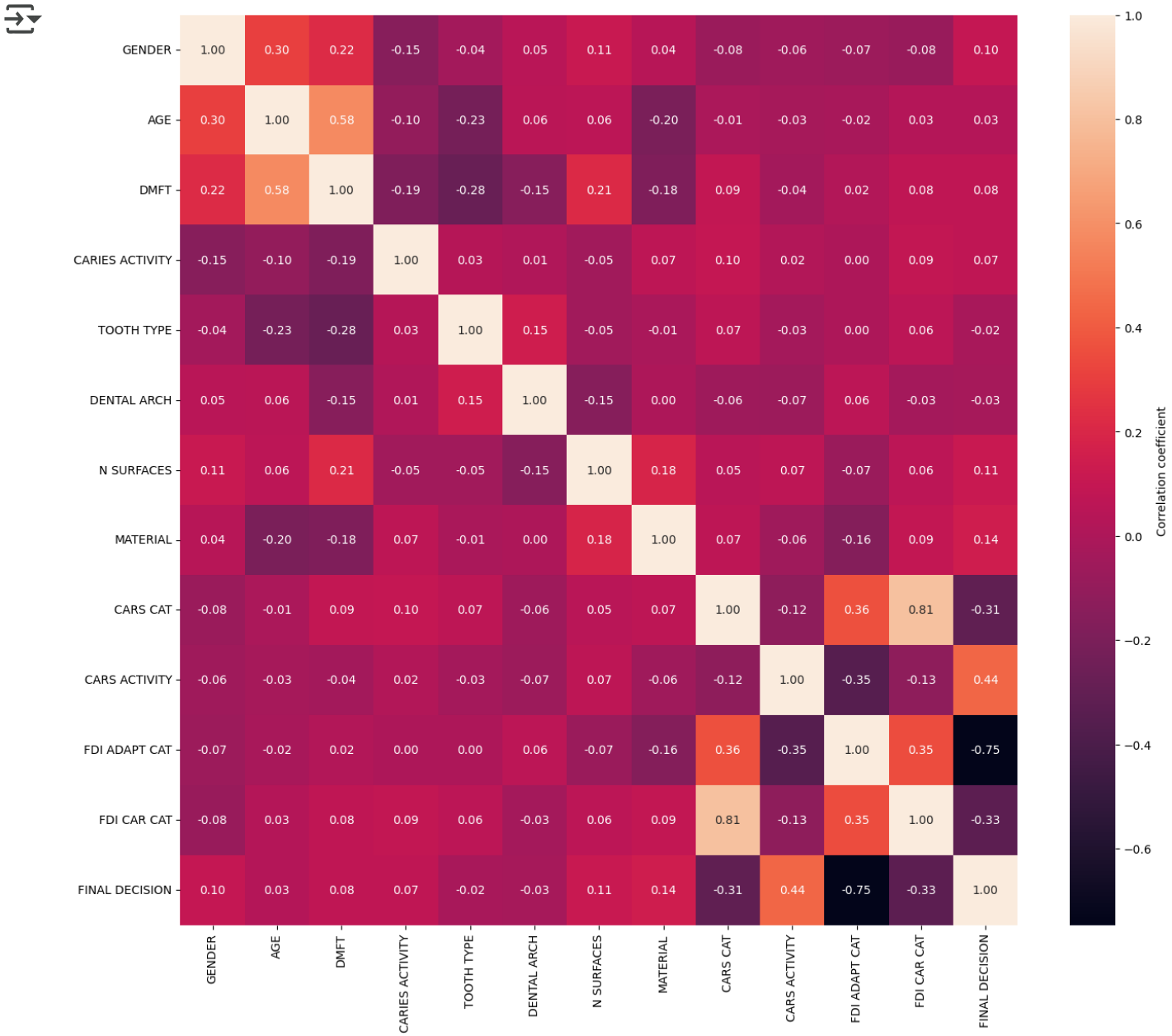
```
!pip install \
    scikit-learn==1.2.2 \
    numpy==1.25.2 \
    pandas==2.0.3 \
    scipy==1.11.2 \
    joblib==1.2.0 \
    threadpoolctl==3.1.0 \
    cython==0.29.36 \
    imbalanced-learn==0.12.0
```

```
Requirement already satisfied: scikit-learn==1.2.2 in /usr/local/lib/python
Requirement already satisfied: numpy==1.25.2 in /usr/local/lib/python3.11/d
Requirement already satisfied: pandas==2.0.3 in /usr/local/lib/python3.11/d
Requirement already satisfied: scipy==1.11.2 in /usr/local/lib/python3.11/d
Requirement already satisfied: joblib==1.2.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: threadpoolctl==3.1.0 in /usr/local/lib/pytho
Requirement already satisfied: cython==0.29.36 in /usr/local/lib/python3.11
Requirement already satisfied: imbalanced-learn==0.12.0 in /usr/local/lib/p
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pyt
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/di
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.11/
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-p
```

```
pip freeze > new_env_requirements.txt
```

```
# Importing necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer

# Load the data from an Excel file
data = pd.read_excel('AllFinal_CaCIA_Prediction_ML.xlsx')

# Split the dataset into training and testing sets based on a unique identifier
# This ensures that data related to the same 'N PART' is not split across both
unique_n_part = data['N PART'].unique()
train_n_part, test_n_part = train_test_split(unique_n_part, test_size=0.3, rand

# Filter the original dataset to create training data that includes only the 'N
train_data = data[data['N PART'].isin(train_n_part)]
# Similarly, filter the original dataset to create testing data that includes d
test_data = data[data['N PART'].isin(test_n_part)]

# Separate features and target variable for training set
# 'drop' removes specified columns from the dataset, in this case removing targ
X_train = train_data.drop(['ANY FAILURE', 'N TEETH', 'N PART'], axis=1)
y_train = train_data['ANY FAILURE'] # Isolate the target variable for the train

# Separate features and target variable for testing set following the same proc
X_test = test_data.drop(['ANY FAILURE', 'N TEETH', 'N PART'], axis=1)
y_test = test_data['ANY FAILURE'] # Isolate the target variable for the test se

# Impute missing values in 'DMFT' using median
imputer = SimpleImputer(strategy='median')
X_train['DMFT'] = imputer.fit_transform(X_train[['DMFT']])
X_test['DMFT'] = imputer.transform(X_test[['DMFT']])


import seaborn as sns
import matplotlib.pyplot as plt

# Calculate the correlation matrix of the training data.
# The correlation matrix quantifies the linear relationships between the variab
corr_matrix = X_train.corr()

# Initialize a matplotlib figure with a specified size (width=16 inches, height
# This size is chosen to make the heatmap large enough to be easily readable.
plt.figure(figsize=(16, 14))

# Draw the heatmap using seaborn to visualize the correlation matrix.
# `annot=True` displays the correlation coefficients in the heatmap cells.
# `annot_kws={"size": 10}` sets the font size of the annotations to 10 for bett
# `fmt=".2f"` formats the annotation text to show only two decimal places.
```

```
# `cbar_kws={'label': 'Correlation coefficient'}` adds a label to the color bar
sns.heatmap(corr_matrix, annot=True, annot_kws={"size": 10}, fmt=".2f", cbar_kw

# Display the plot on the screen. This command is necessary to show the figure
plt.show()
```



```
import pandas as pd

# Define the lists for each variable type
numeric_vars = ['AGE', 'DMFT']
categorical_vars = ['GENDER', 'CARIES ACTIVITY', 'TOOTH TYPE', 'DENTAL ARCH', '
                    'CARS CAT', 'FDI ADAPT CAT', 'FDI CAR CAT', 'ANY FAILURE',

def descriptive_statistics(X_train, y_train, X_test, y_test):
    # Merge features and target variable for descriptive statistics on the trai
    train_data_resampled = pd.concat([X_train, y_train], axis=1)

    # Merge features and target variable for descriptive statistics on the test
    test_data = pd.concat([X_test, y_test], axis=1)

    print("Descriptive Statistics for Numeric Variables:")
    print("\nTraining Set:")
    print(train_data_resampled[numeric_vars].describe())
    print("\nTest Set:")
    print(test_data[numeric_vars].describe())

    stats = {}
    for var in categorical_vars:
        stats[var] = {
            "Training Set": {
                "Count": train_data[var].value_counts().to_dict(),
                "Percentage": (train_data[var].value_counts(normalize=True) * 1
            },
            "Test Set": {
                "Count": test_data[var].value_counts().to_dict(),
                "Percentage": (test_data[var].value_counts(normalize=True) * 10
            }
        }

    # Print Categorical Statistics
    for var, data in stats.items():
        print(f"\n{var} Statistics:")
        for dataset, values in data.items():
            print(f"\n{dataset}:")
            for metric, metric_values in values.items():
```

```
            print(f"{metric}: {metric_values}")

# Call the function to display descriptive statistics for the train and test se
descriptive_statistics(X_train, y_train, X_test, y_test)
```

```
CARS CAT Statistics:

Training Set:
Count: {0: 219, 1: 95, 2: 39}
Percentage: {0: 62.03966005665722, 1: 26.912181303116146, 2: 11.04815864022

Test Set:
Count: {0: 101, 1: 33, 2: 14}
Percentage: {0: 68.24324324324324, 1: 22.2972972972973, 2: 9.45945945945946

FDI ADAPT CAT Statistics:

Training Set:
Count: {2: 259, 1: 78, 3: 16}
Percentage: {2: 73.37110481586402, 1: 22.096317280453256, 3: 4.532577903682

Test Set:
Count: {2: 106, 1: 35, 3: 7}
Percentage: {2: 71.62162162162163, 1: 23.64864864864865, 3: 4.7297297297297

FDI CAR CAT Statistics:

Training Set:
Count: {1: 213, 2: 115, 3: 25}
Percentage: {1: 60.3399433427762, 2: 32.577903682719544, 3: 7.0821529745042

Test Set:
Count: {1: 98, 2: 46, 3: 4}
Percentage: {1: 66.21621621621621, 2: 31.08108108108108, 3: 2.7027027027027

ANY FAILURE Statistics:

Training Set:
Count: {0: 307, 1: 46}
Percentage: {0: 86.96883852691218, 1: 13.031161473087819}

Test Set:
Count: {0: 132, 1: 16}
Percentage: {0: 89.1891891891892, 1: 10.81081081081081}

FINAL DECISION Statistics:

Training Set:
Count: {0: 289, 1: 36, 2: 28}
Percentage: {0: 81.86968838526913, 1: 10.198300283286118, 2: 7.932011331444

Test Set:
Count: {0: 118, 1: 18, 2: 12}
```

```
    Percentage: {0: 79.72972972972973, 1: 12.162162162162163, 2: 8.108108108108

    N SURFACES Statistics:

    Training Set:
    Count: {1: 181, 2: 113, 3: 39, 4: 16, 5: 4}
    Percentage: {1: 51.27478753541076, 2: 32.01133144475921, 3: 11.048158640226

    Test Set:
    Count: {1: 95, 2: 37, 3: 9, 4: 4, 5: 3}
    Percentage: {1: 64.1891891891892, 2: 25.0, 3: 6.081081081081082, 4: 2.70270
```

```
from sklearn.preprocessing import StandardScaler, OrdinalEncoder
from imblearn.over_sampling import SMOTE

# Initialize OrdinalEncoder
ordinal_encoder = OrdinalEncoder()

# Apply Ordinal Encoding to 'FINAL DECISION', 'CARS CAT', and 'N SURFACES CAT'
X_train[['N SURFACES']] = ordinal_encoder.fit_transform(X_train[['N SURFACES']]

# Apply the same ordinal encoding to the test data
X_test[['N SURFACES']] = ordinal_encoder.transform(X_test[['N SURFACES']])

# Convert specified categorical variables in the training data to 'category' dt
X_train['FINAL DECISION'] = X_train['FINAL DECISION'].astype('category')
X_train['CARS CAT'] = X_train['CARS CAT'].astype('category')
X_train['FDI ADAPT CAT'] = X_train['FDI ADAPT CAT'].astype('category')
X_train['FDI CAR CAT'] = X_train['FDI CAR CAT'].astype('category')
# Apply one-hot encoding to the specified categorical columns in the training d
# 'prefix' argument specifies the prefix to add to the columns resulting from t
one_hot_train = pd.get_dummies(X_train[['FINAL DECISION', 'CARS CAT', 'FDI ADAP
                               prefix=['FINALDECISION', 'CARSCAT','FDI ADAPT CA
# Concatenate the original training data (minus the now-encoded variables) with
X_train = pd.concat([X_train.drop(['FINAL DECISION', 'CARS CAT', 'FDI ADAPT CAT

# Initialize new one-hot encoded columns in the test data with zeros to match t
for col in one_hot_train.columns:
    X_test[col] = 0

# Convert specified categorical variables in the test data to 'category' dtype
X_test['FINAL DECISION'] = X_test['FINAL DECISION'].astype('category')
X_test['CARS CAT'] = X_test['CARS CAT'].astype('category')
X_test['FDI ADAPT CAT'] = X_test['FDI ADAPT CAT'].astype('category')
X_test['FDI CAR CAT'] = X_test['FDI CAR CAT'].astype('category')
one_hot_test = pd.get_dummies(X_test[['FINAL DECISION', 'CARS CAT', 'FDI ADAPT
                              prefix=['FINALDECISION', 'CARSCAT', 'FDI ADAPT CA
# Update the test data with the new one-hot encoded columns.
X_test.update(one_hot_test)
```

```
# Check for any columns that are present in the training data but missing in th
# which might happen if the test data lacks certain categories.
missing_cols = set(X_train.columns) - set(X_test.columns)
for c in missing_cols:
    X_test[c] = 0 # Add these missing columns to the test data, initializing wi

# Ensure the column order in the test data matches that of the training data fo
X_test = X_test[X_train.columns]

# ————————————————————————————————————————————————————————————————————————————
# ADD THIS STEP HERE
bool_cols_train = X_train.select_dtypes(include=['bool']).columns
X_train[bool_cols_train] = X_train[bool_cols_train].astype(int)

bool_cols_test = X_test.select_dtypes(include=['bool']).columns
X_test[bool_cols_test] = X_test[bool_cols_test].astype(int)
# ————————————————————————————————————————————————————————————————————————————

# Define a dictionary to rename the one-hot encoded columns for clarity, making
column_renaming = {
    'FDI CAR CAT_1': 'FDI No Caries',
    'FDI CAR CAT_2': 'FDI Initial Caries',
    'FDI CAR CAT_3': 'FDI Advanced Caries',
    'FDI ADAPT CAT_1': 'FDI No Adaptation',
    'FDI ADAPT CAT_2': 'FDI Initial Adaptation',
    'FDI ADAPT CAT_3': 'FDI Advanced Adaptation',
    'CARSCAT_0': 'CARS No Caries',
    'CARSCAT_1': 'CARS Initial Caries',
    'CARSCAT_2': 'CARS Advanced Caries',
    'FINALDECISION_0': 'No Initial Intervention',
    'FINALDECISION_1': 'Repaired',
    'FINALDECISION_2': 'Replaced'
}

# Rename the columns in both the training and test datasets according to the de
X_train.rename(columns=column_renaming, inplace=True)
X_test.rename(columns=column_renaming, inplace=True)

# Scale the numerical features in both training and test datasets to have mean
# This is crucial for models that are sensitive to the scale of input features.
scaler = StandardScaler()
X_train.loc[:, ['AGE', 'DMFT']] = scaler.fit_transform(X_train[['AGE', 'DMFT']]
X_test.loc[:, ['AGE', 'DMFT']] = scaler.transform(X_test[['AGE', 'DMFT']])

# Define which columns are categorical
categorical_features = list(range(len(X_train.columns)))
for col in ['AGE', 'DMFT']:  # Assuming these are your only continuous features
```

```
    categorical_features.remove(X_train.columns.get_loc(col))

# Use SMOTE to balance the train set
smote = SMOTE(sampling_strategy='minority', random_state=42, k_neighbors=5)  #
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)  #

# Adjust 'N SURFACES' back to original range (1 to 5) by adding 1
X_train_resampled['N SURFACES'] = X_train_resampled['N SURFACES'] + 1
X_test['N SURFACES'] = X_test['N SURFACES'] + 1
```

```
<ipython-input-6-6c6af16f0a10>:75: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs
  X_test.rename(columns=column_renaming, inplace=True)
```

```
import pandas as pd

# Define the lists for each variable type
numeric_vars = ['AGE', 'DMFT']
original_categorical_vars = ['GENDER', 'CARIES ACTIVITY', 'TOOTH TYPE', 'DENTAL
                             'ANY FAILURE']
one_hot_encoded_vars = ['N SURFACES', 'FDI No Caries', 'FDI Initial Caries', 'F
                        'FDI No Adaptation','FDI Initial Adaptation', 'FDI Adva
                        'CARS No Caries', 'CARS Initial Caries', 'CARS Advanced
                        'No Initial Intervention', 'Repaired', 'Replaced']

def descriptive_statistics(X_train_resampled, y_train_resampled, X_test, y_test
    # Merge features and target variable for descriptive statistics on the trai
    train_data_resampled = pd.concat([X_train_resampled, y_train_resampled], ax

    # Merge features and target variable for descriptive statistics on the test
    test_data = pd.concat([X_test, y_test], axis=1)

    print("Descriptive Statistics for Numeric Variables:")
    print("\nResampled Training Set:")
    print(train_data_resampled[numeric_vars].describe())
    print("\nTest Set:")
    print(test_data[numeric_vars].describe())

    stats = {}
    for var in original_categorical_vars:
        stats[var] = {
            "Resampled Training Set": {
                "Count": train_data_resampled[var].value_counts().to_dict(),
                "Percentage": (train_data_resampled[var].value_counts(normalize
            },
            "Test Set": {
```

```
                "Count": test_data[var].value_counts().to_dict(),
                "Percentage": (test_data[var].value_counts(normalize=True) * 10
            }
        }

    # Handle one-hot encoded variables
    for var in one_hot_encoded_vars:
        encoded_columns = [col for col in train_data_resampled if col.startswit
        for col in encoded_columns:
            stats[col] = {
                "Resampled Training Set": {
                    "Count": train_data_resampled[col].value_counts().to_dict()
                    "Percentage": (train_data_resampled[col].value_counts(norma
                },
                "Test Set": {
                    "Count": test_data[col].value_counts().to_dict(),
                    "Percentage": (test_data[col].value_counts(normalize=True)
                }
            }

    # Print Categorical Statistics
    for var, data in stats.items():
        print(f"\n{var} Statistics:")
        for dataset, values in data.items():
            print(f"\n{dataset}:")
            for metric, metric_values in values.items():
                print(f"{metric}: {metric_values}")

# Call the function to display descriptive statistics for the resampled train a
descriptive_statistics(X_train_resampled, y_train_resampled, X_test, y_test)
```

```
CARS No Caries Statistics:

Resampled Training Set:
Count: {1: 343, 0: 271}
Percentage: {1: 55.86319218241043, 0: 44.13680781758957}

Test Set:
Count: {1: 101, 0: 47}
Percentage: {1: 68.24324324324324, 0: 31.756756756756754}

CARS Initial Caries Statistics:

Resampled Training Set:
Count: {0: 454, 1: 160}
Percentage: {0: 73.9413680781759, 1: 26.058631921824105}

Test Set:
Count: {0: 115, 1: 33}
```

```
Percentage: {0: 77.7027027027027, 1: 22.2972972972973}

CARS Advanced Caries Statistics:

Resampled Training Set:
Count: {0: 559, 1: 55}
Percentage: {0: 91.04234527687296, 1: 8.957654723127035}

Test Set:
Count: {0: 134, 1: 14}
Percentage: {0: 90.54054054054053, 1: 9.45945945945946}

No Initial Intervention Statistics:

Resampled Training Set:
Count: {1: 501, 0: 113}
Percentage: {1: 81.59609120521174, 0: 18.403908794788272}

Test Set:
Count: {1: 118, 0: 30}
Percentage: {1: 79.72972972972973, 0: 20.27027027027027}

Repaired Statistics:

Resampled Training Set:
Count: {0: 566, 1: 48}
Percentage: {0: 92.18241042345277, 1: 7.81758957654723}

Test Set:
Count: {0: 130, 1: 18}
Percentage: {0: 87.83783783783784, 1: 12.162162162162163}

Replaced Statistics:

Resampled Training Set:
Count: {0: 582, 1: 32}
Percentage: {0: 94.78827361563518, 1: 5.211726384364821}

Test Set:
Count: {0: 136, 1: 12}
Percentage: {0: 91.8918918918919, 1: 8.108108108108109}
```

```
# Define custom metrics
def sensitivity(y_true, y_pred):
    tn, fp, fn, tp = confusion_matrix(y_true, y_pred).ravel()
    return tp / (tp + fn)

def specificity(y_true, y_pred):
    tn, fp, fn, tp = confusion_matrix(y_true, y_pred).ravel()
    return tn / (tn + fp)
```

```
!pip install xgboost shap

import pandas as pd
import numpy as np
import shap
import sys
import tensorflow as tf
import matplotlib.pyplot as plt
import random
import seaborn as sns
from imblearn.pipeline import Pipeline as IMBPipeline
from sklearn.model_selection import cross_val_score
from sklearn.calibration import CalibratedClassifierCV
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.model_selection import cross_validate, StratifiedKFold, GridSearch
from sklearn.metrics import make_scorer, accuracy_score, roc_auc_score, f1_scor
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, BatchNormalization
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau, Learni
from tensorflow.keras.regularizers import l2
from scipy import stats
```

```
Requirement already satisfied: xgboost in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: shap in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: nvidia-nccl-cu12 in /usr/local/lib/python3.1
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/di
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: tqdm>=4.27.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: packaging>20.9 in /usr/local/lib/python3.11/
Requirement already satisfied: slicer==0.0.8 in /usr/local/lib/python3.11/d
Requirement already satisfied: numba in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.11/dis
Requirement already satisfied: llvmlite<0.45,>=0.44.0dev0 in /usr/local/lib
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pyt
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/di
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.11/
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.11/d
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/pytho
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-p
```

```
def evaluate_model(model, name, grid, X_train, y_train, X_test, y_test, cv, scor
    print(f"\nEvaluating {name} with seed {seed}...")

    # Define inner and outer CV splits using the provided seed
```

```
    inner_cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=seed)
    outer_cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=seed)

    # Grid search using inner CV
    clf = GridSearchCV(model, grid, cv=inner_cv, scoring='roc_auc')
    nested_scores = cross_validate(clf, X=X_train, y=y_train, cv=outer_cv, scori

    # Fit grid search on full training set and extract the best estimator
    clf.fit(X_train, y_train)
    best_model = clf.best_estimator_
    best_params = clf.best_params_

    print(f"Best parameters for {name}: {best_params}")

    # Calibrate the best model
    calibrated_clf = CalibratedClassifierCV(estimator=best_model, method='sigmoi
    calibrated_clf.fit(X_train, y_train)

    # Get predicted probabilities on the test set from the calibrated classifier
    y_probs = calibrated_clf.predict_proba(X_test)[:, 1]

    # --- Calculate Training Metrics ---
    y_train_pred = best_model.predict(X_train)
    y_train_probs = best_model.predict_proba(X_train)[:, 1]
    train_acc   = accuracy_score(y_train, y_train_pred)
    train_sens  = sensitivity(y_train, y_train_pred)
    train_spec  = specificity(y_train, y_train_pred)
    train_f1    = f1_score(y_train, y_train_pred)
    train_roc_auc = roc_auc_score(y_train, y_train_probs)

    print(f"Training Metrics - Accuracy: {train_acc:.3f}, Sensitivity: {train_se

    # --- Calculate Test Metrics for the manually set threshold ---
    y_pred_manual = (y_probs >= manual_threshold).astype(int)
    manual_acc   = accuracy_score(y_test, y_pred_manual)
    manual_sens  = sensitivity(y_test, y_pred_manual)
    manual_spec  = specificity(y_test, y_pred_manual)
    manual_f1    = f1_score(y_test, y_pred_manual)
    manual_roc_auc = roc_auc_score(y_test, y_probs)

    print(f"\nTest Metrics for manual threshold {manual_threshold}:")
    print(f"Accuracy: {manual_acc:.3f}, Sensitivity: {manual_sens:.3f}, Specific

    # --- Evaluate metrics across a range of thresholds ---
    threshold_metrics = {}
    for threshold in threshold_list:
        y_pred_threshold = (y_probs >= threshold).astype(int)
        threshold_acc   = accuracy_score(y_test, y_pred_threshold)
        threshold_sens  = sensitivity(y_test, y_pred_threshold)
```

