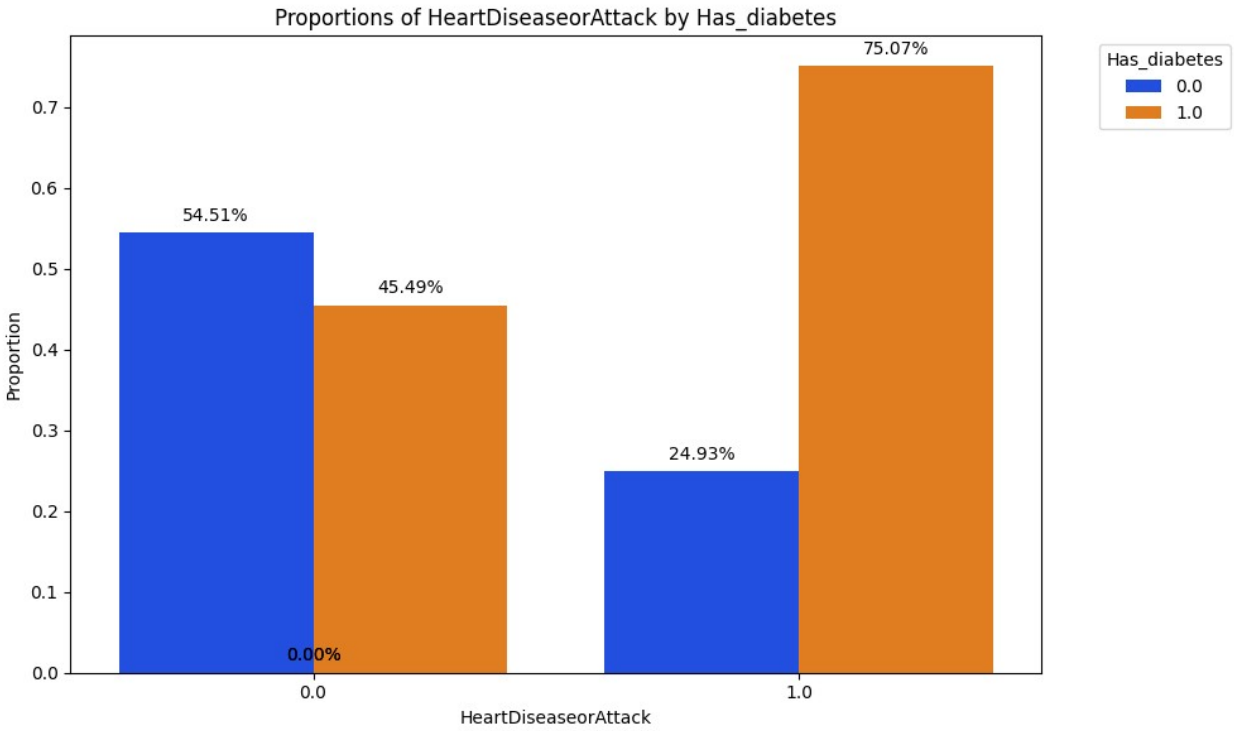

Proportions of HighBP by Has_diabetes
Has_diabetes
0.0
1.0
72.06%
27.94%
0.00%
33.14%
66.86%
Proportion
HighBP
0.0
1.0

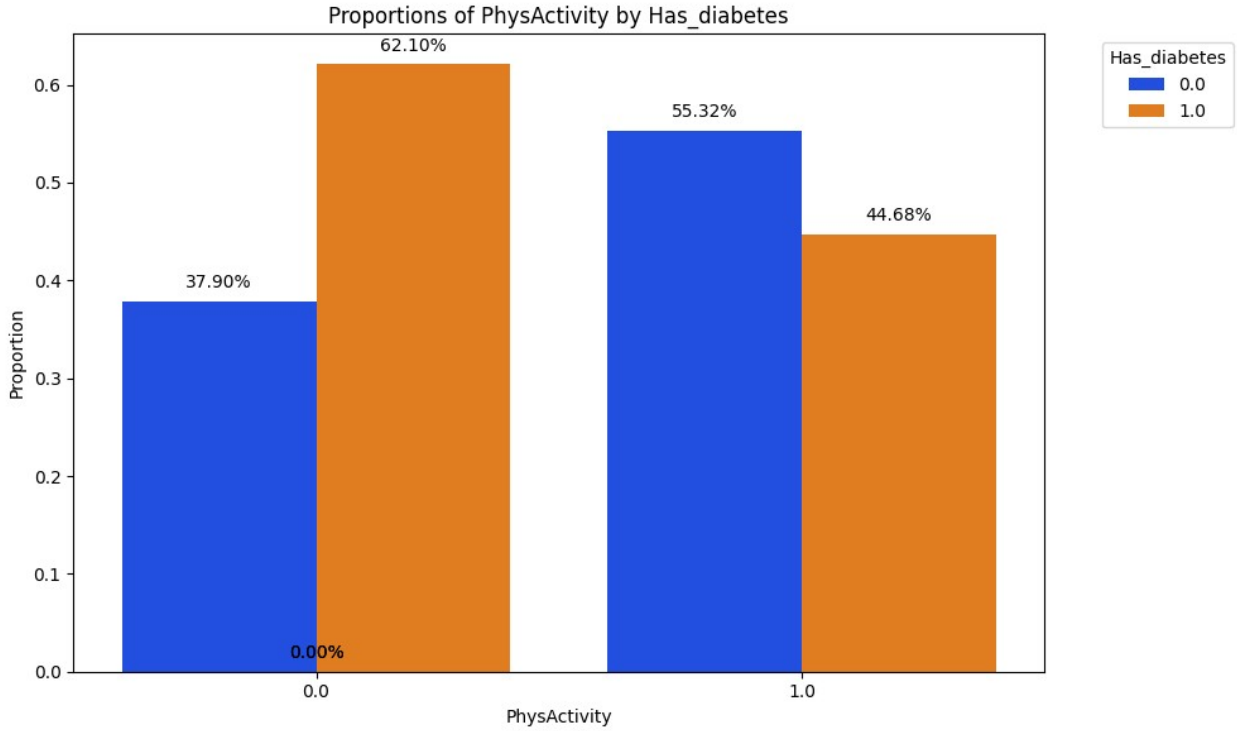

Proportions of HighChol by Has_diabetes
Has_diabetes
0.0
1.0
65.26%
34.74%
0.00%
36.53%
63.47%
Proportion
HighChol
0.0
1.0

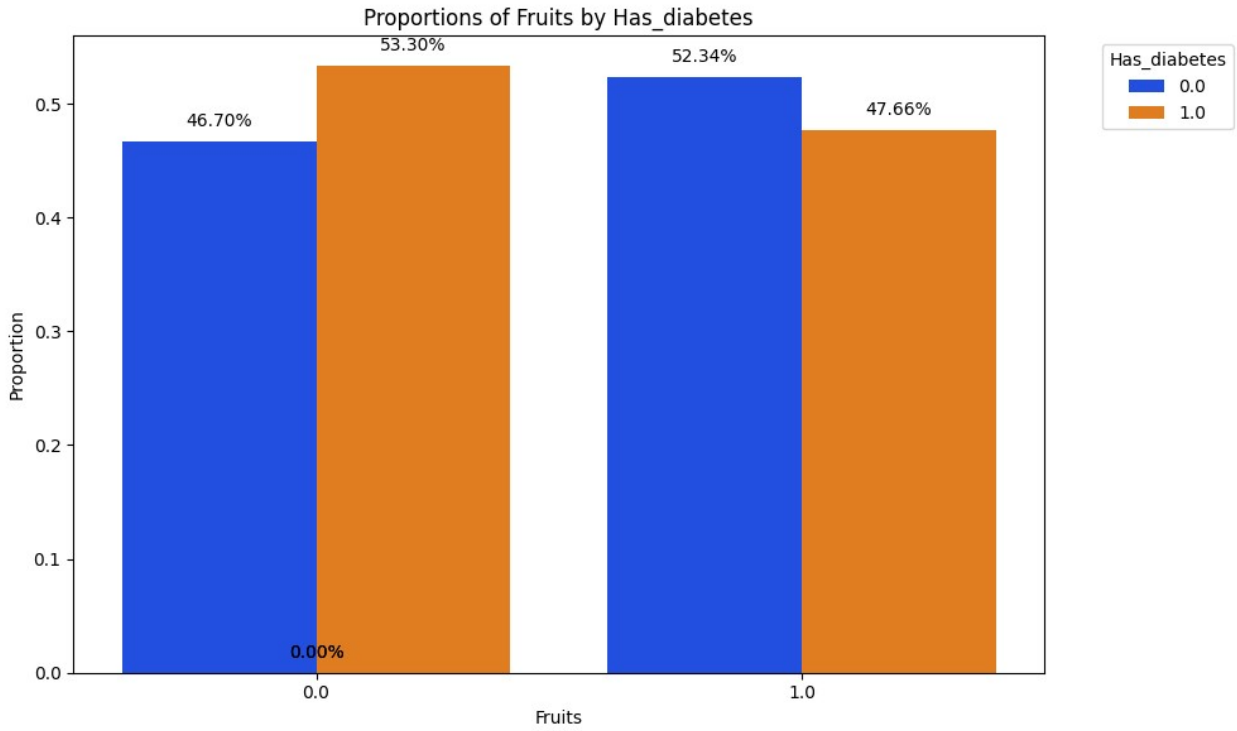

Proportions of CholCheck by Has_diabetes
Has_diabetes
0.0
1.0
86.23%
13.77%
0.00%
49.23%
50.77%
Proportion
0.0
0.2
0.4
0.6
0.8
0.0
1.0
CholCheck

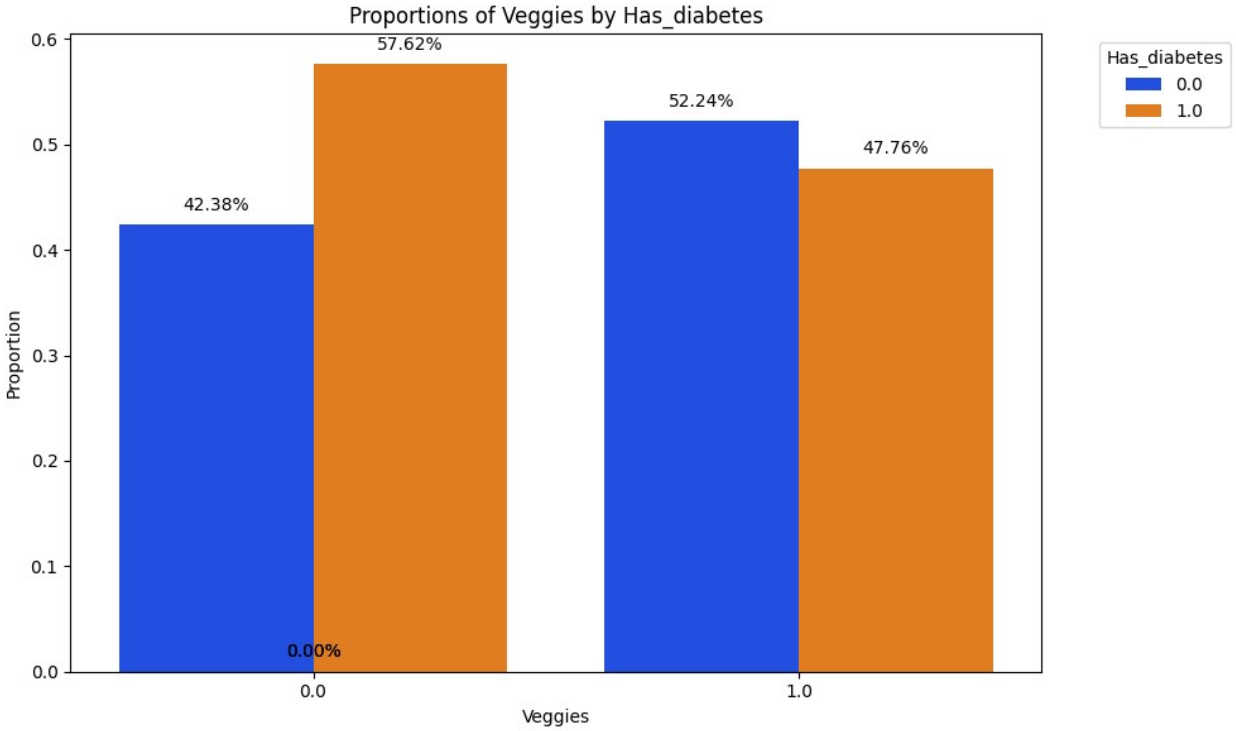

Proportions of Smoker by Has_diabetes
Has_diabetes
0.0
1.0
54.10%
45.90%
0.00%
45.76%
54.24%
Proportion
0.0
0.1
0.2
0.3
0.4
0.5
0.0
1.0
Smoker

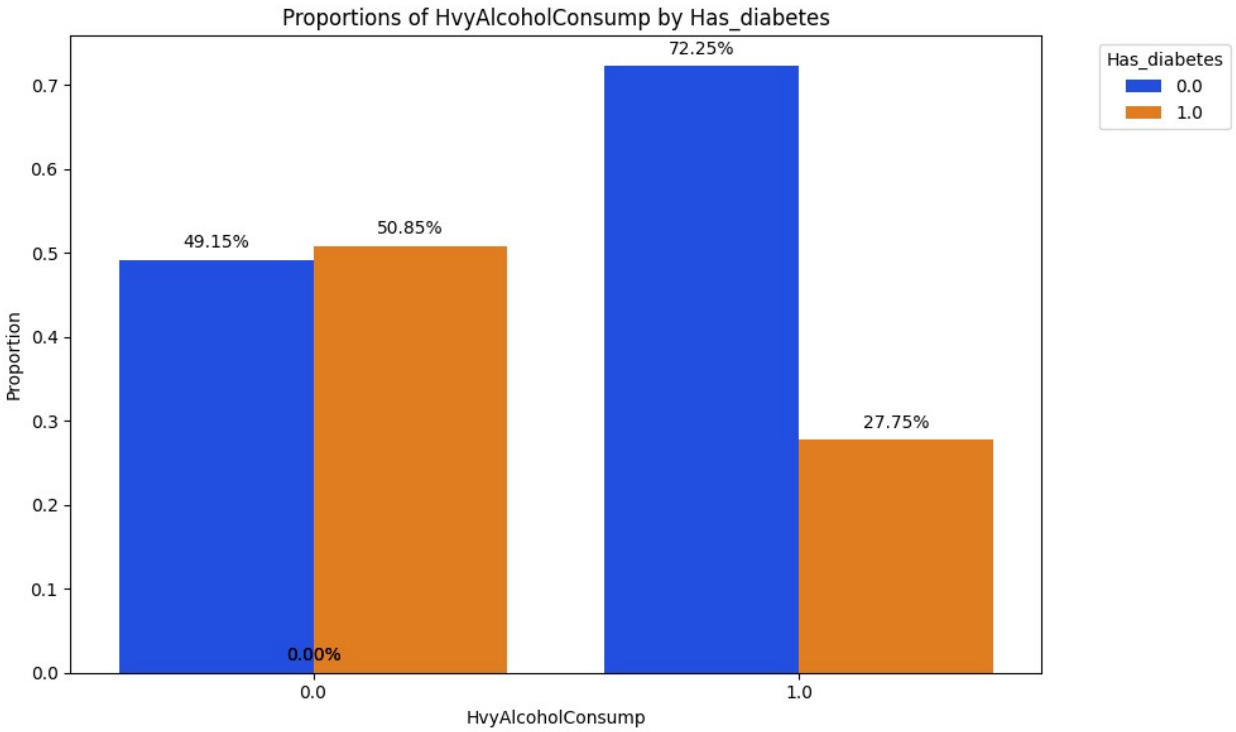

Proportions of Stroke by Has_diabetes
Has_diabetes
0.0
1.0
73.98%
51.74%
48.26%
26.02%
0.00%
Proportion
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.0
1.0
Stroke

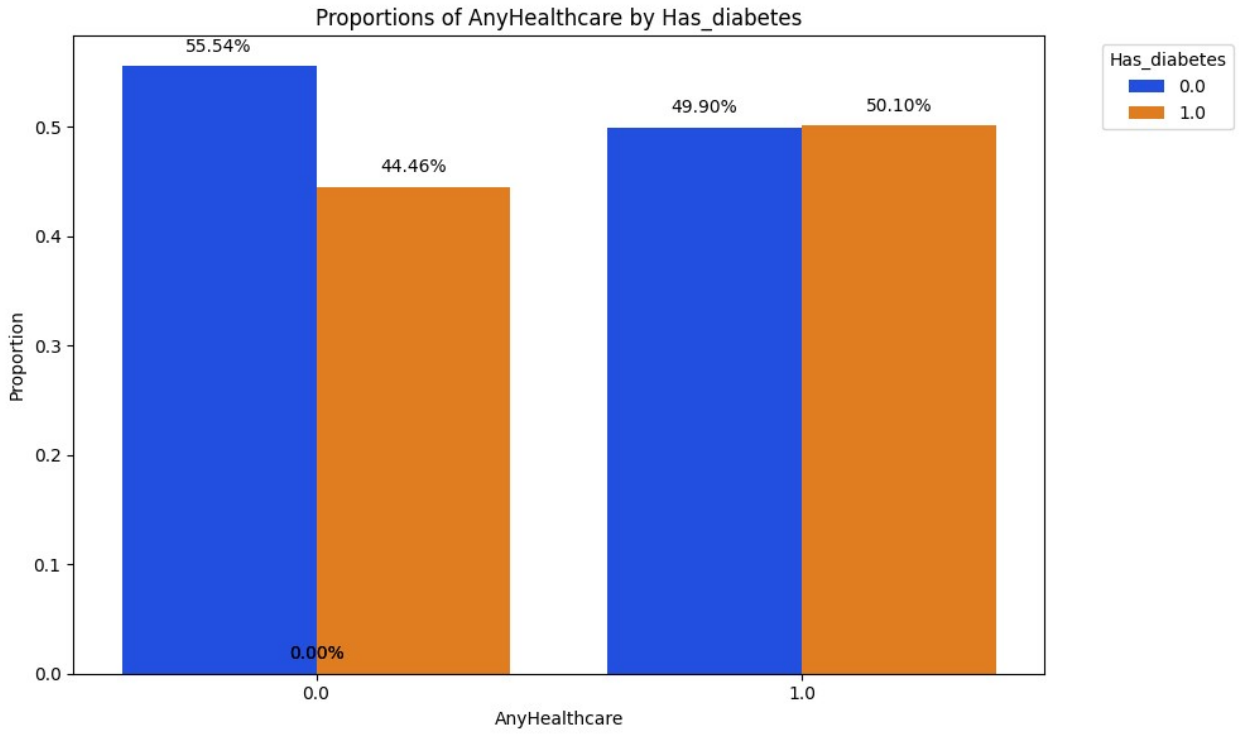

Proportions of HeartDiseaseorAttack by Has_diabetes
Has_diabetes
0.0
1.0
75.07%
54.51%
45.49%
24.93%
0.00%
Proportion
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.0
1.0
HeartDiseaseorAttack