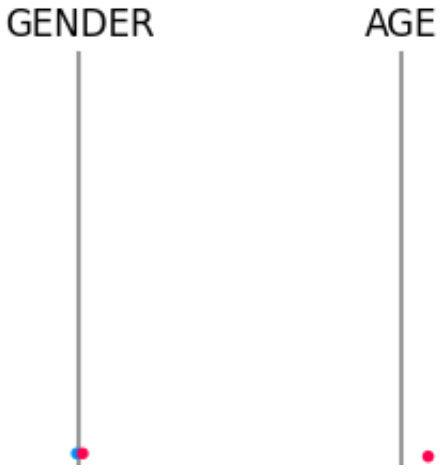
```
        threshold_sens  = sensitivity(y_test, y_pred_threshold)
        threshold_spec  = specificity(y_test, y_pred_threshold)
        threshold_f1    = f1_score(y_test, y_pred_threshold)
        threshold_metrics[threshold] = {
            'Accuracy': threshold_acc,
            'Sensitivity': threshold_sens,
            'Specificity': threshold_spec,
            'F1': threshold_f1,
            'ROC AUC': manual_roc_auc  # same ROC AUC regardless of threshold
        }
    for threshold, metrics in threshold_metrics.items():
        print(f"Threshold: {threshold:.2f}, Metrics: {metrics}")

    # Plot SHAP summary
    calculate_and_plot_shap(best_model, X_train, X_test, name)

    # Prepare dictionary of test metrics at the manual threshold for aggregation
    test_metrics = {
        "accuracy": manual_acc,
        "sensitivity": manual_sens,
        "specificity": manual_spec,
        "f1": manual_f1,
        "roc_auc": manual_roc_auc
    }

    return best_model, manual_threshold, best_params, nested_scores, calibrated_

def calculate_and_plot_shap(model, X_train, X_test, model_name):
    # Use TreeExplainer if model is a decision tree; otherwise use KernelExplain
    if isinstance(model, DecisionTreeClassifier):
        explainer = shap.TreeExplainer(model)
    else:
        explainer = shap.KernelExplainer(model.predict_proba, X_train.sample(100
    shap_values = explainer.shap_values(X_test)
    print(f"SHAP Summary for {model_name}")
    shap.summary_plot(shap_values, X_test, max_display=10)

def plot_confusion_matrix(y_true, y_pred):
    matrix = confusion_matrix(y_true, y_pred)
    sns.heatmap(matrix, annot=True, fmt='d', cmap='Blues',
                xticklabels=['Predicted Success', 'Predicted Failure'],
                yticklabels=['Actual Success', 'Actual Failure'])
    plt.title('Confusion Matrix')
    plt.show()

def plot_roc_curve(y_true, y_probs):
    fpr, tpr, thresholds = roc_curve(y_true, y_probs)
    roc_auc = auc(fpr, tpr)
    plt.figure()
```

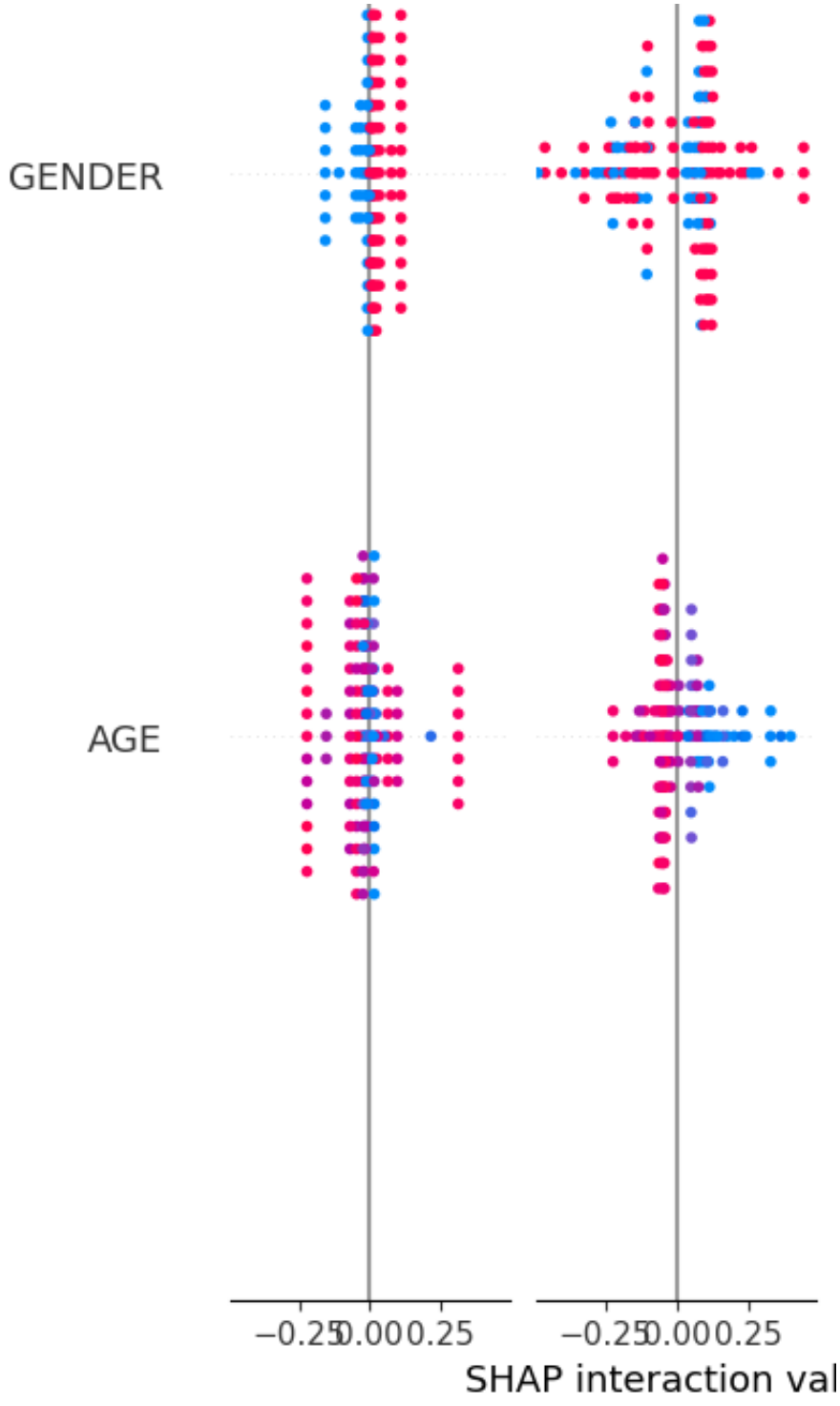
```
    plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area = {roc_
    plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver Operating Characteristic')
    plt.legend(loc="lower right")
    plt.show()

    # --- Added code to output ROC curve metrics ---
    print("ROC Curve Metrics:")
    print("FPR:", fpr)
    print("TPR:", tpr)
    print("ROC AUC: {:.3f}".format(roc_auc))

    return fpr, tpr, roc_auc

def evaluate_decision_tree(X_train_resampled, y_train_resampled, X_test, y_test,
    # Initialize the Decision Tree with the current seed and define grid for hyp
    model = DecisionTreeClassifier(random_state=seed)
    grid = {
        'max_depth': [10],
        'criterion': ['entropy'],
        'min_samples_split': [10],
        'min_samples_leaf': [10],
        'ccp_alpha': [0.002]
    }
    return evaluate_model(model, "Decision Tree", grid, X_train_resampled, y_tra

# ============================================================================
# MAIN FUNCTION: AGGREGATING METRICS ACROSS SEEDS
# ============================================================================
def main(X_train_resampled, y_train_resampled, X_test, y_test):
    # Define outer CV (used only for scoring here) and scoring metrics
    cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=10, random_state=42)
    scoring = {
        'accuracy': make_scorer(accuracy_score),
        'sensitivity': make_scorer(sensitivity),
        'specificity': make_scorer(specificity),
        'f1': make_scorer(f1_score),
        'roc_auc': make_scorer(roc_auc_score)
    }
    manual_threshold = 0.4
    threshold_list = np.arange(0.1, 1.05, 0.05)

    # List to collect test metrics from each seed iteration
    aggregated_metrics = []
```

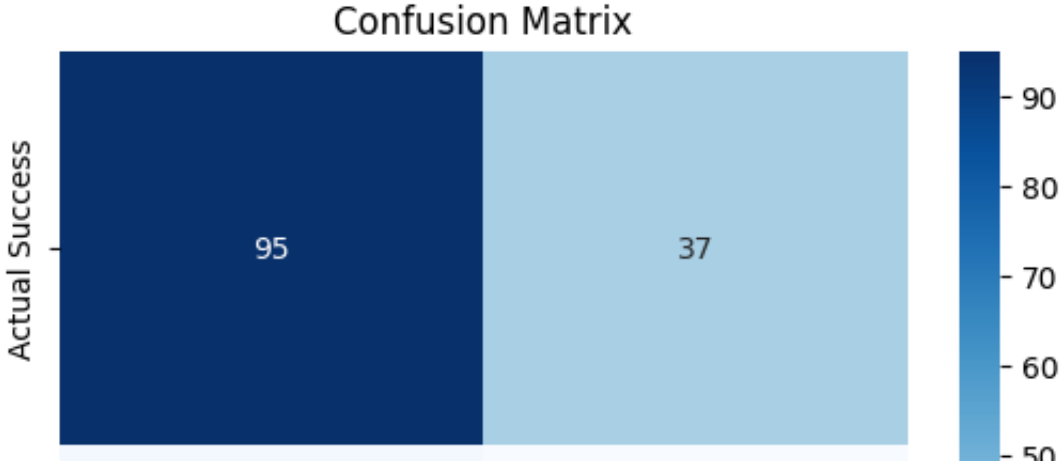
```
# Loop over different seeds for model training/evaluation
for seed in range(40, 50):
    print(f"\nRunning evaluation with seed {seed}")
    (best_model, manual_threshold, best_params, nested_scores,
     calibrated_clf, threshold_metrics, test_metrics) = evaluate_decision_tr
        X_train_resampled, y_train_resampled, X_test, y_test, cv, scoring, m
    )

    # Predict probabilities using the calibrated classifier for plotting pur
    y_probs = calibrated_clf.predict_proba(X_test)[:, 1]
    y_pred_manual = (y_probs >= manual_threshold).astype(int)

    # Plot confusion matrix and ROC curve for this seed
    plot_confusion_matrix(y_test, y_pred_manual)
    # The following call now prints FPR, TPR, and ROC AUC values.
    plot_roc_curve(y_test, y_probs)

    # Append the test set metrics from this seed for later aggregation
    aggregated_metrics.append(test_metrics)

# ============================================================================
# AGGREGATE RESULTS ACROSS SEEDS
# ============================================================================
# Convert list of dictionaries into a DataFrame for easier aggregation
results_df = pd.DataFrame(aggregated_metrics)
n = len(results_df)
print("\nAggregated Test Set Metrics Across Seeds:")
print(results_df)

# Function to compute mean, standard error, and 95% confidence interval usin
def summarize_metric(metric_values):
    mean_val = metric_values.mean()
    std_val = metric_values.std(ddof=1)
    se = std_val / np.sqrt(n)
    t_crit = stats.t.ppf(0.975, df=n-1)  # 95% confidence, two-tailed
    ci_lower = mean_val - t_crit * se
    ci_upper = mean_val + t_crit * se
    return mean_val, se, (ci_lower, ci_upper)

metrics_summary = {}
for metric in results_df.columns:
    mean_val, se, ci = summarize_metric(results_df[metric])
    metrics_summary[metric] = {
        "Mean": mean_val,
        "Standard Error": se,
        "95% CI": ci
    }
```

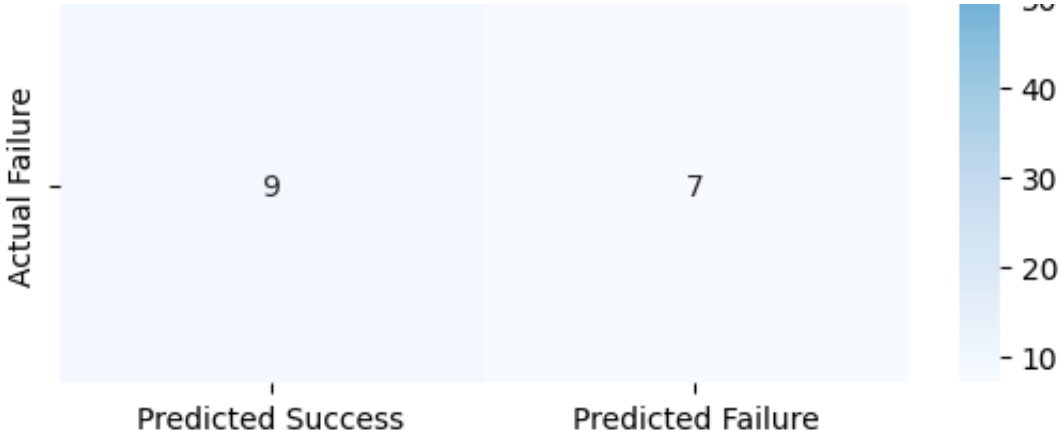
```
    print("\nSummary of Test Set Metrics (Mean, Standard Error, 95% Confidence I
    for metric, summary in metrics_summary.items():
        print(f"{metric.capitalize()}: Mean = {summary['Mean']:.3f}, SE = {summa
              f"95% CI = [{summary['95% CI'][0]:.3f}, {summary['95% CI'][1]:.3f}

# RUN THE MAIN FUNCTION (ASSUMING X_train_resampled, y_train_resampled, X_test,
if __name__ == '__main__':
    main(X_train_resampled, y_train_resampled, X_test, y_test)
```