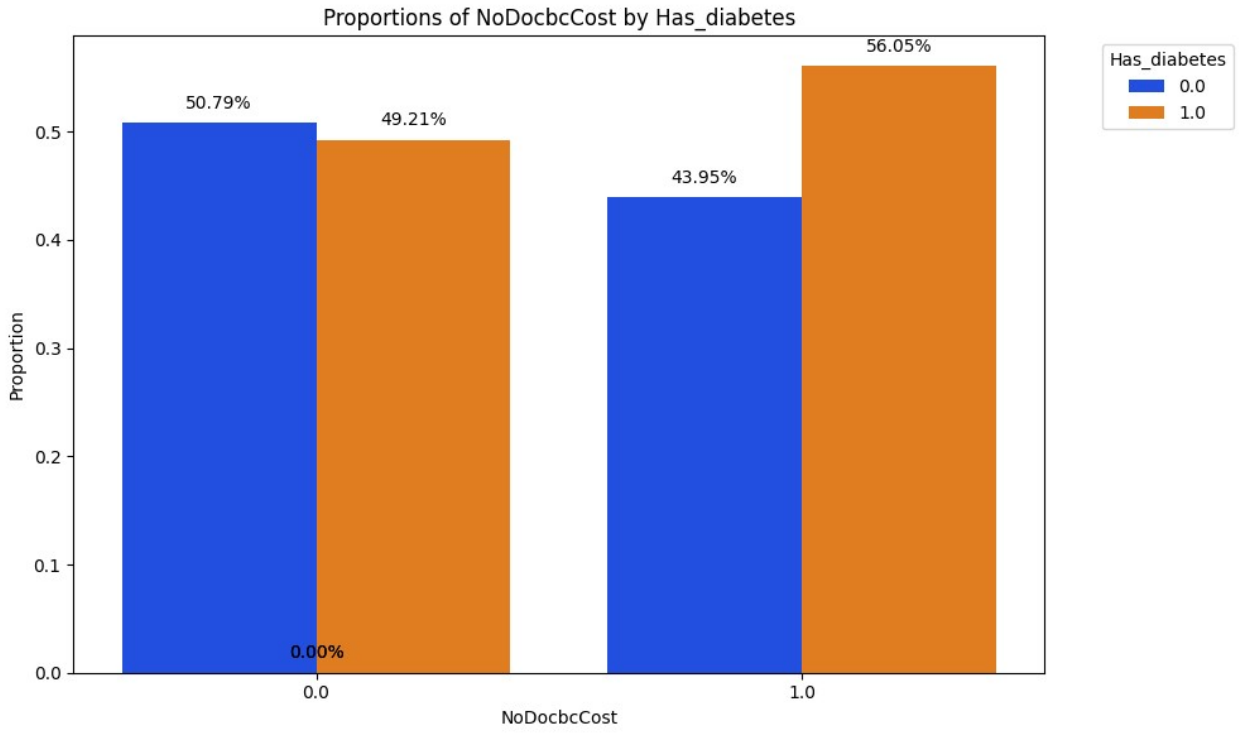

Proportions of PhysActivity by Has_diabetes
62.10%
55.32%
44.68%
37.90%
0.00%
Proportion
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.0
1.0
PhysActivity
Has_diabetes
0.0
1.0

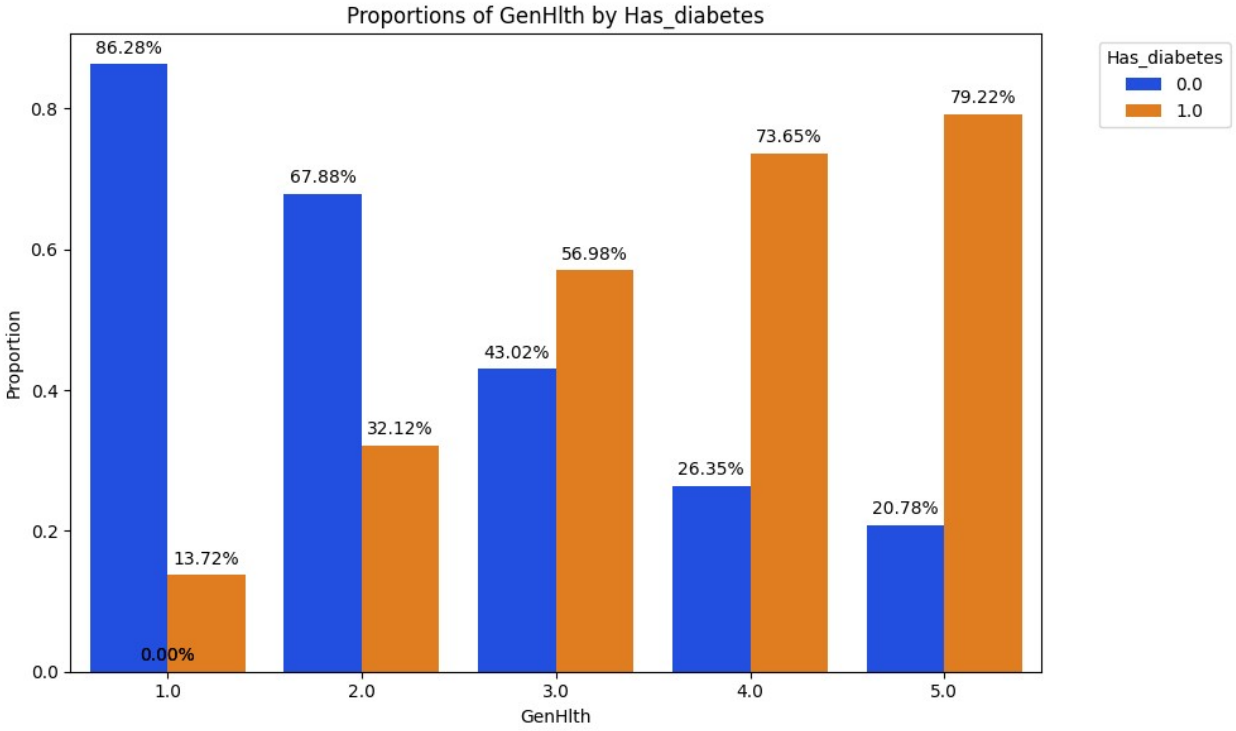

Proportions of Fruits by Has_diabetes
53.30%
52.34%
46.70%
47.66%
0.00%
Proportion
0.0
0.1
0.2
0.3
0.4
0.5
0.0
1.0
Fruits
Has_diabetes
0.0
1.0

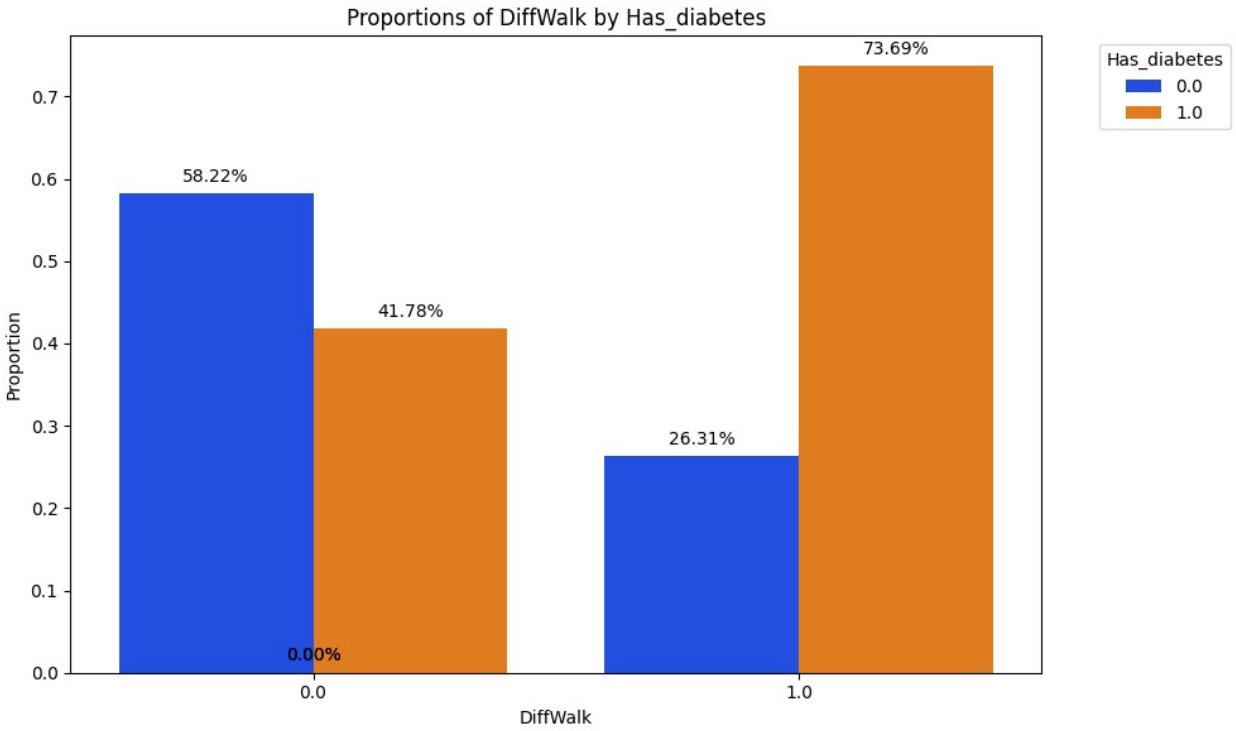

Proportions of Veggies by Has_diabetes
Has_diabetes
0.0
1.0
Proportion
Veggies
42.38%
57.62%
0.00%
52.24%
47.76%
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.0
1.0

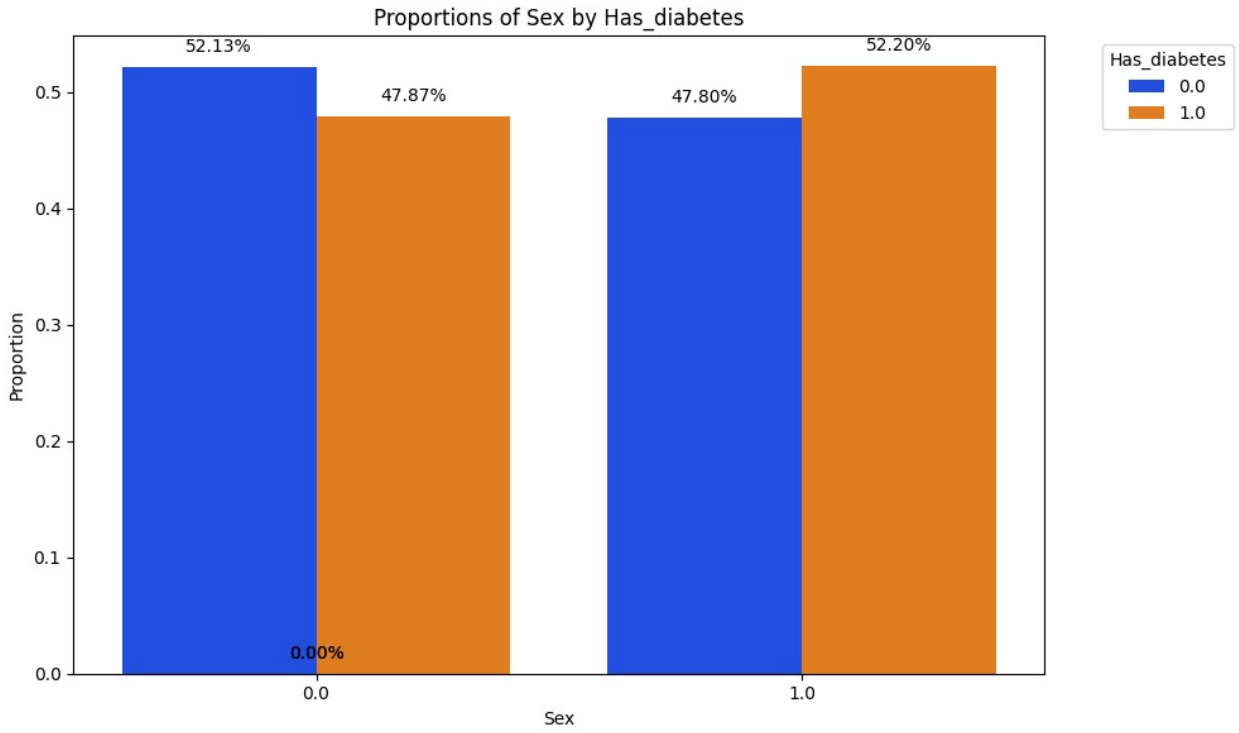

Proportions of HvyAlcoholConsump by Has_diabetes
Has_diabetes
0.0
1.0
Proportion
HvyAlcoholConsump
49.15%
50.85%
0.00%
72.25%
27.75%
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.0
1.0

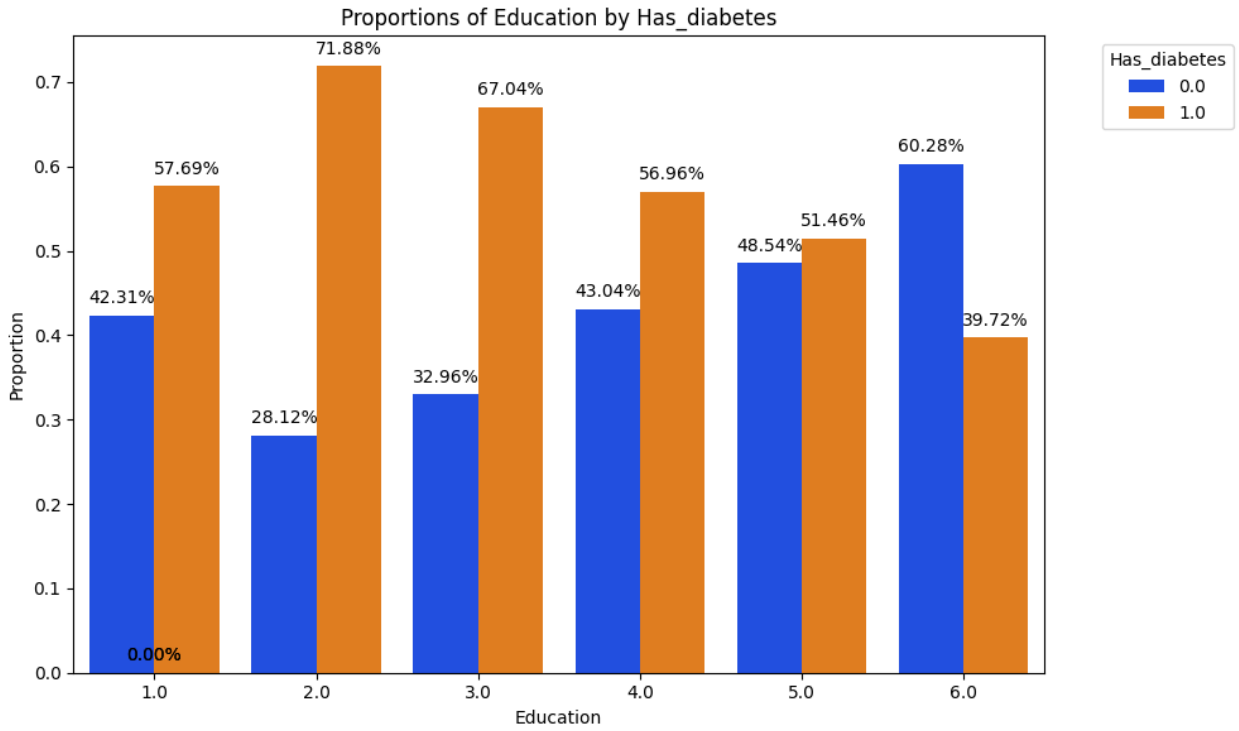

Proportions of AnyHealthcare by Has_diabetes
55.54%
44.46%
49.90%
50.10%
0.00%
Proportion
AnyHealthcare
0.0
1.0
Has_diabetes
0.0
1.0

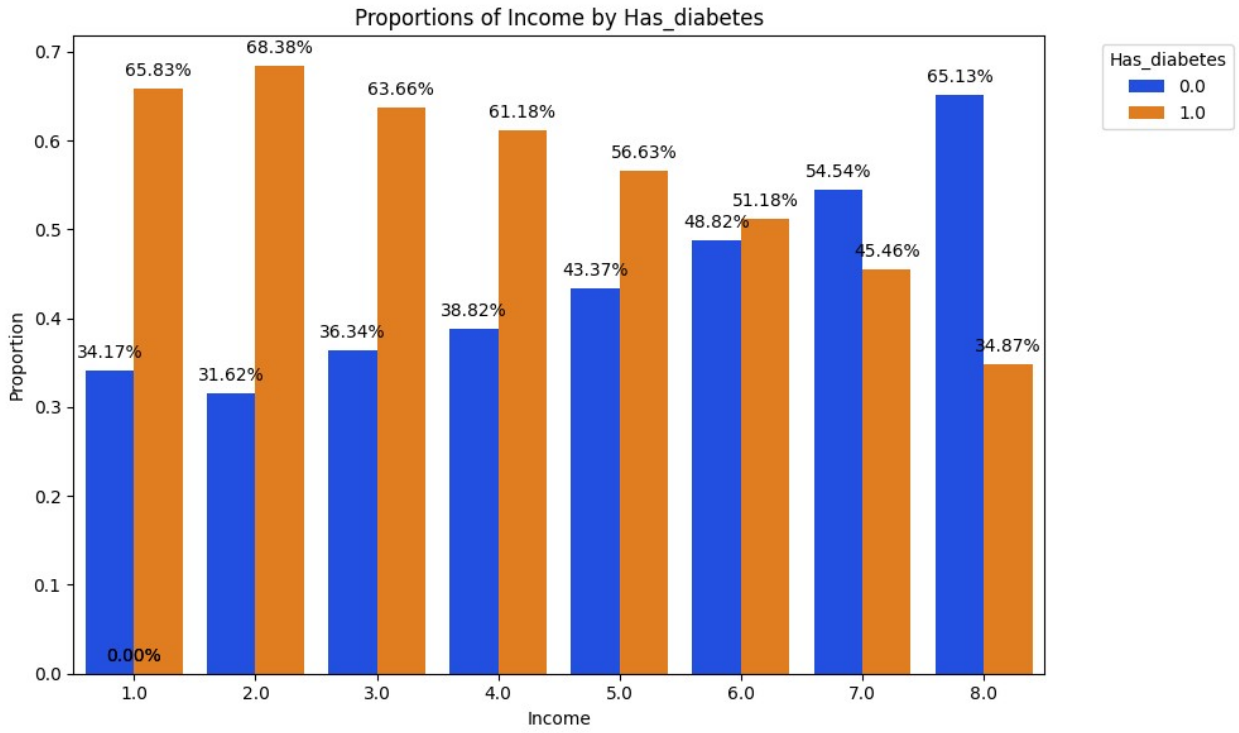

Proportions of NoDocbcCost by Has_diabetes
50.79%
49.21%
43.95%
56.05%
0.00%
Proportion
NoDocbcCost
0.0
1.0
Has_diabetes
0.0
1.0


Proportions of GenHlth by Has_diabetes
Has_diabetes
0.0
1.0
86.28%
13.72%
67.88%
32.12%
43.02%
56.98%
26.35%
73.65%
20.78%
79.22%
0.00%
Proportion
GenHlth
1.0
2.0
3.0
4.0
5.0