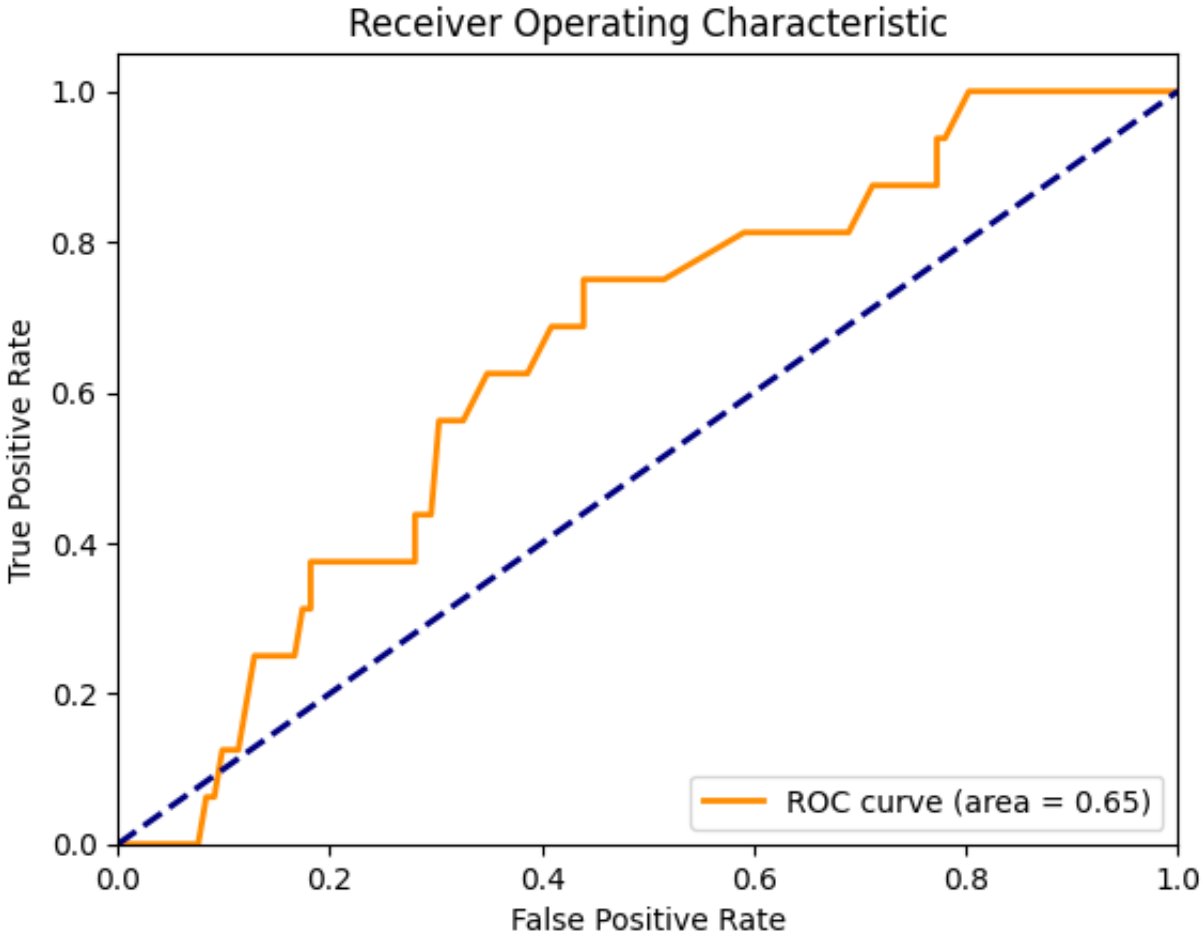
```
Running evaluation with seed 40

Evaluating Decision Tree with seed 40...
Best parameters for Decision Tree: {'ccp_alpha': 0.002, 'criterion': 'entro
Training Metrics - Accuracy: 0.862, Sensitivity: 0.860, Specificity: 0.863,

Test Metrics for manual threshold 0.4:
Accuracy: 0.689, Sensitivity: 0.438, Specificity: 0.720, F1: 0.233, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.10810810810810811, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.20270270270270271, 'Sensitivity':
Threshold: 0.20, Metrics: {'Accuracy': 0.4391891891891892, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.5337837837837838, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6418918918918919, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6891891891891891, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7094594594594594, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Decision Tree
```




GENDER
AGE
−0.25
0.00
0.25
−0.25
0.00
0.25
SHAP interaction val


Confusion Matrix
Actual Success
95
37
90
80
70
60
50

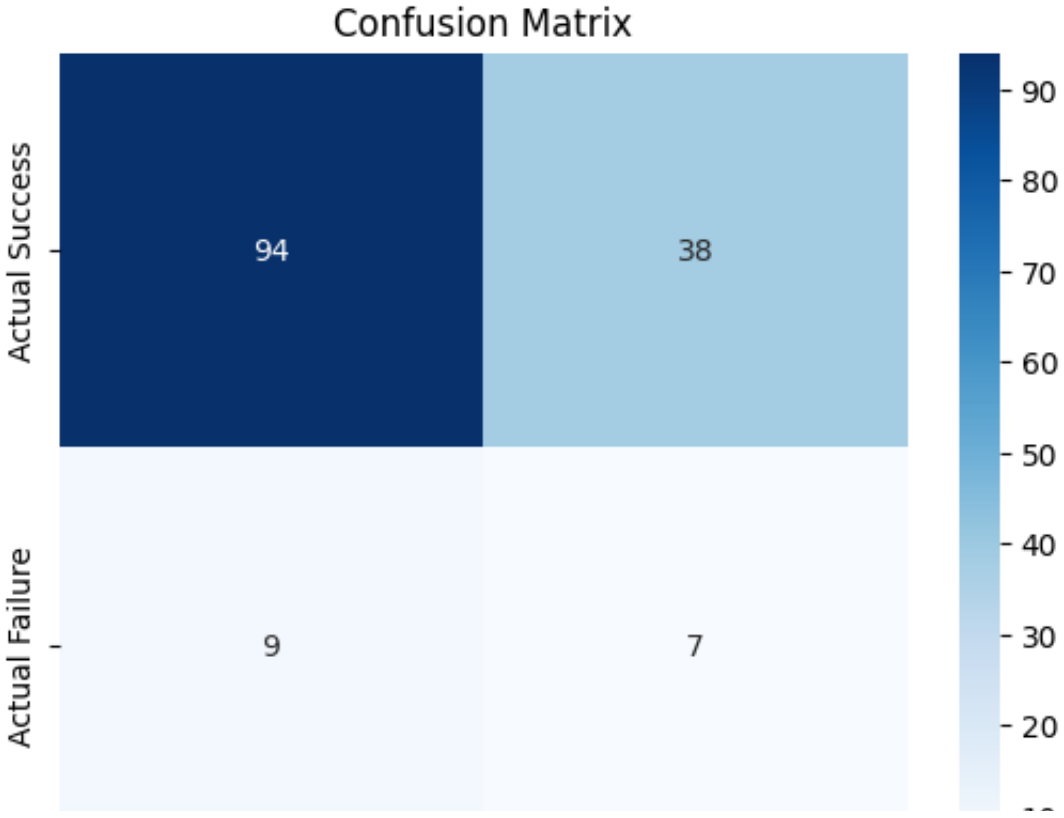


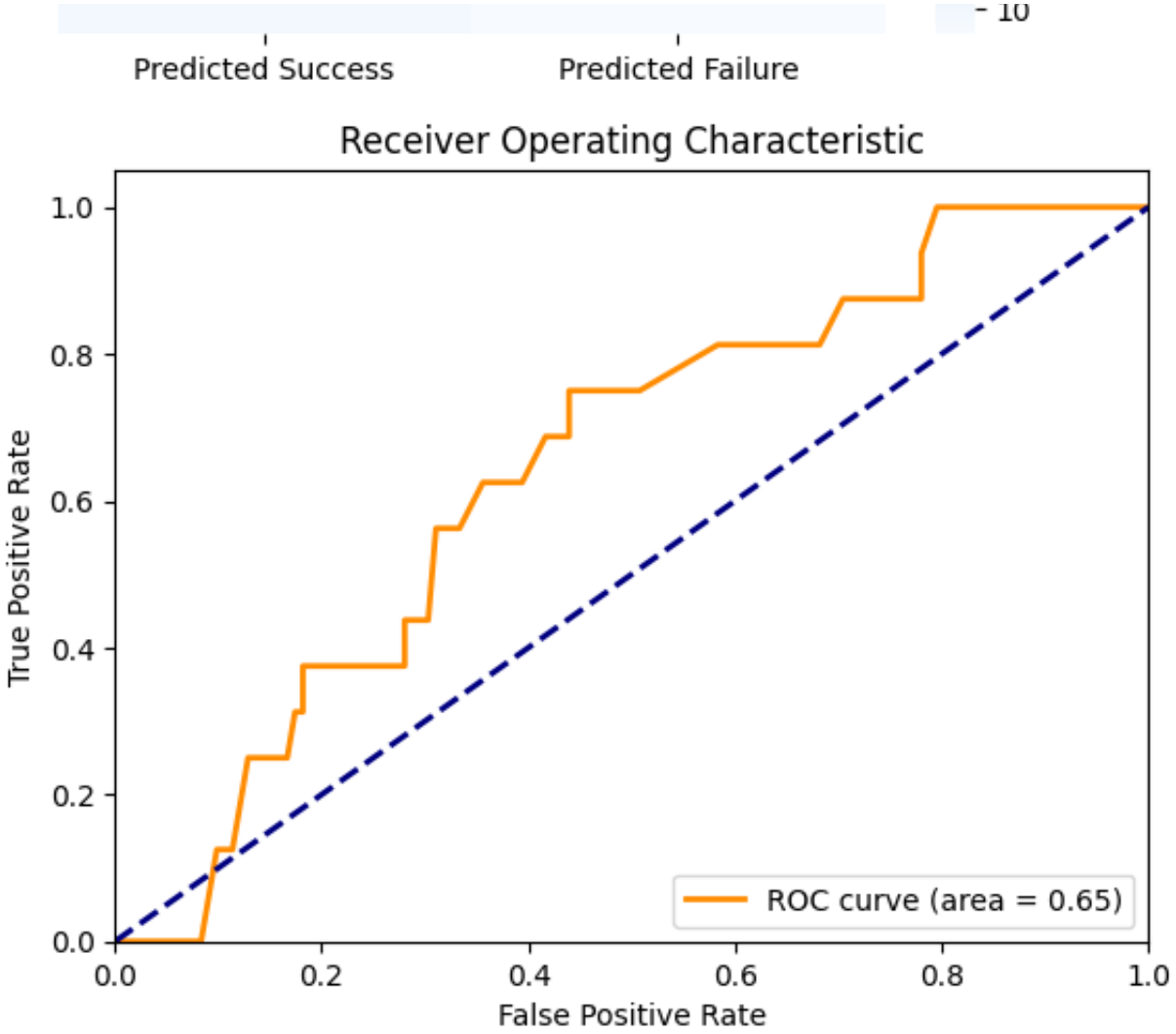



```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.03787879 0.06818182 0.07575758
 0.08333333 0.09090909 0.09848485 0.11363636 0.12878788 0.15151515
 0.16666667 0.17424242 0.18181818 0.18181818 0.1969697  0.21969697
 0.25       0.28030303 0.28030303 0.29545455 0.3030303  0.32575758
 0.34848485 0.35606061 0.38636364 0.40909091 0.42424242 0.43939394
 0.43939394 0.4469697  0.47727273 0.50757576 0.51515152 0.59090909
 0.61363636 0.62878788 0.65151515 0.68181818 0.68939394 0.71212121
 0.76515152 0.77272727 0.77272727 0.78030303 0.8030303  0.82575758
 0.85606061 0.86363636 0.89393939 1.        ]
TPR: [0.     0.     0.     0.     0.     0.     0.0625 0.0625 0.125  0.125
 0.25   0.25   0.25   0.3125 0.3125 0.375  0.375  0.375  0.375  0.375
 0.4375 0.4375 0.5625 0.5625 0.625  0.625  0.625  0.6875 0.6875 0.6875
 0.75   0.75   0.75   0.75   0.75   0.8125 0.8125 0.8125 0.8125 0.8125
 0.8125 0.875  0.875  0.875  0.9375 0.9375 1.     1.     1.     1.
```

```
 1.     1.    ]
ROC AUC: 0.647

Running evaluation with seed 41

Evaluating Decision Tree with seed 41...
Best parameters for Decision Tree: {'ccp_alpha': 0.002, 'criterion': 'entro
Training Metrics - Accuracy: 0.862, Sensitivity: 0.860, Specificity: 0.863,

Test Metrics for manual threshold 0.4:
Accuracy: 0.682, Sensitivity: 0.438, Specificity: 0.712, F1: 0.230, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.10810810810810811, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.22972972972972974, 'Sensitivity':
Threshold: 0.20, Metrics: {'Accuracy': 0.4527027027027027, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.5472972972972973, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6081081081081081, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6418918918918919, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6824324324324325, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7027027027027027, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.7905405405405406, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Decision Tree
```

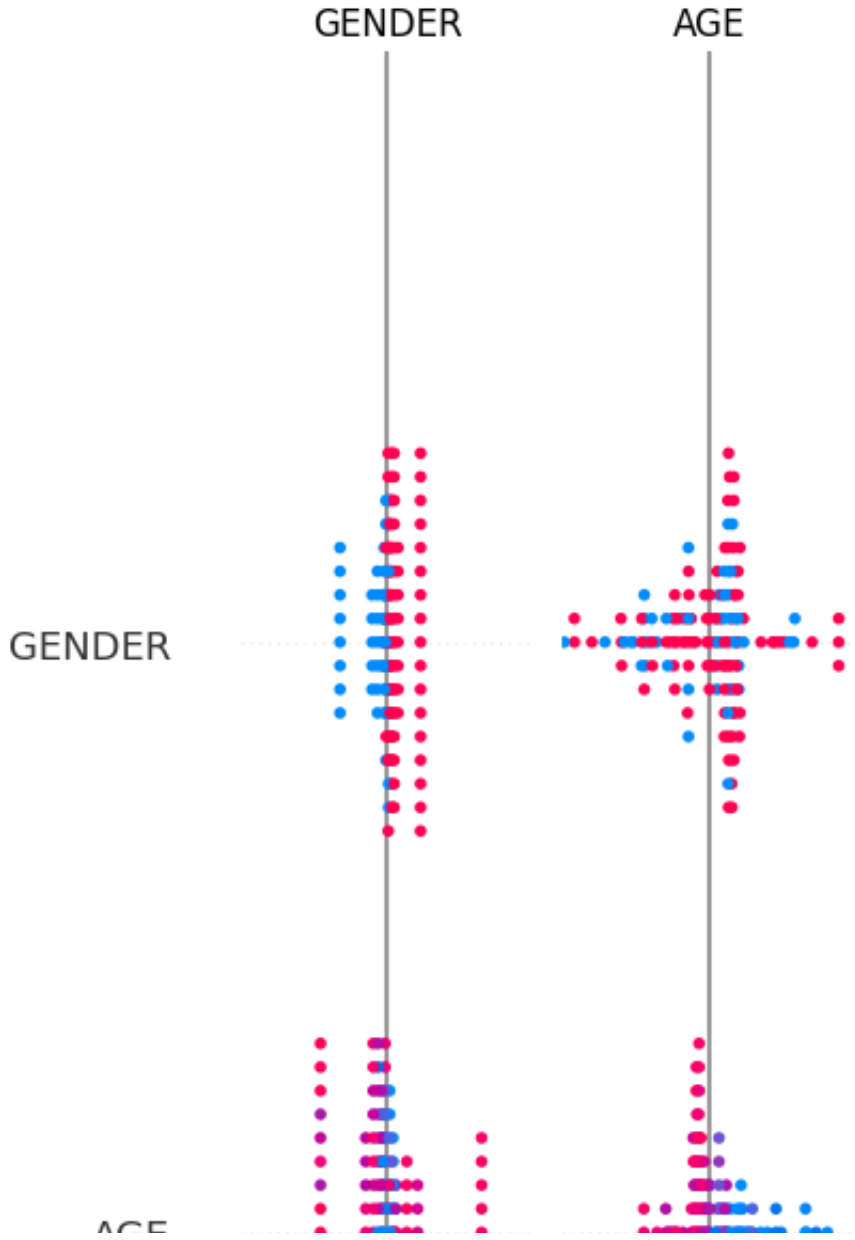

AGE
−0.250.000.25
−0.250.000.25
SHAP interaction val

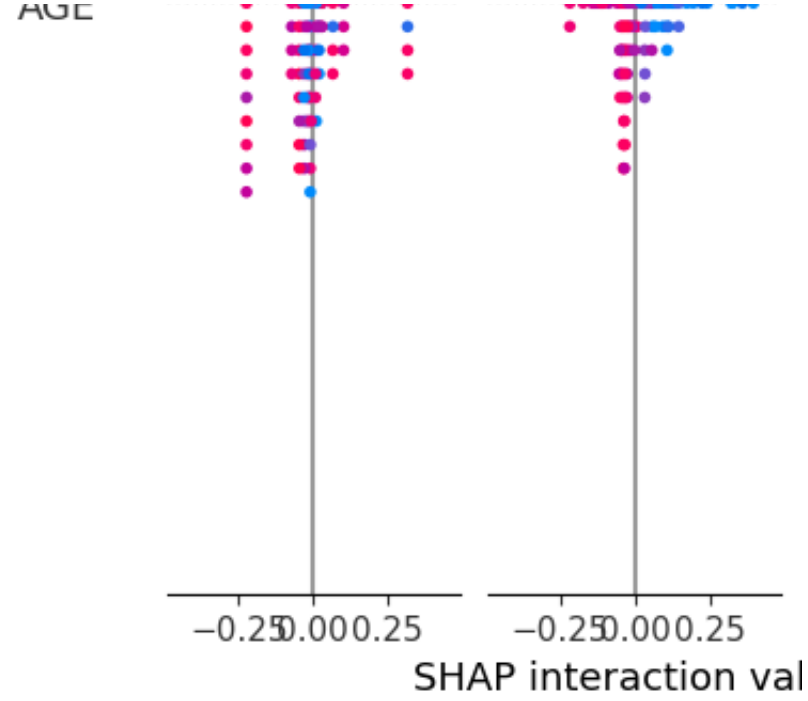

Confusion Matrix
Actual Success
Actual Failure
94
38
9
7
90
80
70
60
50
40
30
20

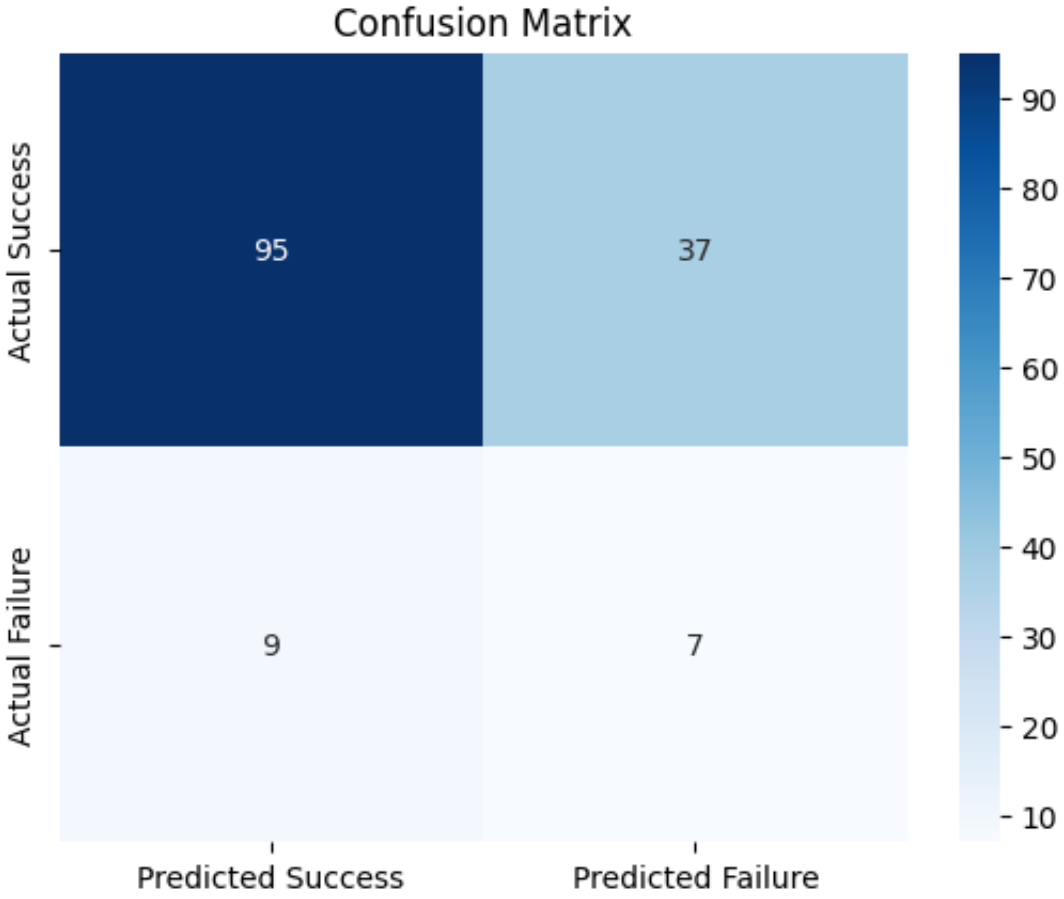


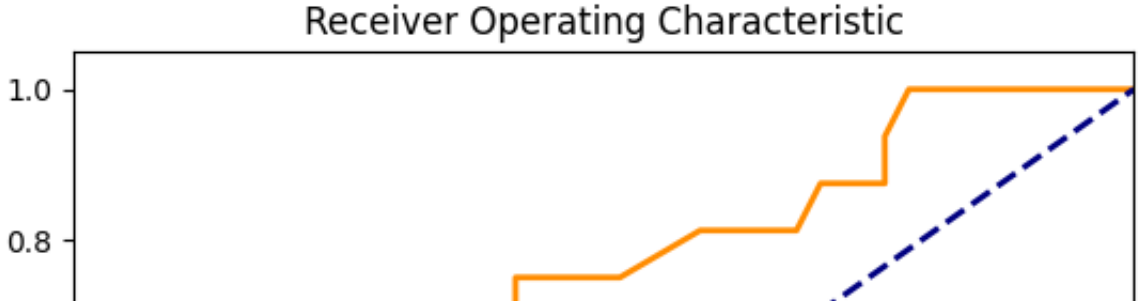

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.03787879 0.06818182 0.08333333
 0.09848485 0.11363636 0.12878788 0.15151515 0.16666667 0.17424242
 0.18181818 0.18181818 0.1969697  0.21969697 0.25       0.28030303
 0.28030303 0.3030303  0.31060606 0.33333333 0.35606061 0.37121212
 0.37878788 0.39393939 0.41666667 0.43181818 0.43939394 0.43939394
 0.4469697  0.47727273 0.50757576 0.58333333 0.61363636 0.62878788
 0.64393939 0.67424242 0.68181818 0.70454545 0.71212121 0.76515152
 0.78030303 0.78030303 0.79545455 0.83333333 0.89393939 1.        ]
TPR: [0.     0.     0.     0.     0.     0.     0.125  0.125  0.25   0.25
 0.25   0.3125 0.3125 0.375  0.375  0.375  0.375  0.375  0.4375 0.4375
 0.5625 0.5625 0.625  0.625  0.625  0.625  0.6875 0.6875 0.6875 0.75
 0.75   0.75   0.75   0.8125 0.8125 0.8125 0.8125 0.8125 0.8125 0.875
 0.875  0.875  0.875  0.9375 1.     1.     1.     1.     ]
ROC AUC: 0.646

Running evaluation with seed 42

Evaluating Decision Tree with seed 42...
Best parameters for Decision Tree: {'ccp_alpha': 0.002, 'criterion': 'entro
Training Metrics - Accuracy: 0.862, Sensitivity: 0.860, Specificity: 0.863,

Test Metrics for manual threshold 0.4:
Accuracy: 0.689, Sensitivity: 0.438, Specificity: 0.720, F1: 0.233, ROC AUC
```

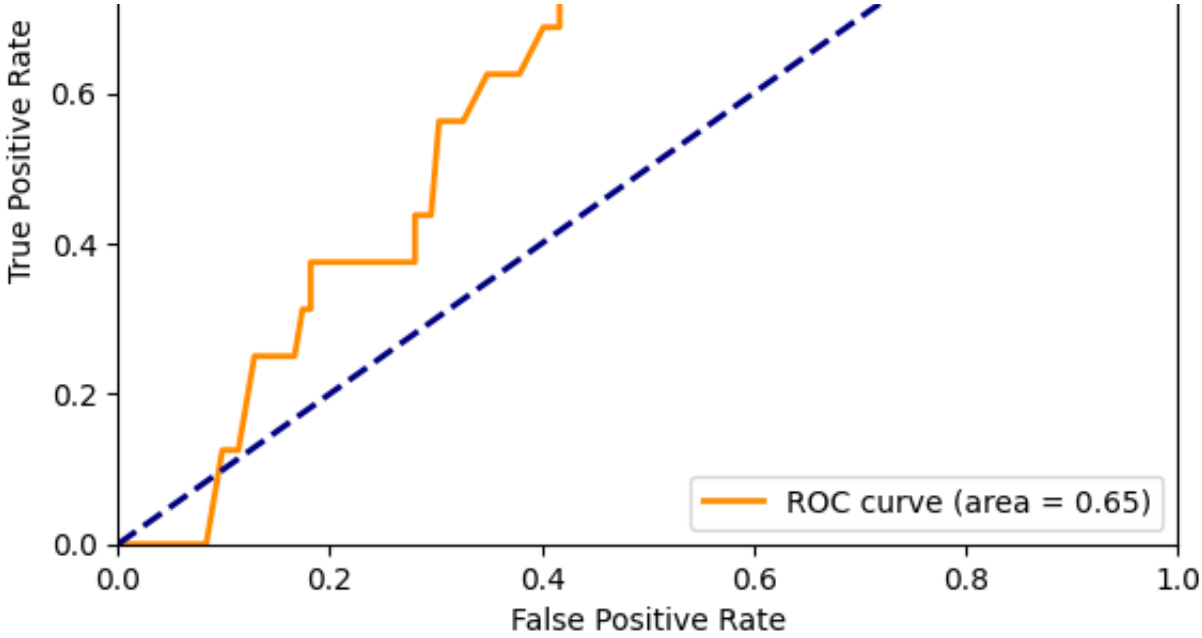
```
Threshold: 0.10, Metrics: {'Accuracy': 0.10810810810810811, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.21621621621621623, 'Sensitivity':
Threshold: 0.20, Metrics: {'Accuracy': 0.44594594594594594, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5337837837837838, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6081081081081081, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6486486486486487, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6891891891891891, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7094594594594594, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.7837837837837838, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Decision Tree
```




AGE
−0.250.000.25
−0.250.000.25
SHAP interaction val

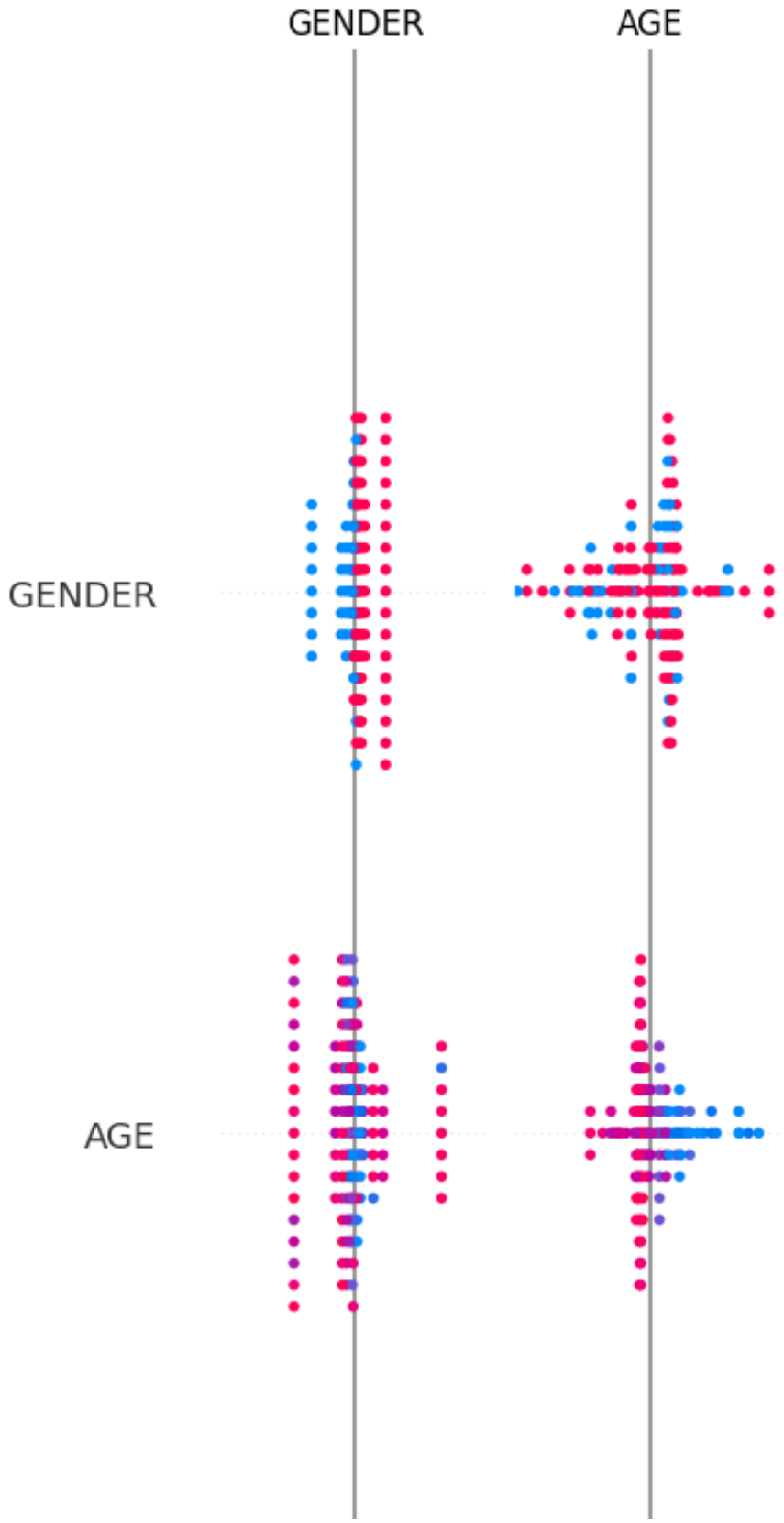

Confusion Matrix
Actual Success
Actual Failure
95
37
9
7
Predicted Success
Predicted Failure
90
80
70
60
50
40
30
20
10

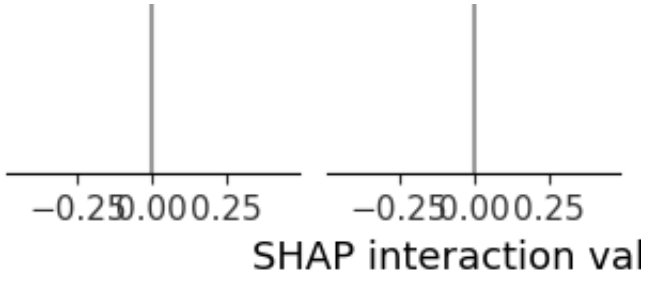

Receiver Operating Characteristic
1.0
0.8

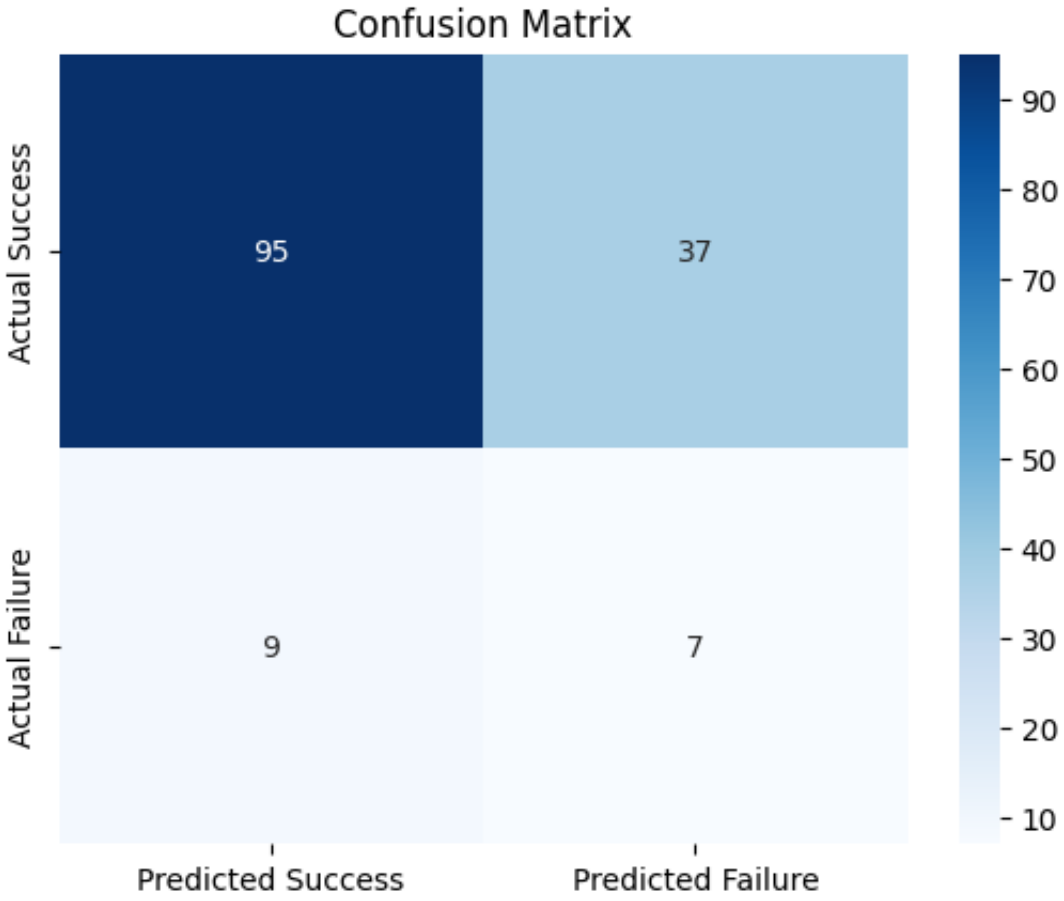


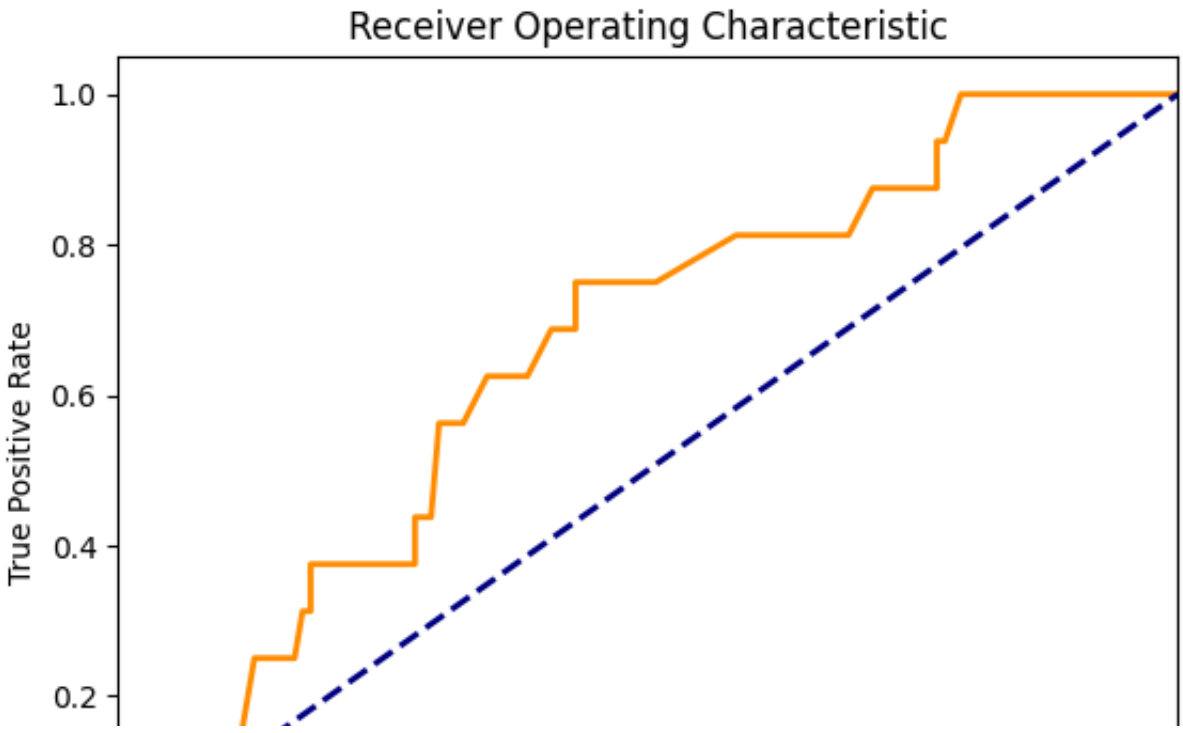

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.03787879 0.06818182 0.08333333
 0.09848485 0.11363636 0.12878788 0.15151515 0.16666667 0.17424242
 0.18181818 0.18181818 0.1969697  0.21969697 0.25       0.28030303
 0.28030303 0.29545455 0.3030303  0.32575758 0.34848485 0.37878788
 0.40151515 0.41666667 0.41666667 0.43939394 0.46969697 0.47727273
 0.50757576 0.51515152 0.59090909 0.61363636 0.62878788 0.64393939
 0.67424242 0.68181818 0.70454545 0.75757576 0.76515152 0.76515152
 0.78787879 0.82575758 0.85606061 0.86363636 0.89393939 1.        ]
TPR: [0.     0.     0.     0.     0.     0.     0.125  0.125  0.25   0.25
 0.25   0.3125 0.3125 0.375  0.375  0.375  0.375  0.375  0.4375 0.4375
 0.5625 0.5625 0.625  0.625  0.6875 0.6875 0.75   0.75   0.75   0.75
 0.75   0.75   0.8125 0.8125 0.8125 0.8125 0.8125 0.8125 0.875  0.875
 0.875  0.9375 1.     1.     1.     1.     1.     1.     ]
ROC AUC: 0.651

Running evaluation with seed 43

Evaluating Decision Tree with seed 43...
Best parameters for Decision Tree: {'ccp_alpha': 0.002, 'criterion': 'entro
Training Metrics - Accuracy: 0.862, Sensitivity: 0.860, Specificity: 0.863,

Test Metrics for manual threshold 0.4:
Accuracy: 0.689, Sensitivity: 0.438, Specificity: 0.720, F1: 0.233, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.10810810810810811, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.21621621621621623, 'Sensitivity':
Threshold: 0.20, Metrics: {'Accuracy': 0.44594594594594594, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5337837837837838, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6418918918918919, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6891891891891891, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7094594594594594, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
```