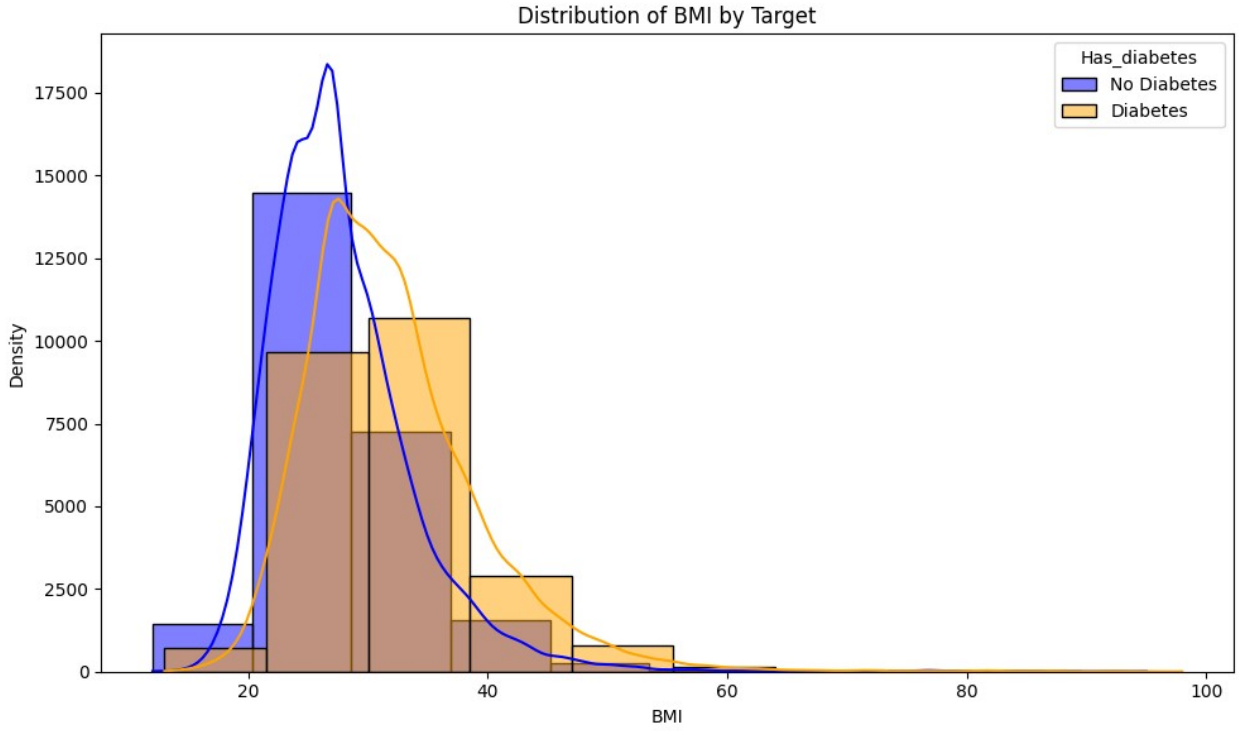

Proportions of DiffWalk by Has_diabetes
Has_diabetes
0.0
1.0
58.22%
41.78%
26.31%
73.69%
0.00%
Proportion
DiffWalk
0.0
1.0

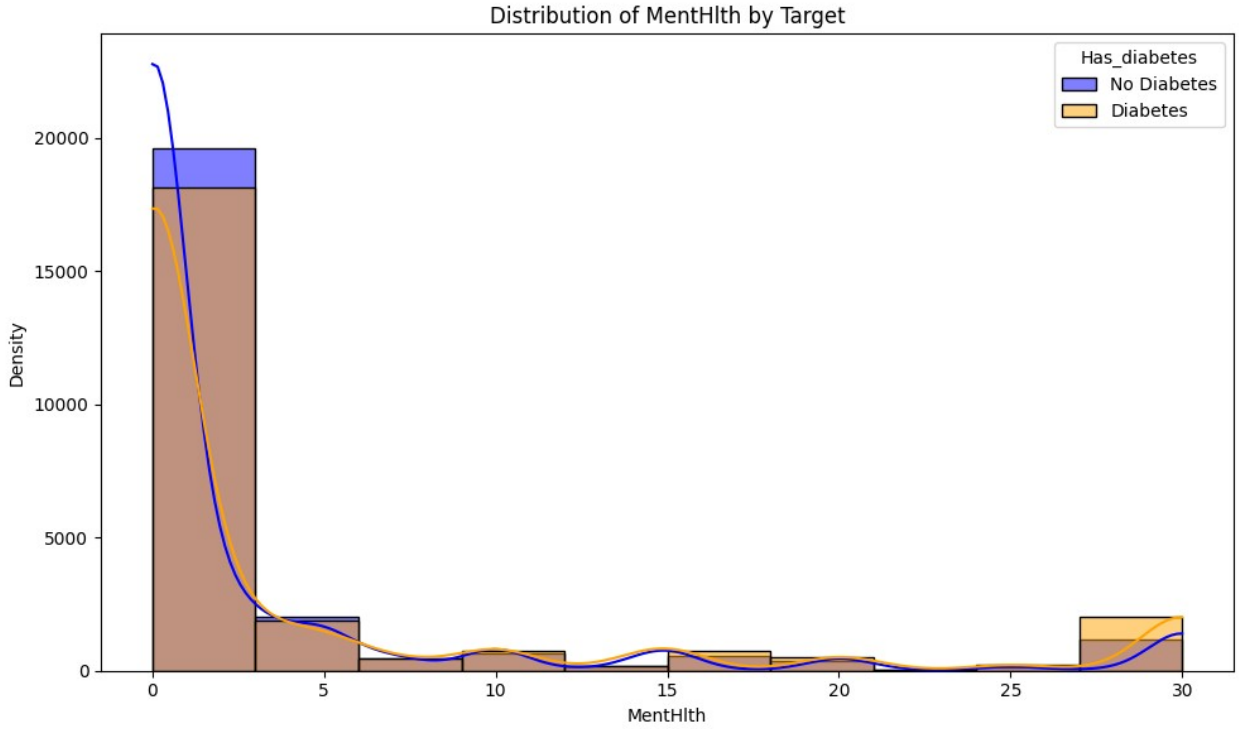

Proportions of Sex by Has_diabetes
Has_diabetes
0.0
1.0
52.13%
47.87%
47.80%
52.20%
0.00%
Proportion
Sex
0.0
0.1
0.2
0.3
0.4
0.5
0.0
1.0

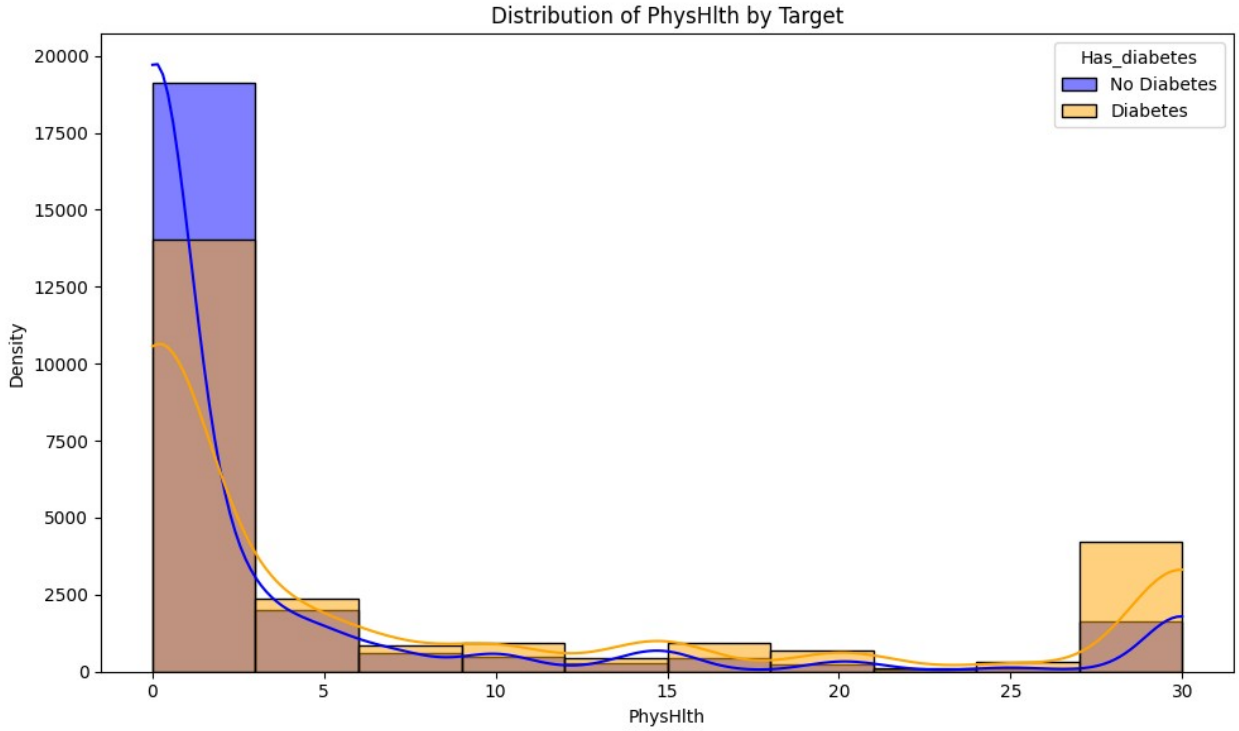

Proportions of Education by Has_diabetes
Has_diabetes
0.0
1.0
42.31%
57.69%
28.12%
71.88%
32.96%
67.04%
43.04%
56.96%
48.54%
51.46%
60.28%
39.72%
0.00%
Proportion
Education
1.0
2.0
3.0
4.0
5.0
6.0