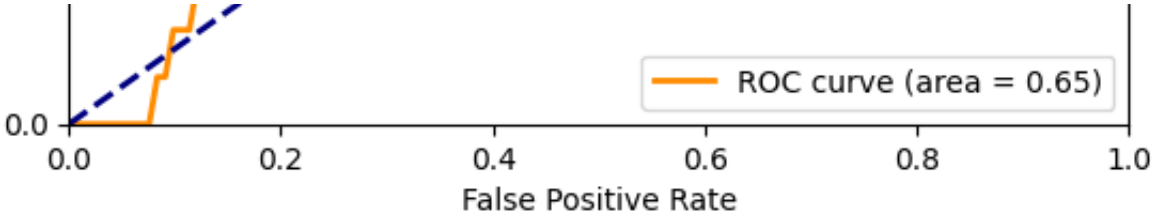
```
Threshold: 0.70, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Decision Tree
```




−0.25
0.00
0.25
−0.25
0.00
0.25
SHAP interaction val

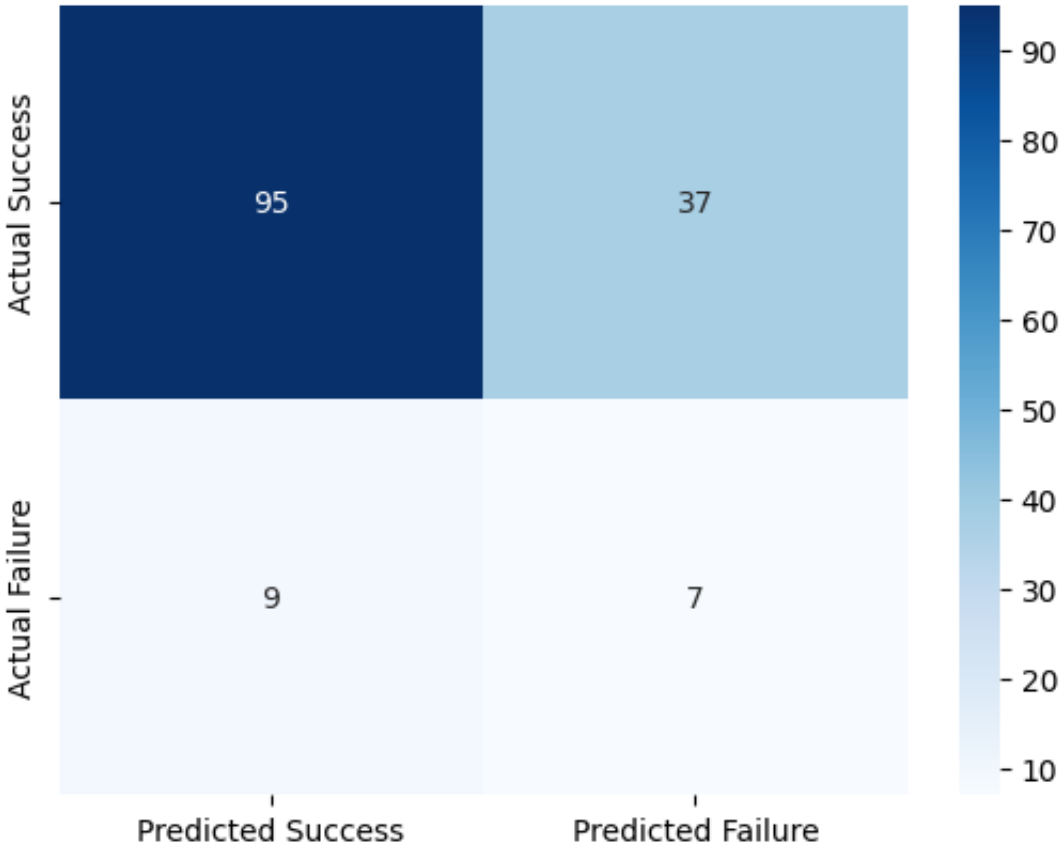

Confusion Matrix
Actual Success
Actual Failure
95
37
9
7
Predicted Success
Predicted Failure
90
80
70
60
50
40
30
20
10

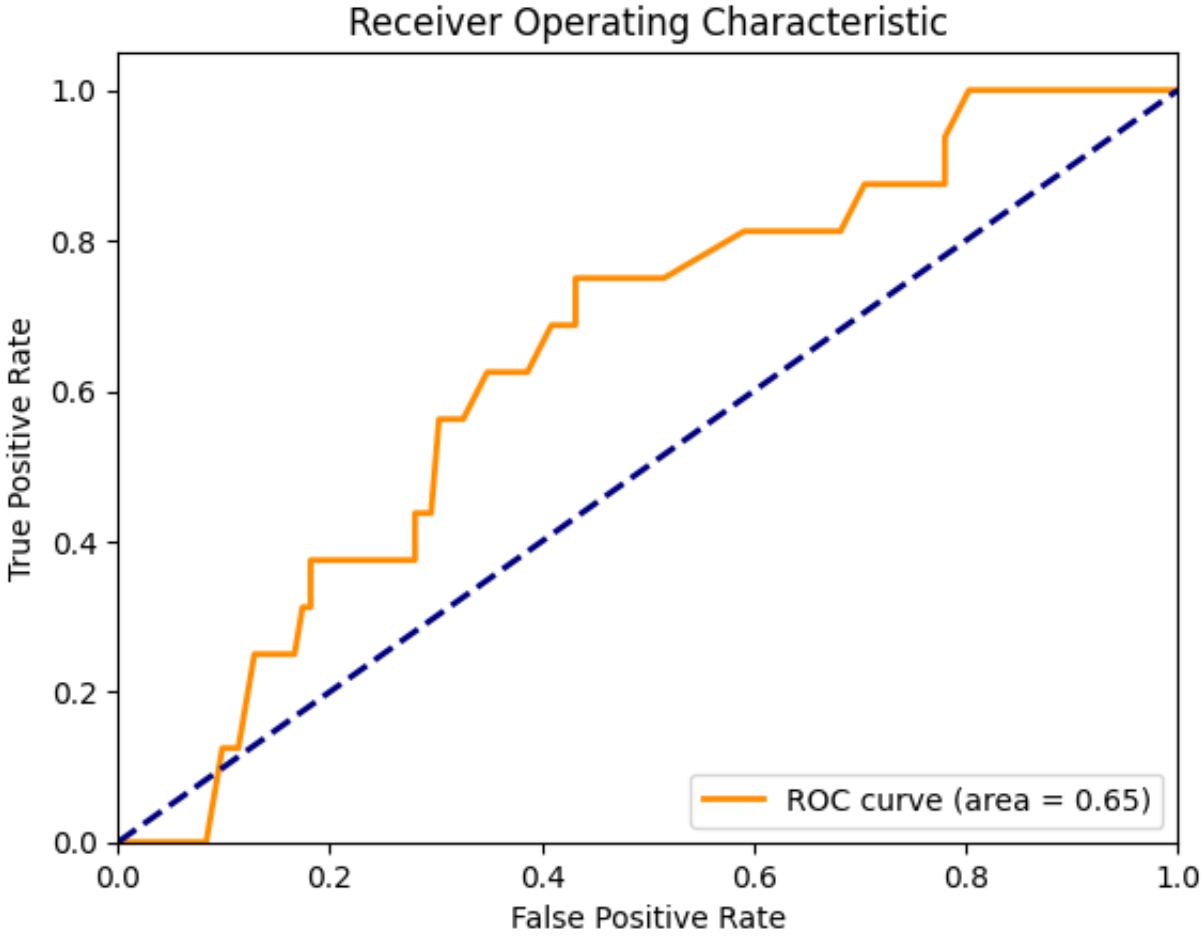

Receiver Operating Characteristic
True Positive Rate
1.0
0.8
0.6
0.4
0.2




```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.03787879 0.06818182 0.07575758
 0.08333333 0.09090909 0.09848485 0.11363636 0.12878788 0.15151515
 0.16666667 0.17424242 0.18181818 0.18181818 0.1969697  0.21969697
 0.25       0.28030303 0.28030303 0.29545455 0.3030303  0.32575758
 0.34848485 0.35606061 0.38636364 0.40909091 0.42424242 0.43181818
 0.43181818 0.4469697  0.47727273 0.50757576 0.58333333 0.62121212
 0.63636364 0.65151515 0.68181818 0.68939394 0.71212121 0.76515152
 0.77272727 0.77272727 0.78030303 0.79545455 0.82575758 0.85606061
 0.86363636 0.89393939 1.        ]
TPR: [0.     0.     0.     0.     0.     0.     0.0625 0.0625 0.125  0.125
 0.25   0.25   0.25   0.3125 0.3125 0.375  0.375  0.375  0.375  0.375
 0.4375 0.4375 0.5625 0.5625 0.625  0.625  0.625  0.6875 0.6875 0.6875
 0.75   0.75   0.75   0.75   0.8125 0.8125 0.8125 0.8125 0.8125 0.8125
 0.875  0.875  0.875  0.9375 0.9375 1.     1.     1.     1.     1.
 1.    ]
ROC AUC: 0.649

Running evaluation with seed 44

Evaluating Decision Tree with seed 44...
Best parameters for Decision Tree: {'ccp_alpha': 0.002, 'criterion': 'entro
Training Metrics - Accuracy: 0.862, Sensitivity: 0.860, Specificity: 0.863,

Test Metrics for manual threshold 0.4:
Accuracy: 0.689, Sensitivity: 0.438, Specificity: 0.720, F1: 0.233, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.10810810810810811, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.22972972972972974, 'Sensitivity':
Threshold: 0.20, Metrics: {'Accuracy': 0.44594594594594594, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5472972972972973, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6148648648648649, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6486486486486487, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6891891891891891, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7027027027027027, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.7837837837837838, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Decision Tree
```