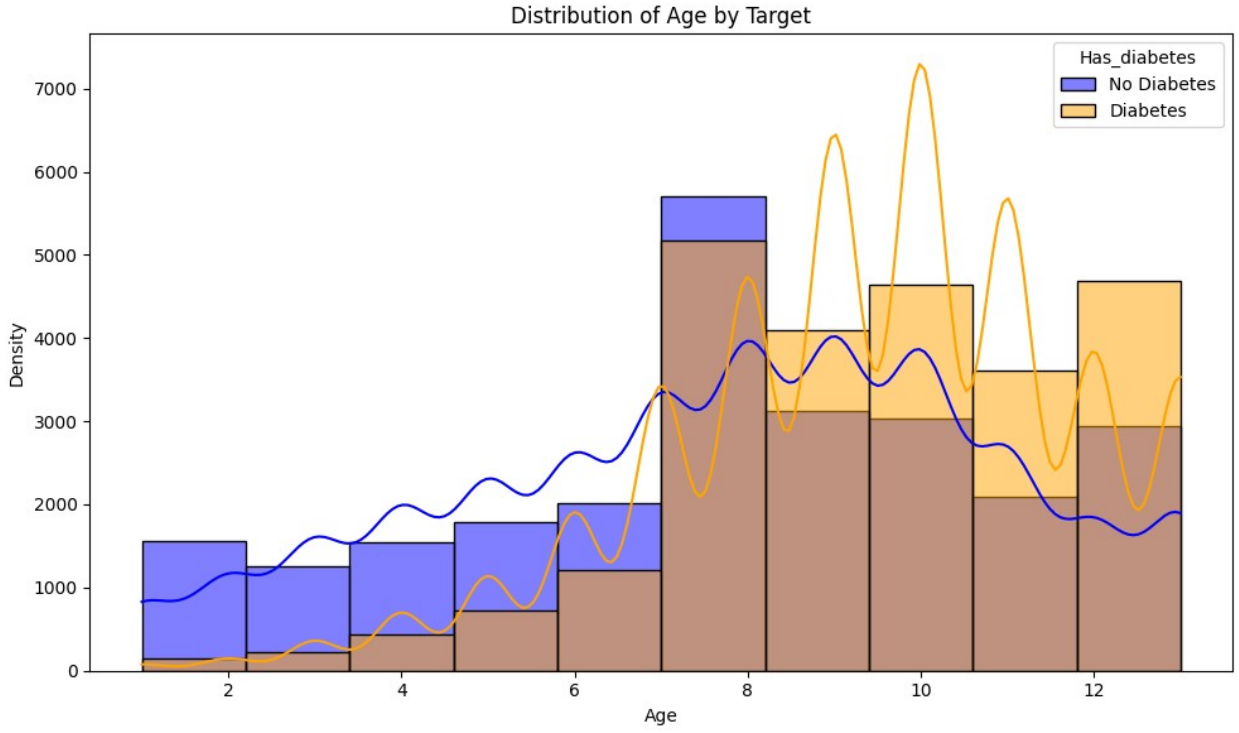



```
# Loop to create and display distribution plots
for col in num_cols:
    plt.figure(figsize=(10, 6))

    # Plot histogram with KDE for 'Has_diabetes' = 0
    sns.histplot(data=df[df['Has_diabetes'] == 0], x=col, kde=True,
color='blue', label='No Diabetes', bins=10)

    # Plot histogram with KDE for 'Has_diabetes' = 1
    sns.histplot(data=df[df['Has_diabetes'] == 1], x=col, kde=True,
color='orange', label='Diabetes', bins=10)

    # Set plot title and labels
    plt.title(f'Distribution of {col} by Target')
    plt.xlabel(col)
    plt.ylabel('Density')

    # Display the legend
    plt.legend(title='Has_diabetes', loc='upper right')

    # Display the plot
    plt.tight_layout()  # Adjust layout to prevent overlap
    plt.show()
```

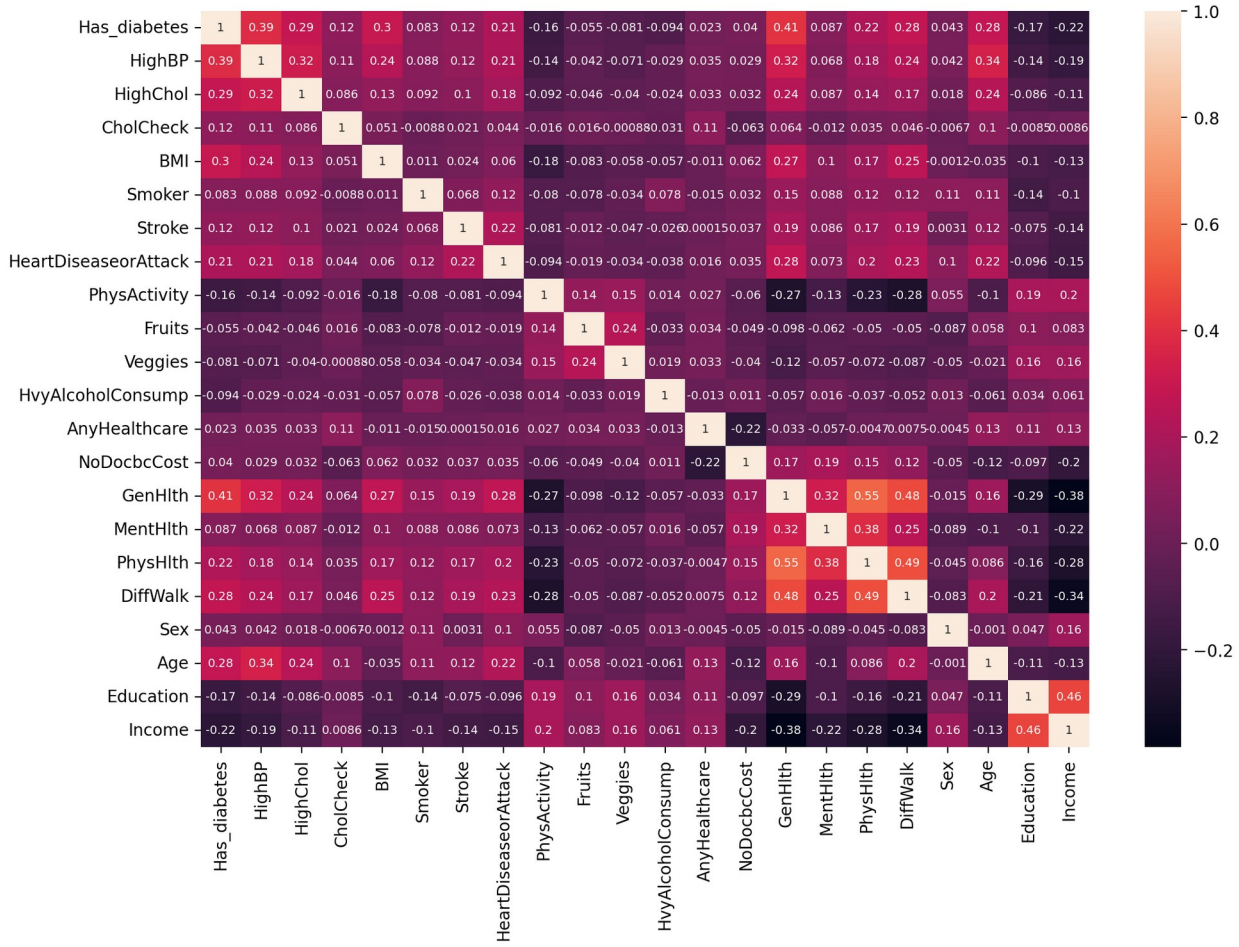

Distribution of BMI by Target
Has_diabetes
No Diabetes
Diabetes
Density
BMI
0
2500
5000
7500
10000
12500
15000
17500
20
40
60
80
100


Distribution of MentHlth by Target
Has_diabetes
No Diabetes
Diabetes
Density
MentHlth
0
5000
10000
15000
20000
0
5
10
15
20
25
30






```
# Heatmap
plt.figure(figsize=(12,8), dpi=200)
sns.heatmap(df.corr(), annot=True, annot_kws={"size" : 7});
```




```
# Features and target
X = df.drop(columns=['Has_diabetes'])  # 21 features - drops
'Has_diabetes'
y = df['Has_diabetes']  # Target variable

# Handling missing values with mean imputation
imputer = SimpleImputer(strategy='mean')
df[df.columns] = imputer.fit_transform(df)

# Splitting data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42, stratify=y)

# Standardizing the features using imputer
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Decision Tree Classifier

```
# Parameter Grid
param_grid_dt = {
    'max_depth': [5, 10, 20, None],
    'min_samples_split': [2, 10, 20],
    'min_samples_leaf': [1, 5, 10],
    'criterion': ['gini', 'entropy'],
    'splitter': ['best', 'random']
}

# Model using a random state
dt = DecisionTreeClassifier(random_state=42)

# Grid Search with "5-fold cross-validation"
grid_dt = GridSearchCV(dt, dt_param_grid, cv=5, scoring='accuracy',
return_train_score=True, n_jobs=-1, verbose=2)
grid_dt.fit(X_train_scaled, y_train)
```

```
Fitting 5 folds for each of 54 candidates, totalling 270 fits
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=10; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=10; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=10; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=10; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=20; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=20; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=20; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=20; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=1,
min_samples_split=20; total time=   0.0s
```

```
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=10; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=10; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=10; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=10; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=20; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=20; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=20; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=2,
min_samples_split=20; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
min_samples_split=10; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
min_samples_split=10; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
min_samples_split=10; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
min_samples_split=20; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
```

```
min_samples_split=10; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
min_samples_split=10; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
min_samples_split=20; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
min_samples_split=20; total time=   0.0s
[CV] END criterion=gini, max_depth=5, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
```

```
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
```

```
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
```

```
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
min_samples_split=2; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
min_samples_split=10; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
min_samples_split=10; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
```

```
min_samples_split=2; total time=   0.0s
[CV] END criterion=gini, max_depth=10, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=gini, max_depth=20, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
min_samples_split=2; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
min_samples_split=2; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
min_samples_split=10; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
min_samples_split=10; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
min_samples_split=20; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
min_samples_split=20; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
min_samples_split=20; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=2; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=2; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=2; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=10; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
min_samples_split=20; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=10; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=1,
min_samples_split=20; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=2; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=2; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=10; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=10; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=10; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=20; total time=   0.0s
```

```
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=20; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=20; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=2; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=2; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=10; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=2,
min_samples_split=20; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=2; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=10; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=2; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=2; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=10; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=10; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=20; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=10; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=20; total time=   0.0s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=20; total time=   0.0s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=20; total time=   0.0s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=5, min_samples_leaf=5,
min_samples_split=20; total time=   0.0s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
```

```
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
```

```
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=10, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
```

```
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=1,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=2; total time=   0.2s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=10; total time=   0.2s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=2,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
min_samples_split=2; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
```

```
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
min_samples_split=10; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
[CV] END criterion=entropy, max_depth=20, min_samples_leaf=5,
min_samples_split=20; total time=   0.1s
```

```
GridSearchCV(cv=5, estimator=DecisionTreeClassifier(random_state=42),
n_jobs=-1,
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [5, 10, 20],
                         'min_samples_leaf': [1, 2, 5],
                         'min_samples_split': [2, 10, 20]},
             return_train_score=True, scoring='accuracy', verbose=2)
```

```
# Best Decision Tree model and its parameters
best_dt_model = grid_dt.best_estimator_
best_dt_model.fit(X_train_scaled, y_train)

# Best Params and Model Accuracy
print(f"Best Decision Tree parameters: {grid_dt.best_params_}")
print(f"Best Decision Tree Cross-Validation Accuracy:
{grid_dt.best_score_:.4f}")
```