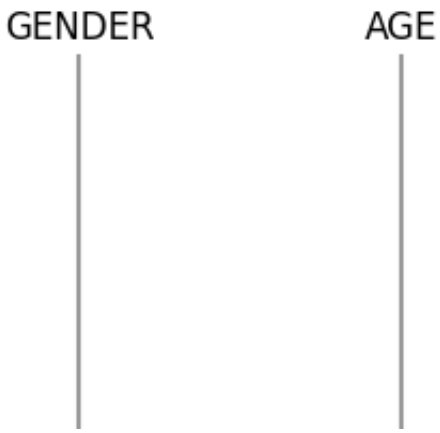
Confusion Matrix

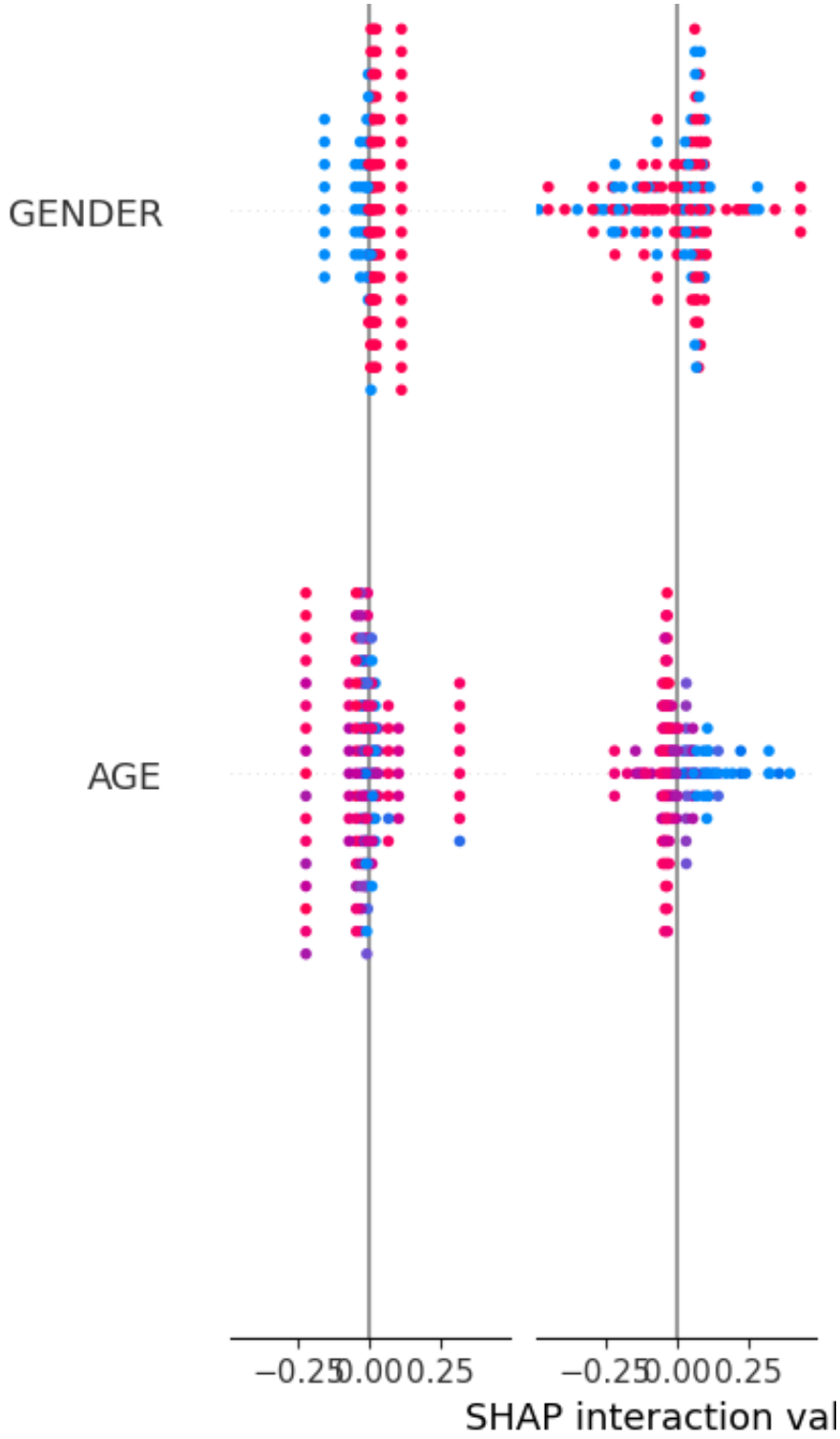


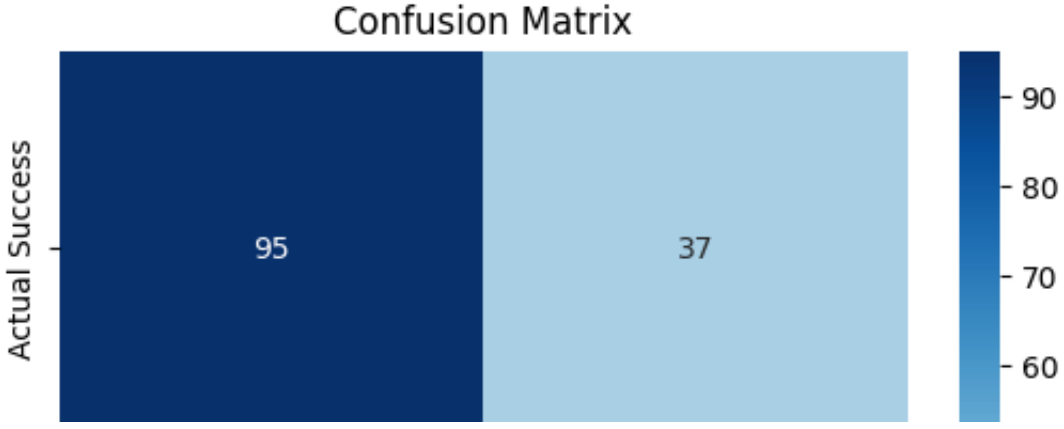


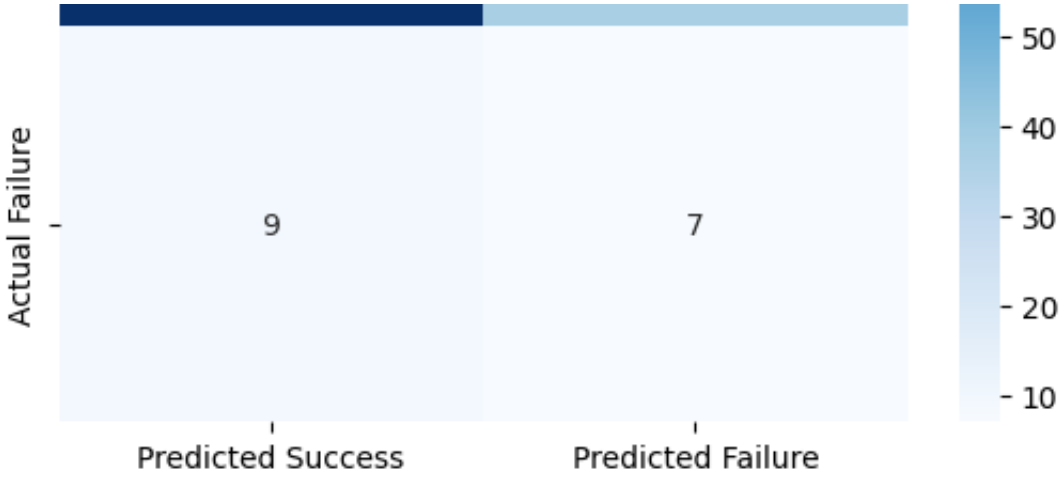

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.03787879 0.06818182 0.08333333
 0.09848485 0.11363636 0.12878788 0.15151515 0.16666667 0.17424242
```

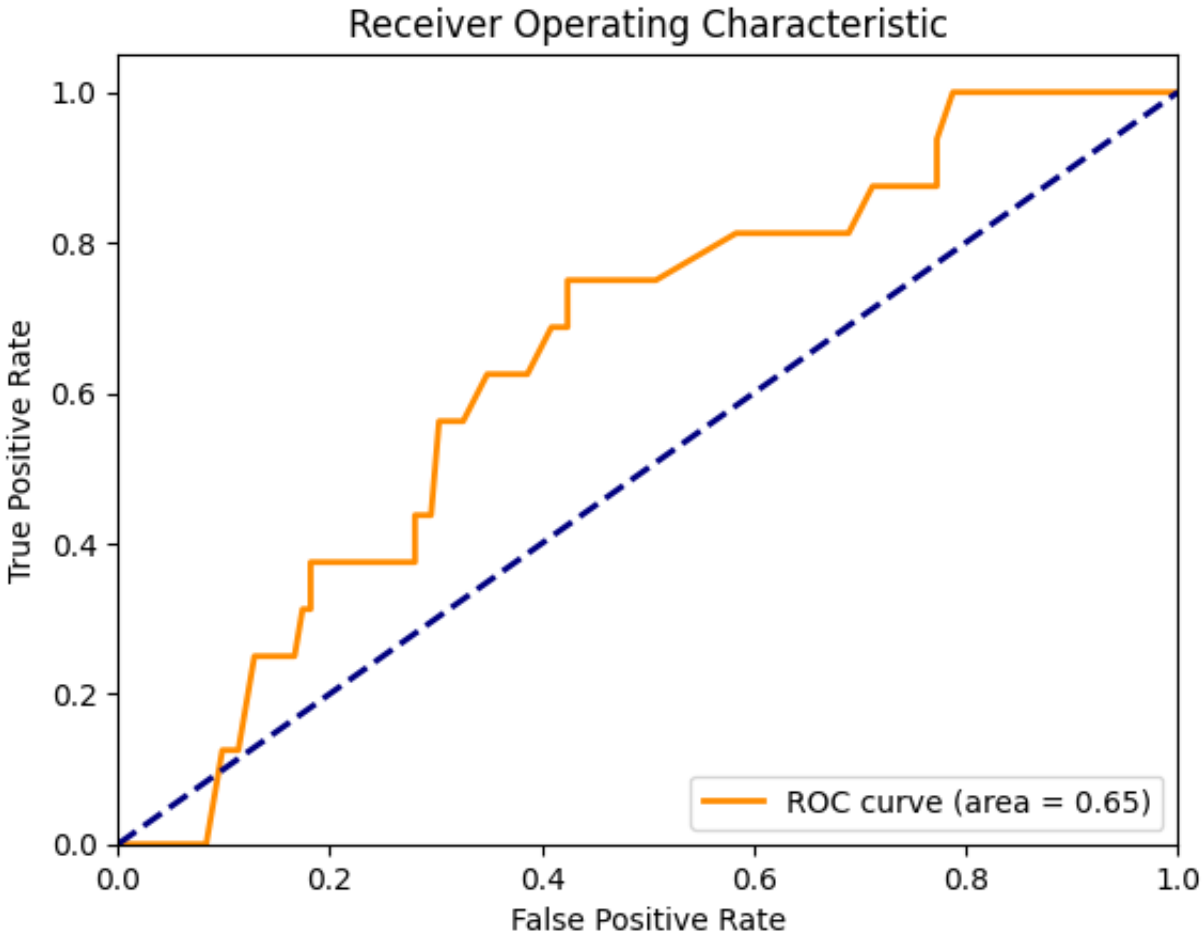
```
 0.18181818 0.18181818 0.1969697  0.21969697 0.25       0.28030303
 0.28030303 0.29545455 0.3030303  0.32575758 0.34848485 0.36363636
 0.37121212 0.38636364 0.40909091 0.42424242 0.43181818 0.43181818
 0.43939394 0.46969697 0.47727273 0.50757576 0.51515152 0.59090909
 0.60606061 0.62121212 0.64393939 0.67424242 0.68181818 0.70454545
 0.71212121 0.76515152 0.78030303 0.78030303 0.8030303  0.83333333
 0.89393939 1.        ]
TPR: [0.     0.     0.     0.     0.     0.     0.125  0.125  0.25   0.25
 0.25   0.3125 0.3125 0.375  0.375  0.375  0.375  0.375  0.4375 0.4375
 0.5625 0.5625 0.625  0.625  0.625  0.625  0.6875 0.6875 0.6875 0.75
 0.75   0.75   0.75   0.75   0.75   0.8125 0.8125 0.8125 0.8125 0.8125
 0.8125 0.875  0.875  0.875  0.875  0.9375 1.     1.     1.     1.    ]
ROC AUC: 0.647

Running evaluation with seed 45

Evaluating Decision Tree with seed 45...
Best parameters for Decision Tree: {'ccp_alpha': 0.002, 'criterion': 'entro
Training Metrics - Accuracy: 0.862, Sensitivity: 0.860, Specificity: 0.863,

Test Metrics for manual threshold 0.4:
Accuracy: 0.689, Sensitivity: 0.438, Specificity: 0.720, F1: 0.233, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.10810810810810811, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.21621621621621623, 'Sensitivity':
Threshold: 0.20, Metrics: {'Accuracy': 0.4527027027027027, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.5472972972972973, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6486486486486487, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6891891891891891, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7094594594594594, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.7905405405405406, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Decision Tree
```




GENDER
AGE
−0.250.000.25
−0.250.000.25
SHAP interaction val

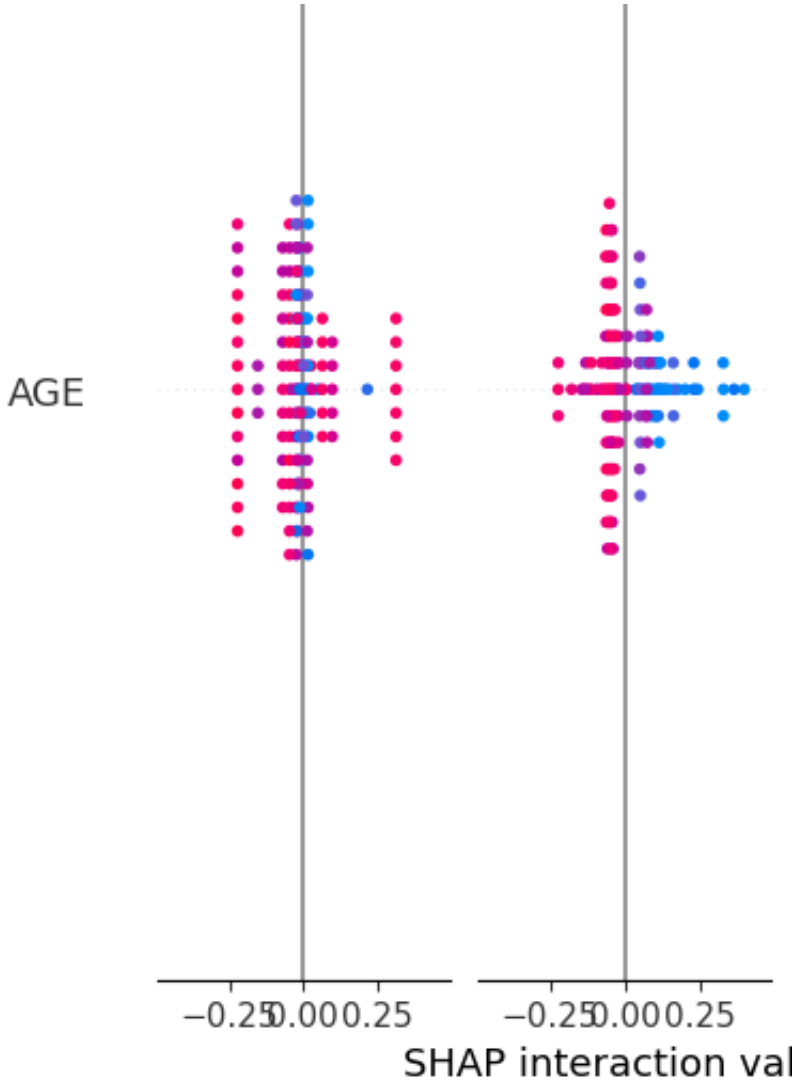

Confusion Matrix
Actual Success
95
37
90
80
70
60

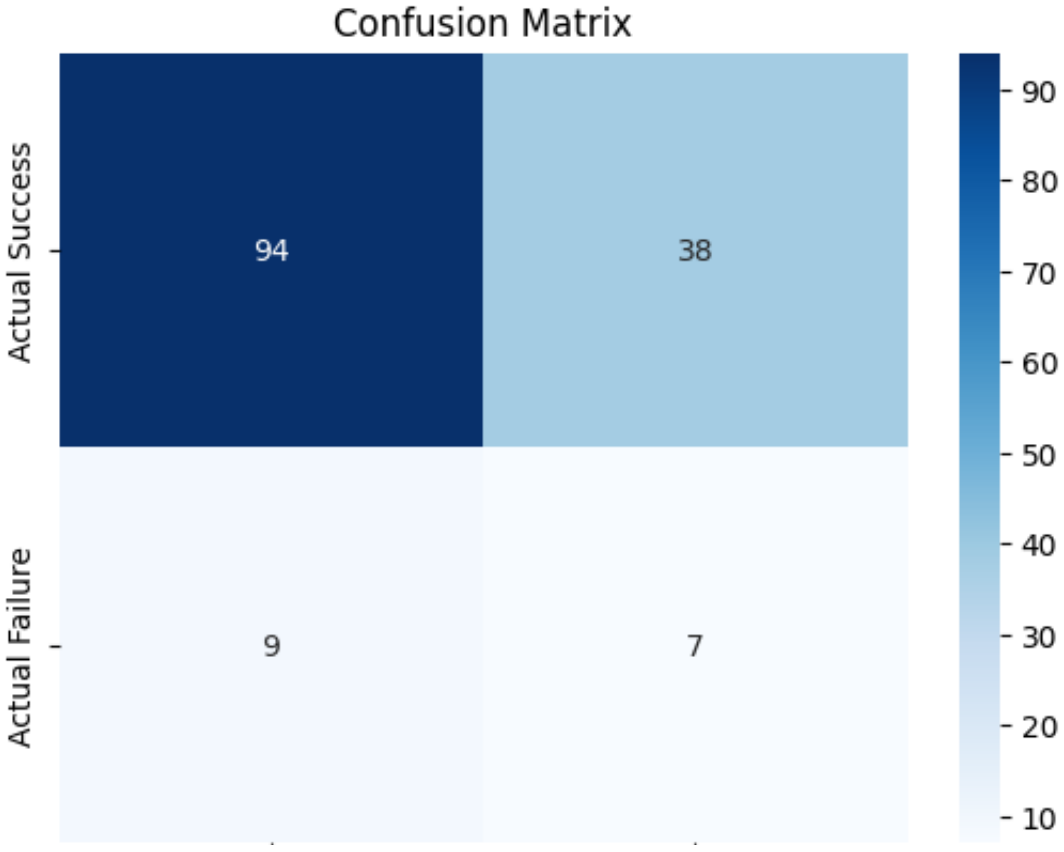


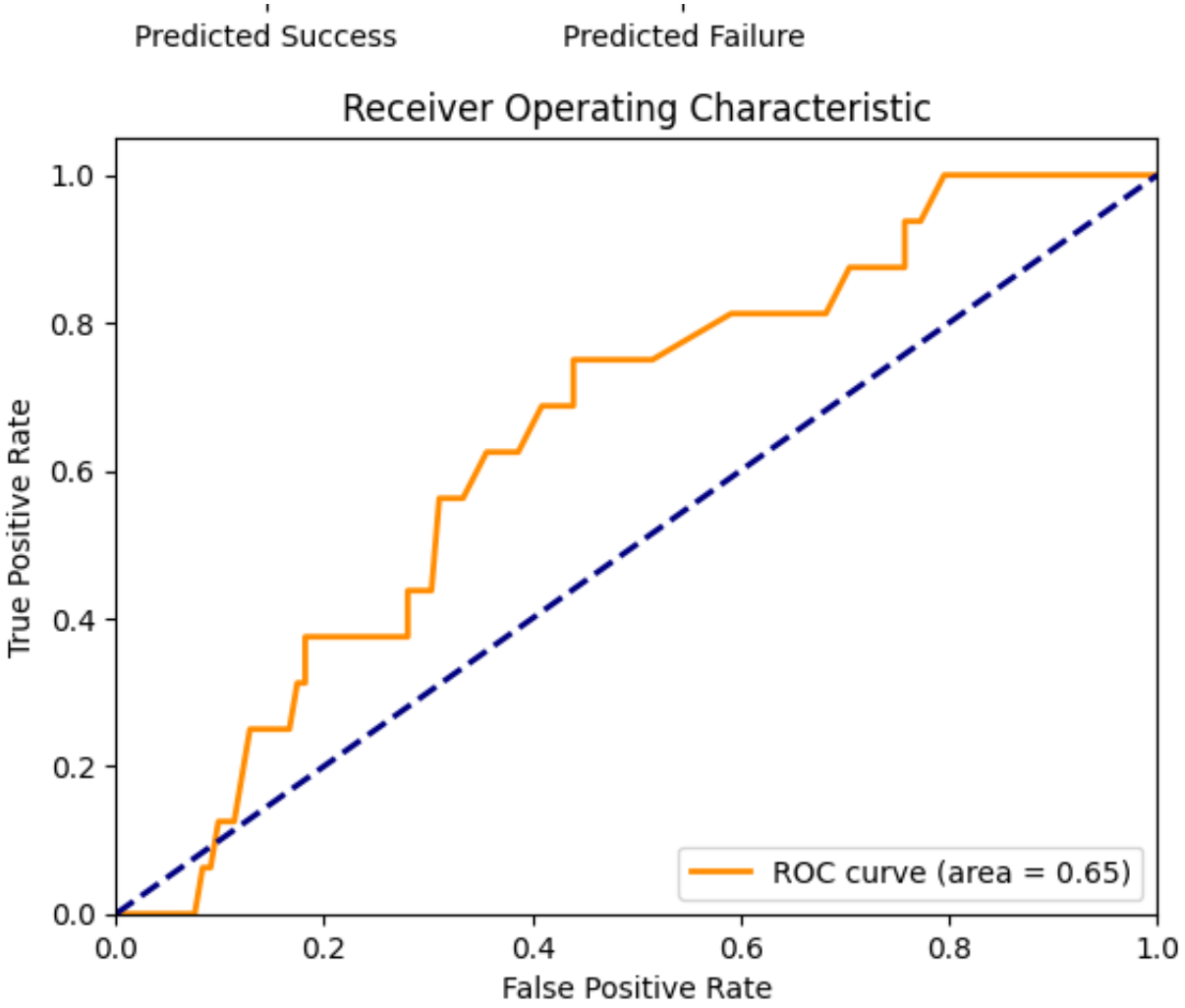



```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.03787879 0.06818182 0.08333333
 0.09848485 0.11363636 0.12878788 0.15151515 0.16666667 0.17424242
 0.18181818 0.18181818 0.1969697  0.21969697 0.25       0.28030303
 0.28030303 0.29545455 0.3030303  0.32575758 0.34848485 0.35606061
 0.38636364 0.40909091 0.42424242 0.42424242 0.4469697  0.47727273
 0.50757576 0.58333333 0.62121212 0.63636364 0.65151515 0.68181818
 0.68939394 0.71212121 0.76515152 0.77272727 0.77272727 0.78787879
 0.82575758 0.85606061 0.86363636 0.89393939 1.        ]
TPR: [0.     0.     0.     0.     0.     0.     0.125  0.125  0.25   0.25
 0.25   0.3125 0.3125 0.375  0.375  0.375  0.375  0.375  0.4375 0.4375
 0.5625 0.5625 0.625  0.625  0.625  0.6875 0.6875 0.75   0.75   0.75
 0.75   0.8125 0.8125 0.8125 0.8125 0.8125 0.8125 0.875  0.875  0.875
 0.9375 1.     1.     1.     1.     1.     1.     1.    ]
```

```
ROC AUC: 0.649

Running evaluation with seed 46

Evaluating Decision Tree with seed 46...
Best parameters for Decision Tree: {'ccp_alpha': 0.002, 'criterion': 'entro
Training Metrics - Accuracy: 0.862, Sensitivity: 0.860, Specificity: 0.863,

Test Metrics for manual threshold 0.4:
Accuracy: 0.682, Sensitivity: 0.438, Specificity: 0.712, F1: 0.230, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.10810810810810811, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.21621621621621623, 'Sensitivity':
Threshold: 0.20, Metrics: {'Accuracy': 0.4391891891891892, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.5337837837837838, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6418918918918919, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6824324324324325, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7094594594594594, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Decision Tree
```

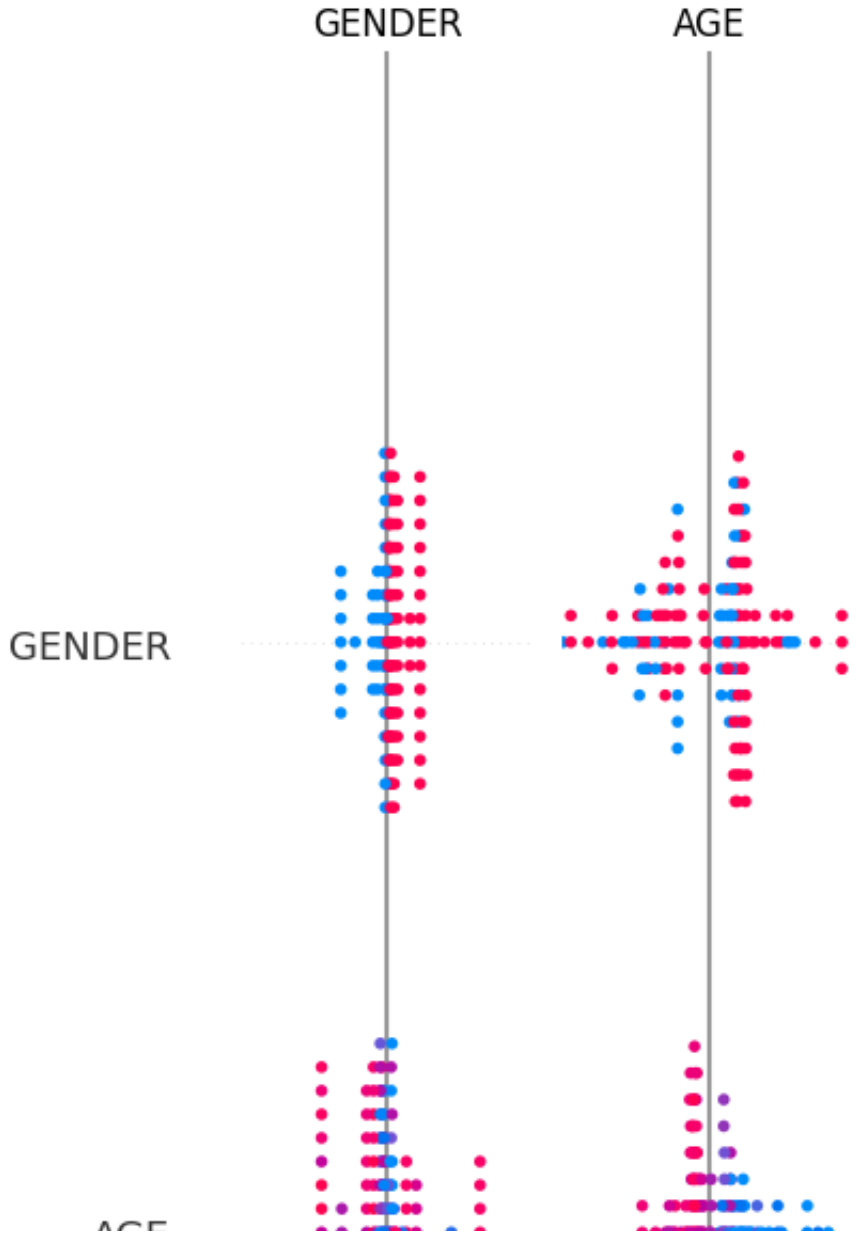

AGE
−0.250.000.25
−0.250.000.25
SHAP interaction val

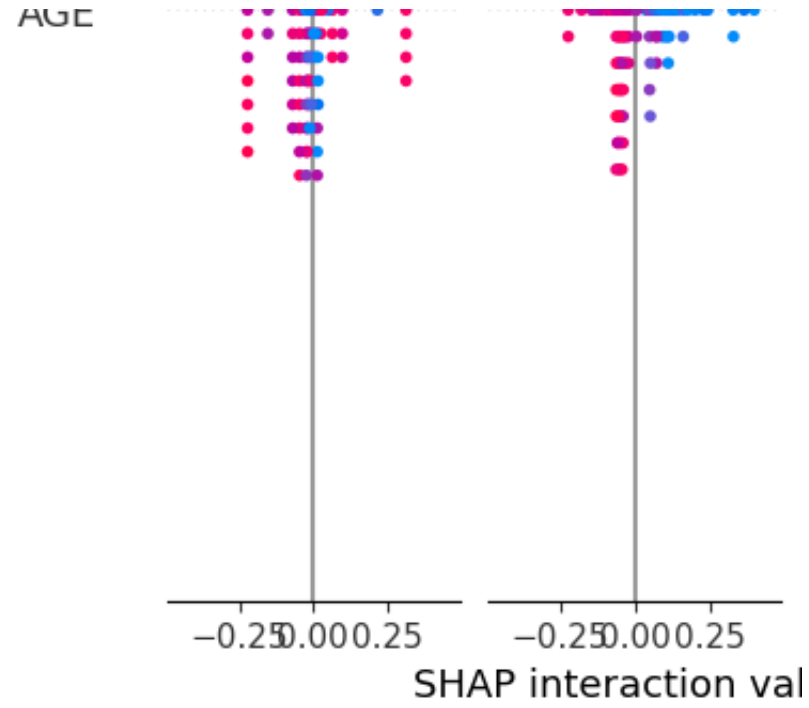

Confusion Matrix
Actual Success
Actual Failure
94
38
9
7
90
80
70
60
50
40
30
20
10

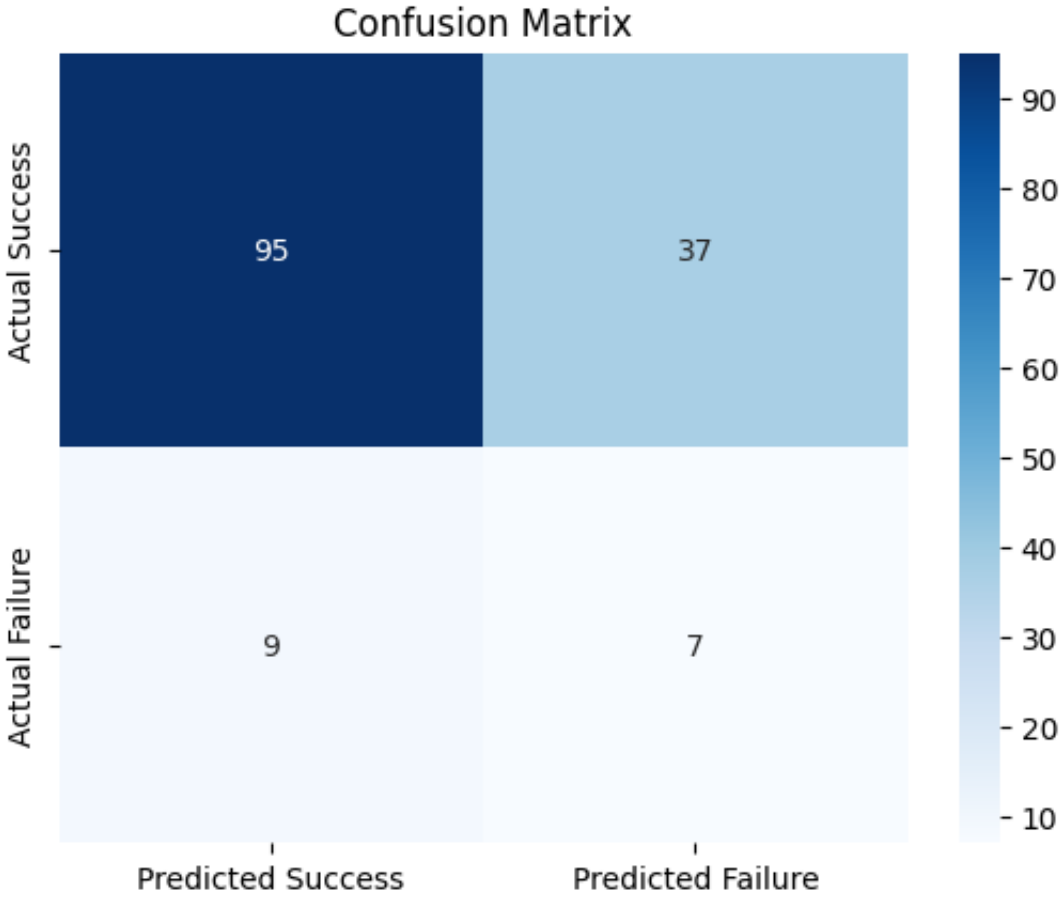


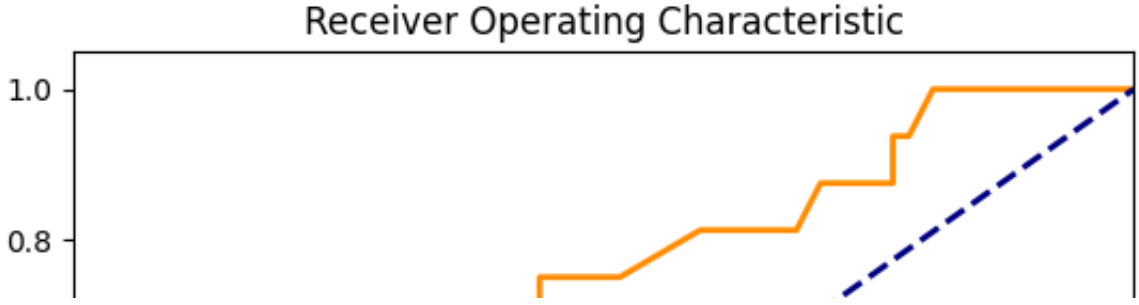

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.03787879 0.06818182 0.07575758
 0.08333333 0.09090909 0.09848485 0.11363636 0.12878788 0.15151515
 0.16666667 0.17424242 0.18181818 0.18181818 0.1969697  0.21969697
 0.25       0.28030303 0.28030303 0.3030303  0.31060606 0.33333333
 0.35606061 0.38636364 0.40909091 0.42424242 0.43939394 0.43939394
 0.4469697  0.47727273 0.50757576 0.51515152 0.59090909 0.61363636
 0.62878788 0.64393939 0.67424242 0.68181818 0.70454545 0.75757576
 0.75757576 0.77272727 0.79545455 0.82575758 0.85606061 0.86363636
 0.89393939 1.        ]
TPR: [0.     0.     0.     0.     0.     0.     0.0625 0.0625 0.125  0.125
 0.25   0.25   0.25   0.3125 0.3125 0.375  0.375  0.375  0.375  0.375
 0.4375 0.4375 0.5625 0.5625 0.625  0.625  0.6875 0.6875 0.6875 0.75
 0.75   0.75   0.75   0.75   0.8125 0.8125 0.8125 0.8125 0.8125 0.8125
 0.875  0.875  0.9375 0.9375 1.     1.     1.     1.     1.     1.    ]
ROC AUC: 0.648

Running evaluation with seed 47

Evaluating Decision Tree with seed 47...
Best parameters for Decision Tree: {'ccp_alpha': 0.002, 'criterion': 'entro
Training Metrics - Accuracy: 0.862, Sensitivity: 0.860, Specificity: 0.863,

Test Metrics for manual threshold 0.4:
Accuracy: 0.689, Sensitivity: 0.438, Specificity: 0.720, F1: 0.233, ROC AUC
```

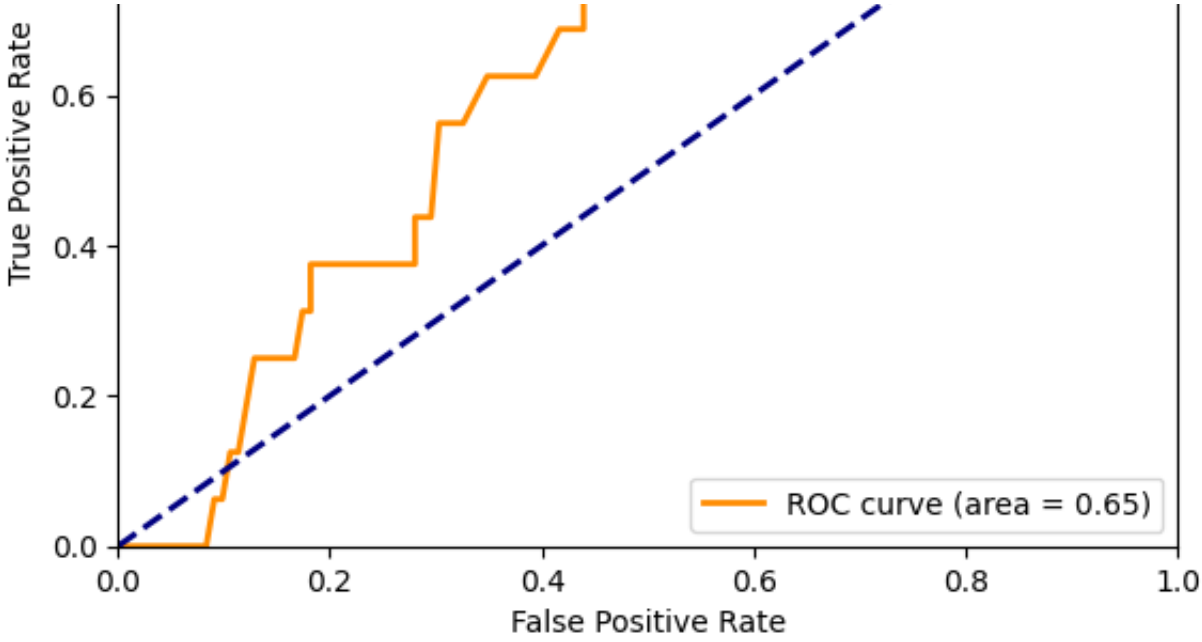
```
Threshold: 0.10, Metrics: {'Accuracy': 0.10810810810810811, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.21621621621621623, 'Sensitivity':
Threshold: 0.20, Metrics: {'Accuracy': 0.44594594594594594, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5337837837837838, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.5945945945945946, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6418918918918919, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6891891891891891, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7027027027027027, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Decision Tree
```




AGE
−0.250.000.25
−0.250.000.25
SHAP interaction val

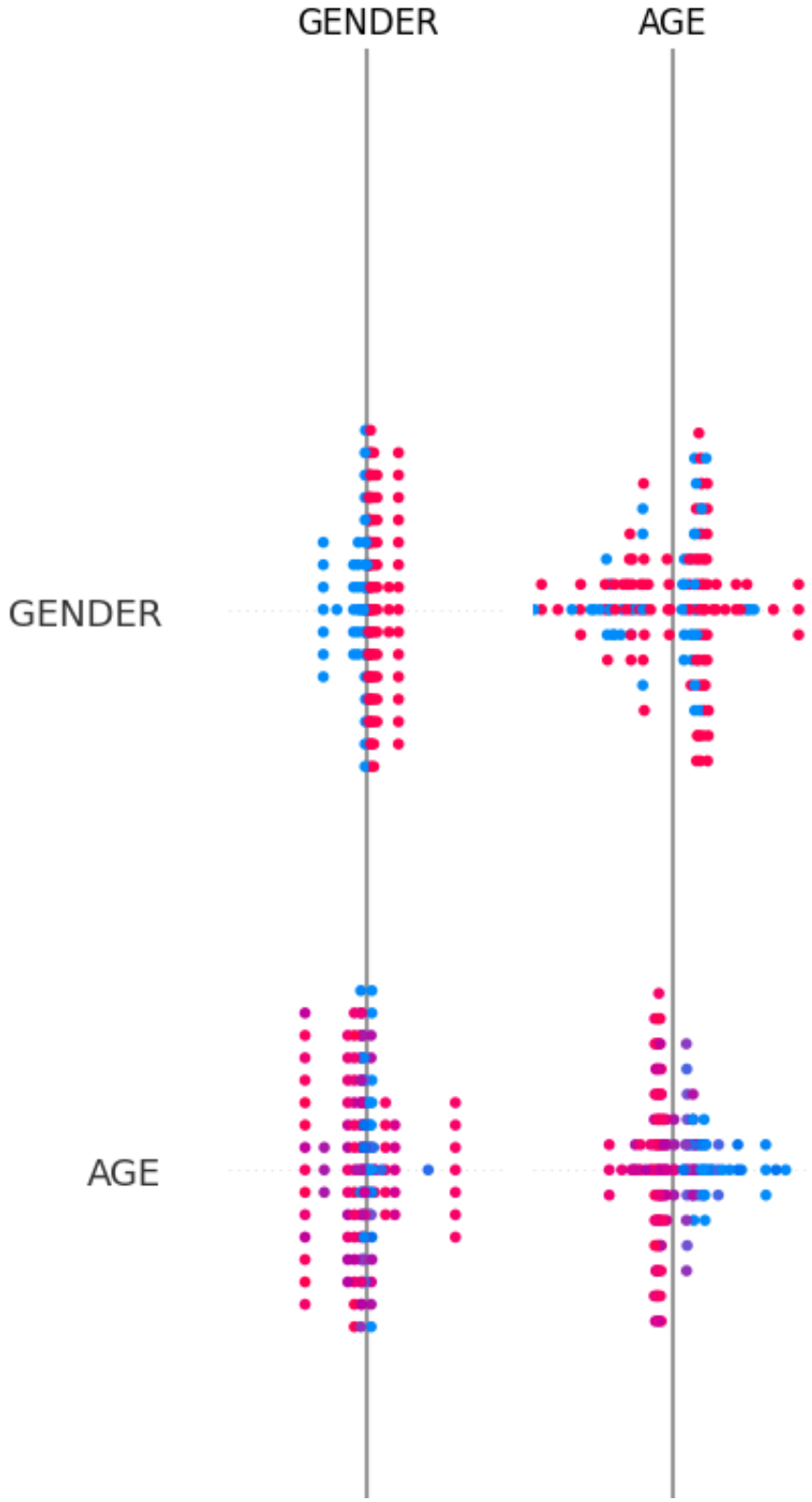

Confusion Matrix
Actual Success
Actual Failure
95
37
9
7
Predicted Success
Predicted Failure
90
80
70
60
50
40
30
20
10