```
Best Decision Tree parameters: {'criterion': 'entropy', 'max_depth':
10, 'min_samples_leaf': 5, 'min_samples_split': 2}
Best Decision Tree Cross-Validation Accuracy: 0.7329
```

```
# Cross-validation mean and standard deviation for Decision Tree
dt_cv_results = cross_val_score(best_dt_model, X_train_scaled,
y_train, cv=5, scoring='accuracy')

# Mean and Standard deviation
print(f"Decision Tree Mean Accuracy: {dt_cv_results.mean():.4f}")
print(f"Decision Tree Accuracy Standard Deviation:
{dt_cv_results.std():.4f}")
```

```
Decision Tree Mean Accuracy: 0.7329
Decision Tree Accuracy Standard Deviation: 0.0057
```

```
# Evaluate the best Decision Tree model on the test set
y_pred_dt = best_dt_model.predict(X_test_scaled)
accuracy_dt = accuracy_score(y_test, y_pred_dt)
print(f"Decision Tree Test Accuracy: {accuracy_dt:.4f}")
```

```
Decision Tree Test Accuracy: 0.7357
```

```
# Classification Report and Confusion Matrix for Decision Tree
print("Decision Tree Classification Report:")
print(classification_report(y_test, y_pred_dt))
```

```
Decision Tree Classification Report:
              precision    recall  f1-score   support

         0.0       0.75      0.70      0.73      5015
         1.0       0.72      0.77      0.74      4985

    accuracy                           0.74     10000
   macro avg       0.74      0.74      0.74     10000
weighted avg       0.74      0.74      0.74     10000
```

```
# Save the best DecisionTree model to a pickle file
with open('best_dt_model.pkl', 'wb') as file:
    pickle.dump(best_dt_model, file)
```

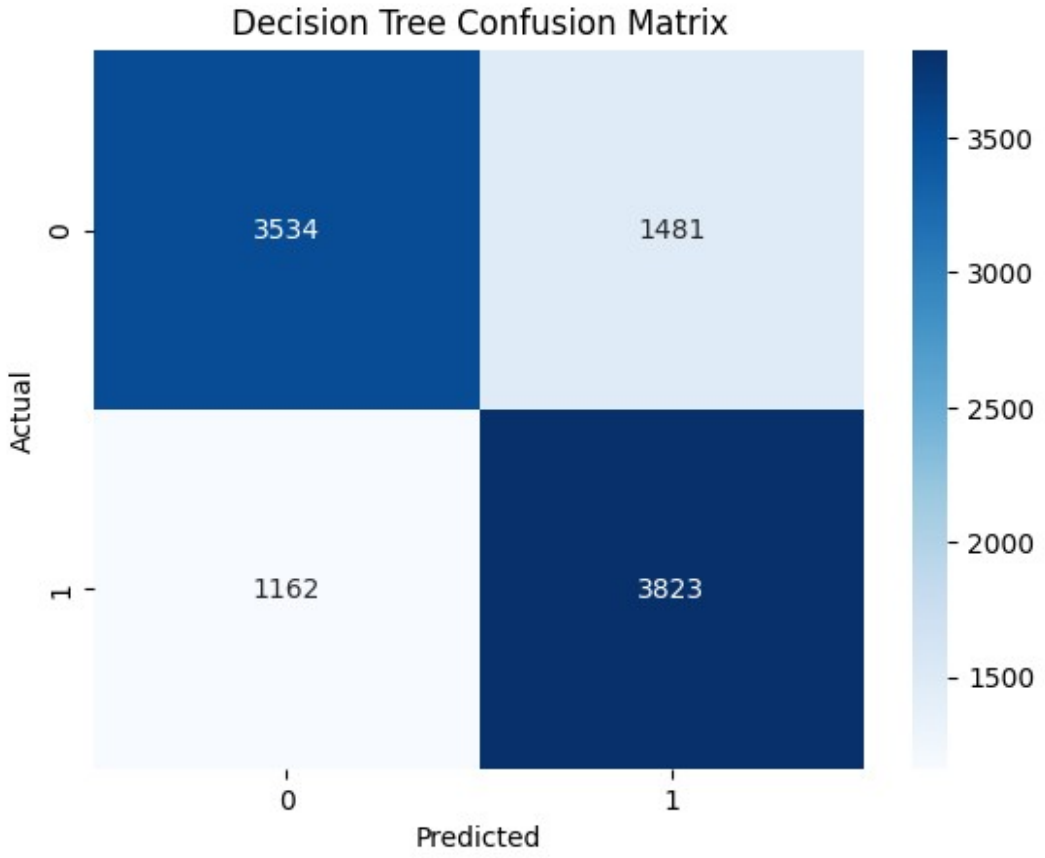
```
# Evaluating dt the model using scaled test data
print("Evaluating model on test data...")
model_evaluate(best_dt_model,X_test_scaled,y_test)
```

```
Evaluating model on test data...
Evaluating model performance on test data
Accuracy: 0.7357

Classification Report:
               precision    recall  f1-score   support

         0.0       0.75      0.70      0.73      5015
         1.0       0.72      0.77      0.74      4985

    accuracy                           0.74     10000
   macro avg       0.74      0.74      0.74     10000
weighted avg       0.74      0.74      0.74     10000
```

```
# Decision Tree Confusion Matrix
print("Decision Tree Confusion Matrix:")
conf_matrix_dt = confusion_matrix(y_test, y_pred_dt)
```

```
sns.heatmap(conf_matrix_dt, annot=True, fmt='d', cmap='Blues')
plt.title('Decision Tree Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

```
Decision Tree Confusion Matrix:
```



Random Forest Classifier

```
# Parameter grid for Random Forest
rf_param_grid = {
    'max_depth': [10, 20, 50],
    'n_estimators': [100, 200, 500],
    'criterion': ['gini','entropy'],
    'bootstrap': [True, False],
    'max_features': ['sqrt','log2'],
    'class_weight': ['balanced'],
    'n_jobs': [-1],
    'random_state': [42]
}
```

```
rf = RandomForestClassifier(random_state=42)


# Grid Search with 5-fold cross-validation
grid_rf = GridSearchCV(rf, rf_param_grid, cv=5, scoring='accuracy',
return_train_score=True, n_jobs=-1, verbose=2)
grid_rf.fit(X_train_scaled, y_train)
```

```
Fitting 5 folds for each of 72 candidates, totalling 360 fits
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   2.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   2.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   2.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   2.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.0s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
```

```
random_state=42; total time=   1.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   2.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   2.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   2.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   2.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.0s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.0s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.9s
```

```
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  11.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
```

```
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  11.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  12.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  12.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  12.0s

/opt/homebrew/Cellar/jupyterlab/4.2.5_1/libexec/lib/python3.12/site-
packages/joblib/externals/loky/process_executor.py:752: UserWarning: A
worker stopped while some jobs were given to the executor. This can be
caused by a too short worker timeout or by a memory leak.
  warnings.warn(

[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
```

```
random_state=42; total time=  11.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  11.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  11.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  12.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  12.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.0s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.0s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.0s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.6s
```

```
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  12.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.0s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.0s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
```

```
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   2.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.0s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.0s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  12.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   6.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   6.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   6.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   2.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
```

```
random_state=42; total time=   2.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   2.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   6.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.0s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   2.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   2.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.0s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   7.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.4s
```

```
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  11.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  12.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  12.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  12.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
```

```
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  11.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  11.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  11.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  11.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  12.0s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  12.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.3s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.4s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
```

```
random_state=42; total time=   4.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.0s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.6s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  12.8s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.5s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.2s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.9s
```

```
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.7s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.6s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  12.7s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.9s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.1s
[CV] END bootstrap=True, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
```

```
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.0s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.7s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.6s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   8.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.1s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.6s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.6s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
```

```
random_state=42; total time=   3.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  10.1s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  10.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.7s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  10.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
```

```
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.1s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  16.0s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  16.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  16.6s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  16.0s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  15.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   5.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.7s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.7s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
```

```
random_state=42; total time=  15.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   7.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  15.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  15.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  16.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   7.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  16.0s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.7s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.6s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   7.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  17.7s
```

```
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  17.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  17.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   7.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  18.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  18.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.0s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.0s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.0s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
```

```
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.1s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   4.1s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  18.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  18.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  18.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  18.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=gini,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  18.6s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.1s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  10.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.0s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   2.1s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.7s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   1.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
```

```
random_state=42; total time=   3.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   3.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.7s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.1s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.1s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.6s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=10, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=   9.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.7s
```

```
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   7.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  17.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  17.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  17.6s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   7.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   7.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  17.7s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
```

```
max_depth=20, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  17.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   4.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   4.1s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   4.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=100, n_jobs=-1,
random_state=42; total time=   4.2s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  17.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  17.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  17.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   8.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  17.6s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=20, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  17.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   7.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   7.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
```

```
random_state=42; total time=   3.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   7.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=200, n_jobs=-1,
random_state=42; total time=   7.7s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.6s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=100, n_jobs=-1,
random_state=42; total time=   3.9s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   7.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  19.1s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   7.7s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  19.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  19.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  19.3s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=sqrt, n_estimators=500, n_jobs=-1,
random_state=42; total time=  19.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   7.6s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.8s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=200, n_jobs=-1,
random_state=42; total time=   6.8s
```

```
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  14.5s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  14.6s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  14.7s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  14.4s
[CV] END bootstrap=False, class_weight=balanced, criterion=entropy,
max_depth=50, max_features=log2, n_estimators=500, n_jobs=-1,
random_state=42; total time=  13.6s

/opt/homebrew/lib/python3.12/site-packages/numpy/ma/core.py:2881:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

GridSearchCV(cv=5, estimator=RandomForestClassifier(random_state=42),
n_jobs=-1,
             param_grid={'bootstrap': [True, False],
                         'class_weight': ['balanced'],
                         'criterion': ['gini', 'entropy'],
                         'max_depth': [10, 20, 50],
                         'max_features': ['sqrt', 'log2'],
                         'n_estimators': [100, 200, 500], 'n_jobs': [-
1],
                         'random_state': [42]},
             return_train_score=True, scoring='accuracy', verbose=2)

# Best Random Forest model and its parameters
best_rf_model = grid_rf.best_estimator_
best_rf_model.fit(X_train_scaled, y_train)

# Best Params and Model Accuracy
print(f"Best Random Forest parameters: {grid_rf.best_params_}")
print(f"Best Random Forest Cross-Validation Accuracy:
{grid_rf.best_score_:.4f}")

Best Random Forest parameters: {'bootstrap': True, 'class_weight':
'balanced', 'criterion': 'gini', 'max_depth': 10, 'max_features':
'sqrt', 'n_estimators': 100, 'n_jobs': -1, 'random_state': 42}
Best Random Forest Cross-Validation Accuracy: 0.7537

# Cross-validation mean and standard deviation for Random Forest
rf_cv_results = cross_val_score(best_rf_model, X_train_scaled,
y_train, cv=5, scoring='accuracy')

# Mean and Standard deviation
```

```
print(f"Random Forest Mean Accuracy: {rf_cv_results.mean():.4f}")
print(f"Random Forest Accuracy Standard Deviation:
{rf_cv_results.std():.4f}")
```

```
Random Forest Mean Accuracy: 0.7537
Random Forest Accuracy Standard Deviation: 0.0075
```

```
# Evaluate the best RandomForest model on the test set
y_pred_rf = best_rf_model.predict(X_test_scaled)
accuracy_rf = accuracy_score(y_test, y_pred_rf)

print(f"Random Forest Test Accuracy: {accuracy_rf:.4f}")
```

```
Random Forest Test Accuracy: 0.7524
```

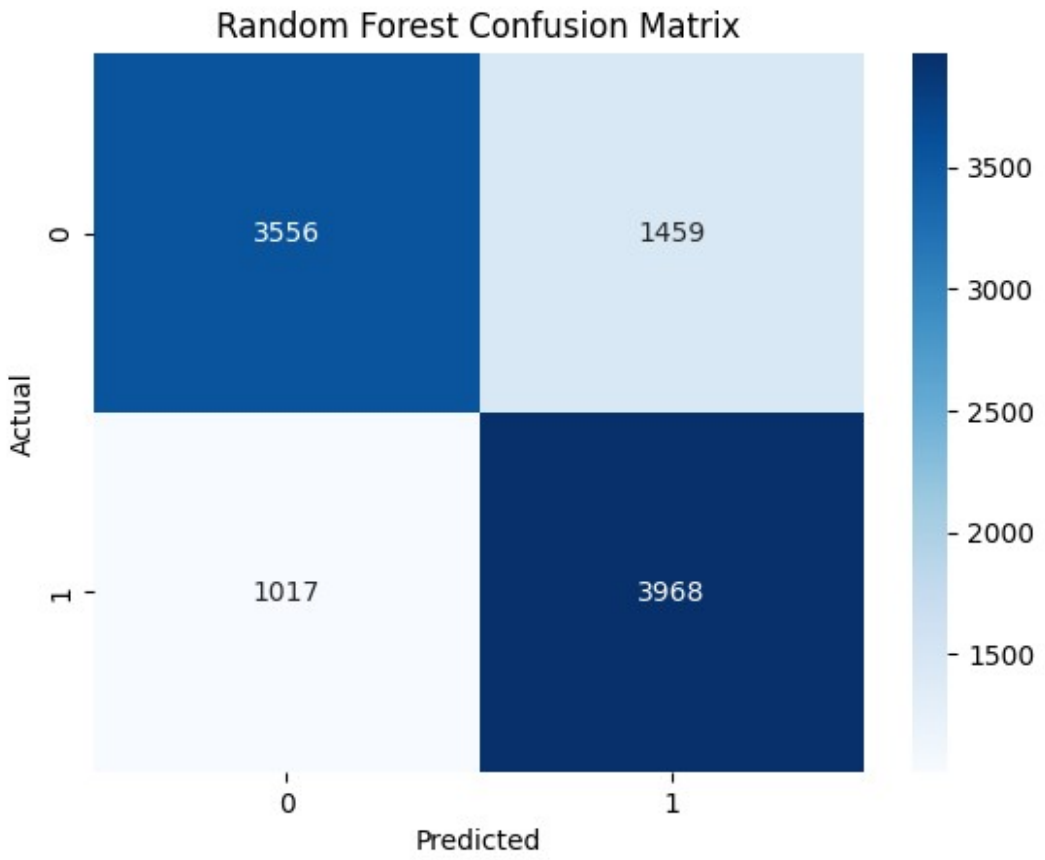
```
# Save the best RandomForest model to a pickle file
with open('best_rf_model.pkl', 'wb') as file:
    pickle.dump(best_rf_model, file)
```

```
# Evaluating rf the model using scaled test data
print("Evaluating model on test data...")
model_evaluate(best_rf_model,X_test_scaled,y_test)
```

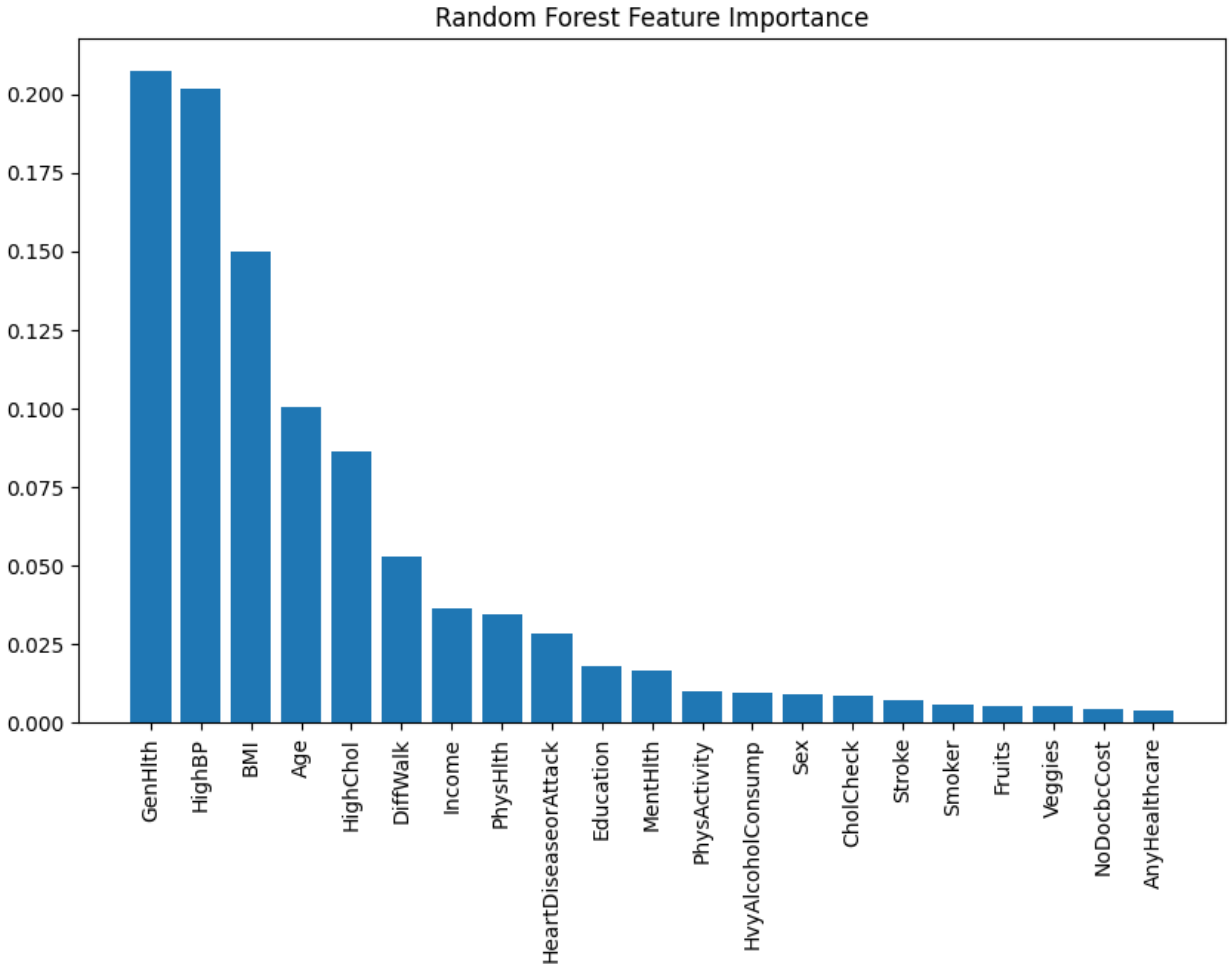
```
Evaluating model on test data...
Evaluating model performance on test data
Accuracy: 0.7524

Classification Report:
              precision    recall  f1-score   support

         0.0       0.78      0.71      0.74      5015
         1.0       0.73      0.80      0.76      4985

    accuracy                           0.75     10000
   macro avg       0.75      0.75      0.75     10000
weighted avg       0.75      0.75      0.75     10000
```

```
# Confusion Martix for Random Forest
print("Random Forest Confusion Matrix:")

conf_matrix_rf = confusion_matrix(y_test, y_pred_rf)
sns.heatmap(conf_matrix_rf, annot=True, fmt='d', cmap='Blues')
plt.title('Random Forest Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

```
Random Forest Confusion Matrix:
```




```
# Feature Importance for Random Forest
feature_importances_rf = best_rf_model.feature_importances_
sorted_idx_rf = np.argsort(feature_importances_rf)[::-1]
plt.figure(figsize=(10, 6))
plt.bar(range(X_train.shape[1]),
feature_importances_rf[sorted_idx_rf], align='center')
plt.xticks(range(X_train.shape[1]), X.columns[sorted_idx_rf],
rotation=90)
plt.title('Random Forest Feature Importance')
plt.show()
```


Random Forest Feature Importance
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
GenHlth
HighBP
BMI
Age
HighChol
DiffWalk
Income
PhysHlth
HeartDiseaseorAttack
Education
MentHlth
PhysActivity
HvyAlcoholConsump
Sex
CholCheck
Stroke
Smoker
Fruits
Veggies
NoDocbcCost
AnyHealthcare