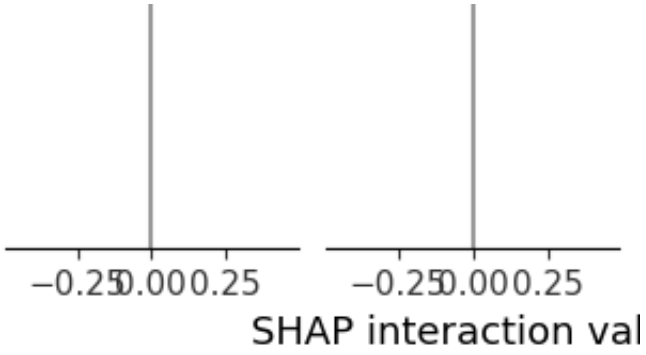

Receiver Operating Characteristic
1.0
0.8

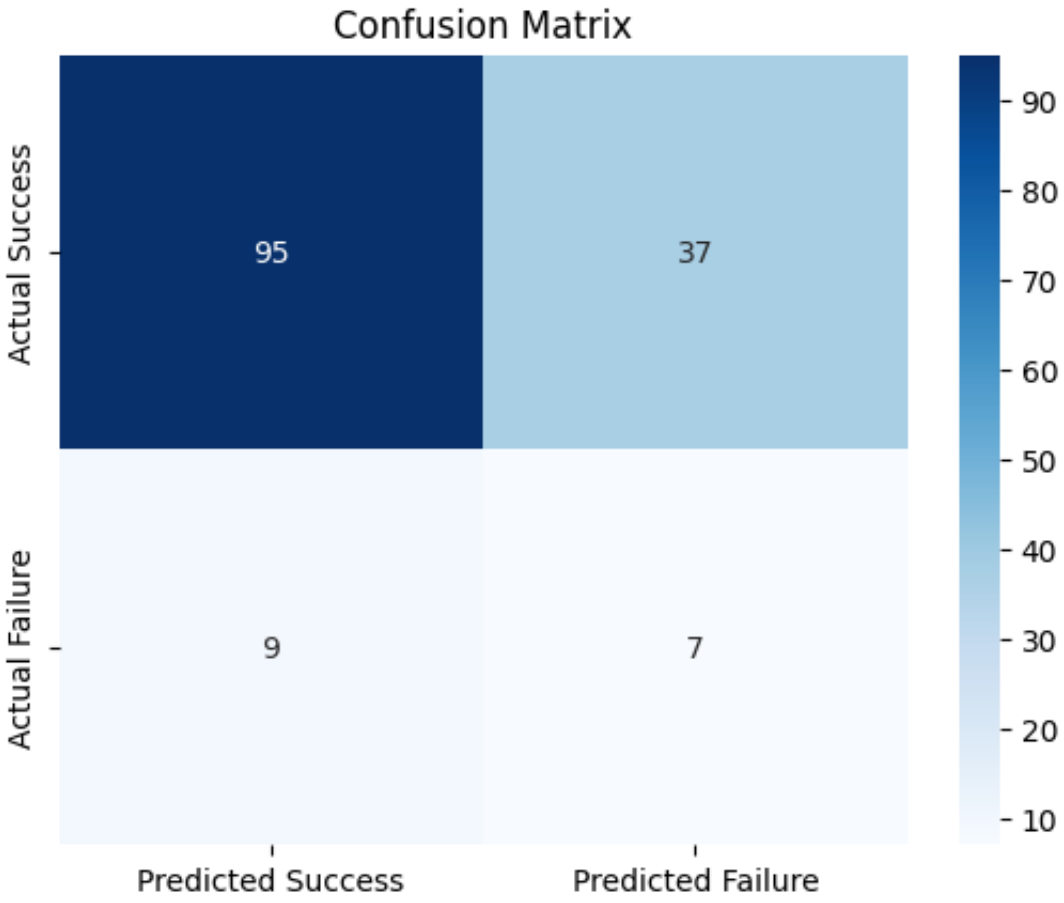


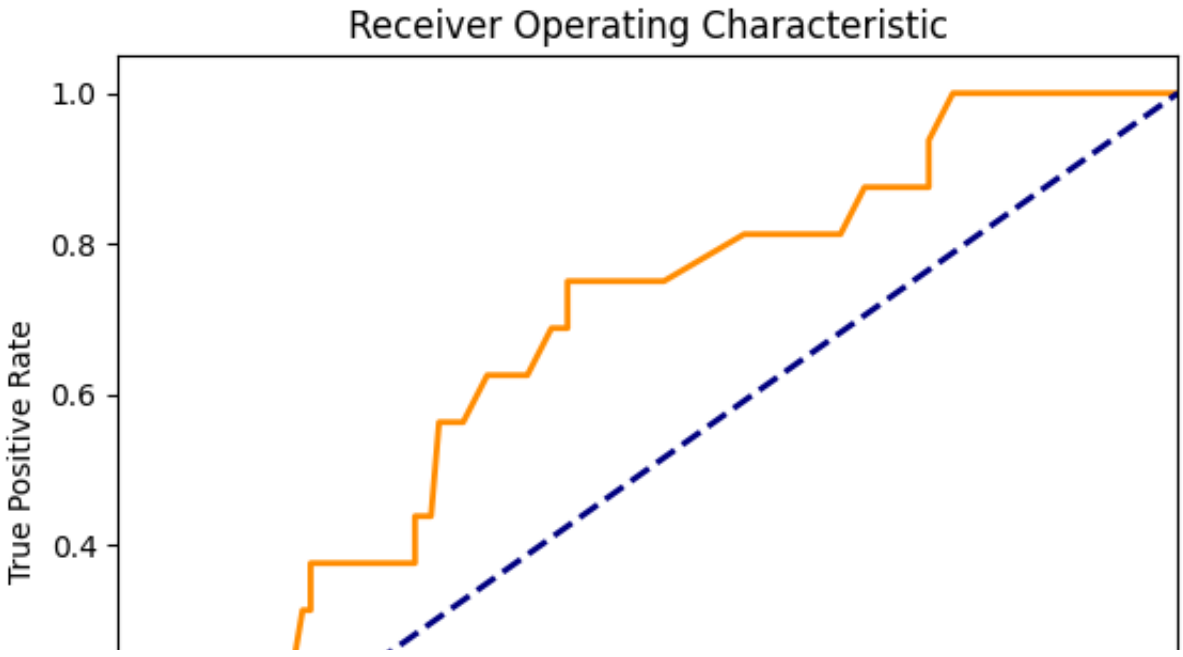

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.03787879 0.06818182 0.08333333
 0.09090909 0.09848485 0.10606061 0.11363636 0.12878788 0.15151515
 0.16666667 0.17424242 0.18181818 0.18181818 0.1969697  0.21969697
 0.25       0.28030303 0.28030303 0.29545455 0.3030303  0.32575758
 0.34848485 0.35606061 0.37121212 0.37878788 0.39393939 0.41666667
 0.43181818 0.43939394 0.43939394 0.4469697  0.47727273 0.50757576
 0.51515152 0.59090909 0.60606061 0.62121212 0.64393939 0.67424242
 0.68181818 0.70454545 0.71969697 0.77272727 0.77272727 0.78787879
 0.81060606 0.83333333 0.89393939 1.        ]
TPR: [0.     0.     0.     0.     0.     0.     0.0625 0.0625 0.125  0.125
 0.25   0.25   0.25   0.3125 0.3125 0.375  0.375  0.375  0.375  0.375
 0.4375 0.4375 0.5625 0.5625 0.625  0.625  0.625  0.625  0.625  0.6875
 0.6875 0.6875 0.75   0.75   0.75   0.75   0.75   0.8125 0.8125 0.8125
 0.8125 0.8125 0.8125 0.875  0.875  0.875  0.9375 0.9375 1.     1.
 1.     1.     ]
ROC AUC: 0.646

Running evaluation with seed 48

Evaluating Decision Tree with seed 48...
Best parameters for Decision Tree: {'ccp_alpha': 0.002, 'criterion': 'entro
Training Metrics - Accuracy: 0.862, Sensitivity: 0.860, Specificity: 0.863,

Test Metrics for manual threshold 0.4:
Accuracy: 0.689, Sensitivity: 0.438, Specificity: 0.720, F1: 0.233, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.10810810810810811, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.21621621621621623, 'Sensitivity':
Threshold: 0.20, Metrics: {'Accuracy': 0.44594594594594594, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5337837837837838, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6418918918918919, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6891891891891891, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7094594594594594, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
```

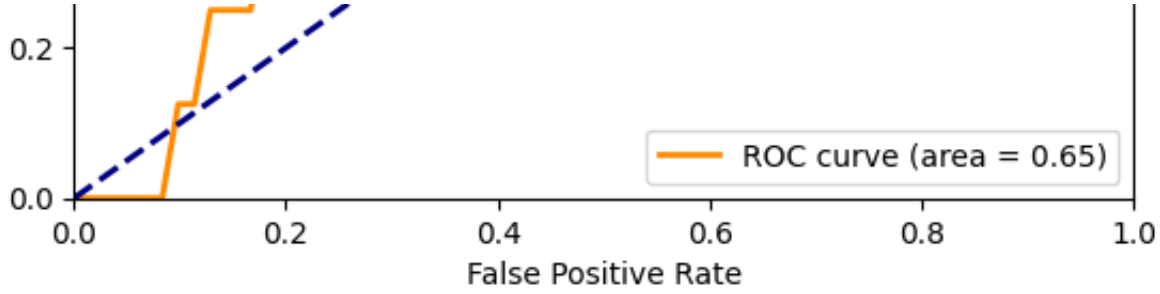
```
Threshold: 0.60, Metrics: {'Accuracy': 0.7837837837837838, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Decision Tree
```



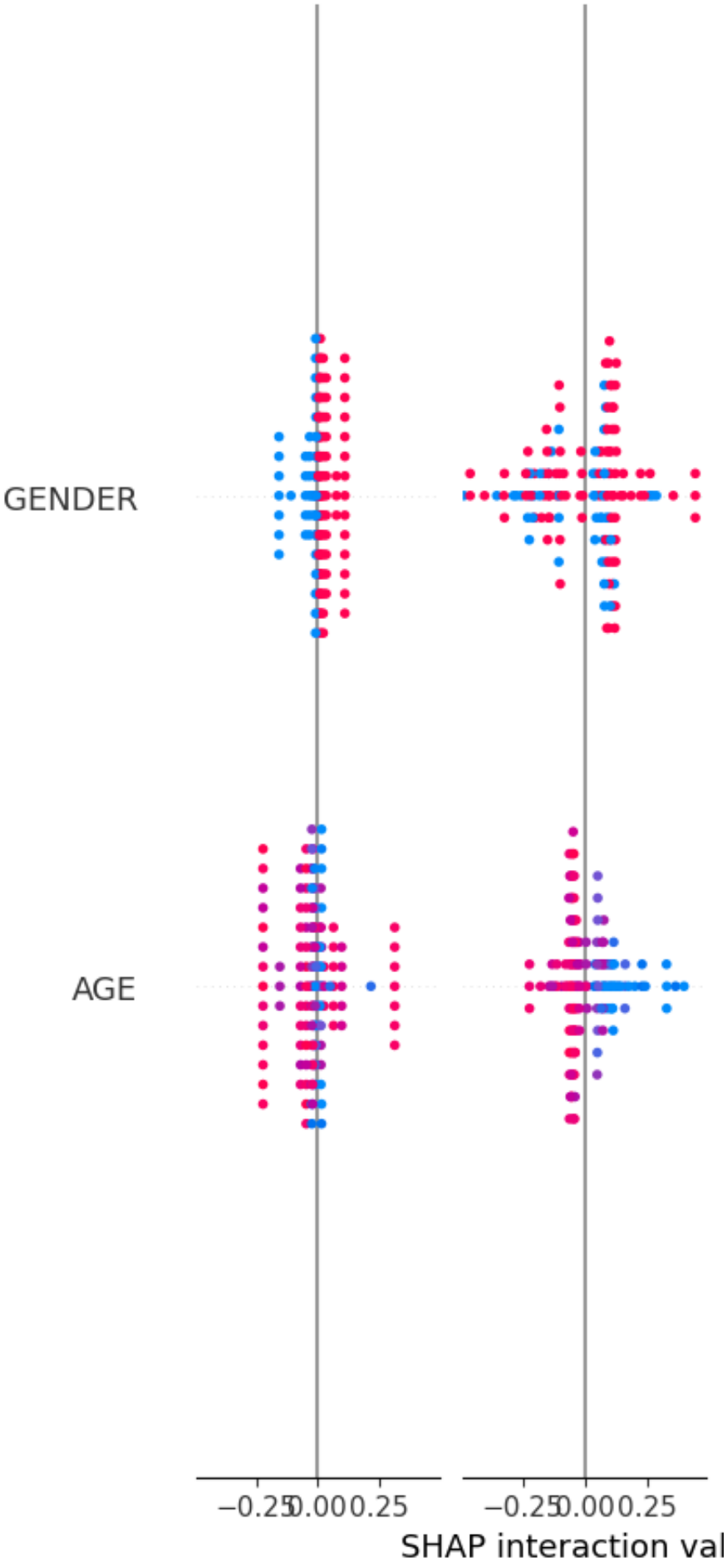

−0.250.000.25
−0.250.000.25
SHAP interaction val


Confusion Matrix
Actual Success
Actual Failure
95
37
9
7
Predicted Success
Predicted Failure
90
80
70
60
50
40
30
20
10

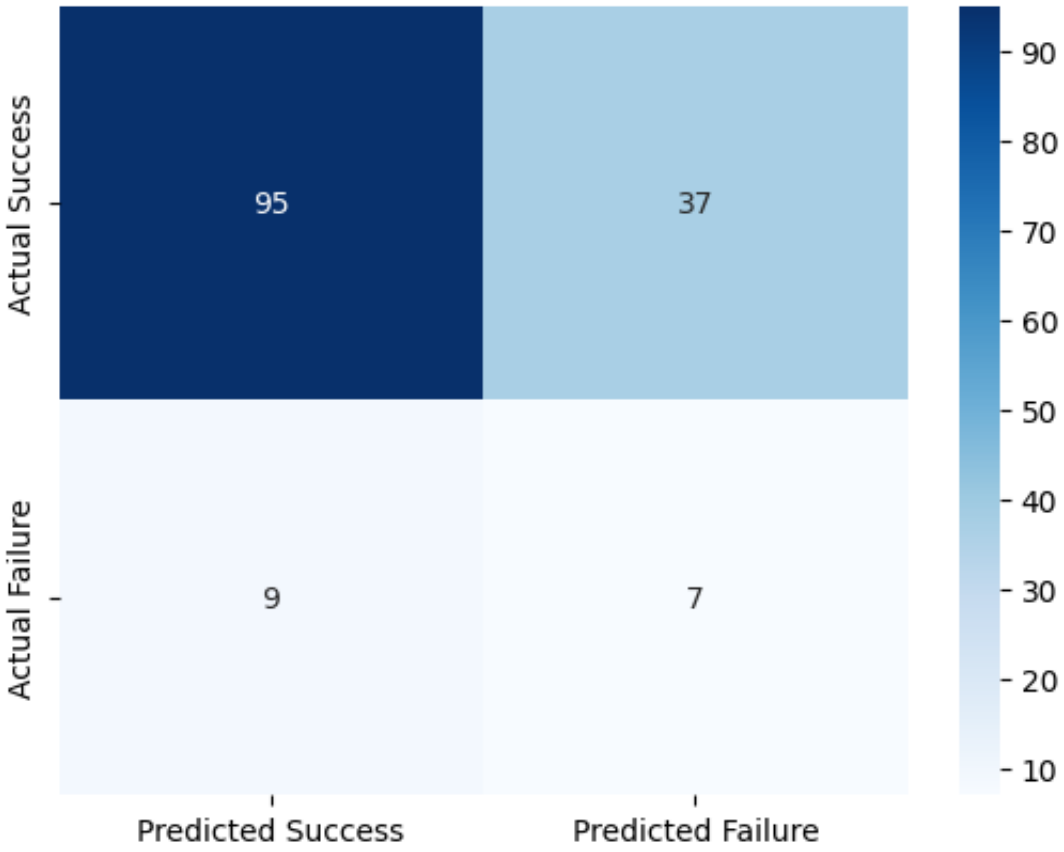

Receiver Operating Characteristic
True Positive Rate
1.0
0.8
0.6
0.4

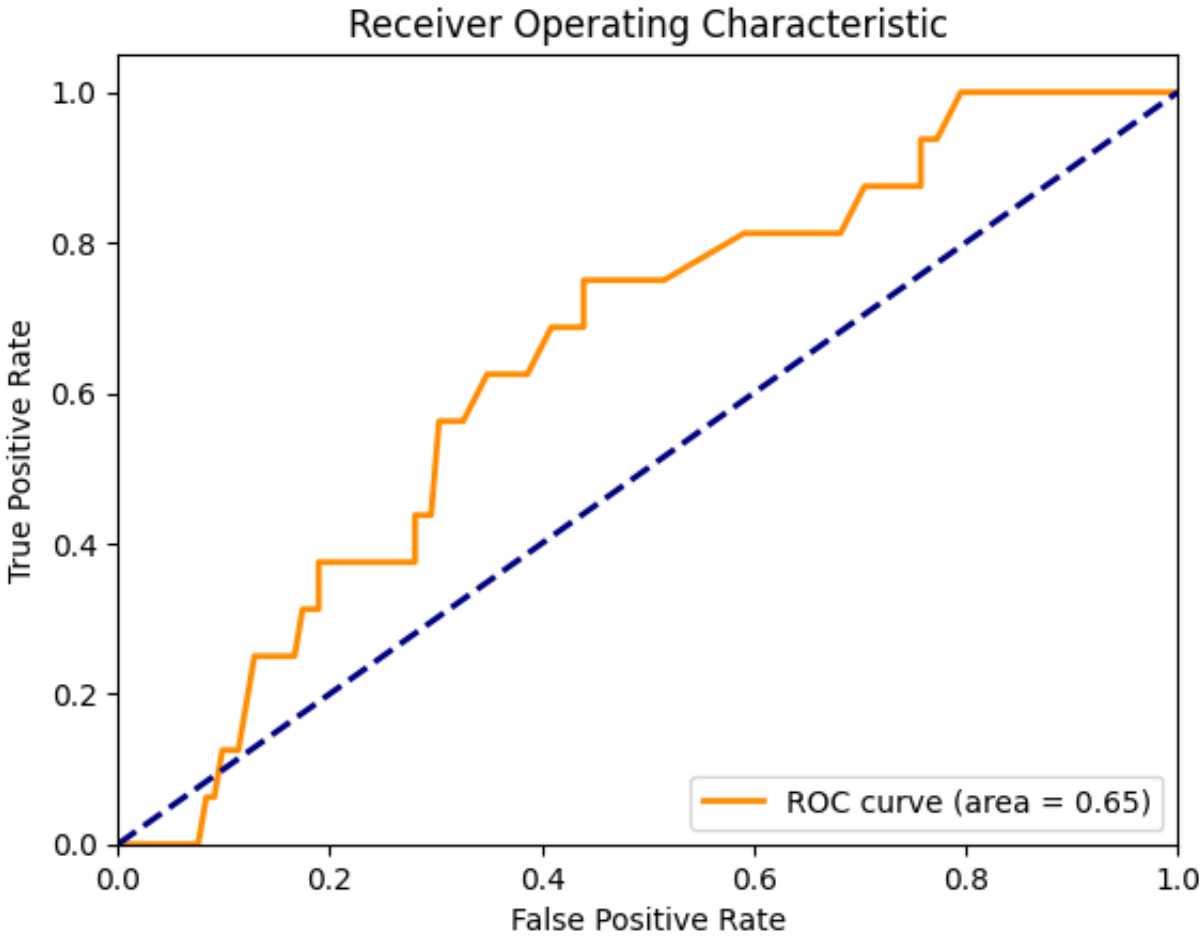



```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.03787879 0.06818182 0.08333333
 0.09848485 0.11363636 0.12878788 0.15151515 0.16666667 0.17424242
 0.18181818 0.18181818 0.1969697  0.21969697 0.25       0.28030303
 0.28030303 0.29545455 0.3030303  0.32575758 0.34848485 0.35606061
 0.38636364 0.40909091 0.42424242 0.42424242 0.4469697  0.47727273
 0.50757576 0.51515152 0.59090909 0.61363636 0.62878788 0.64393939
 0.67424242 0.68181818 0.70454545 0.75757576 0.76515152 0.76515152
 0.78787879 0.82575758 0.85606061 0.86363636 0.89393939 1.        ]
TPR: [0.     0.     0.     0.     0.     0.     0.125  0.125  0.25   0.25
 0.25   0.3125 0.3125 0.375  0.375  0.375  0.375  0.375  0.4375 0.4375
 0.5625 0.5625 0.625  0.625  0.625  0.6875 0.6875 0.75   0.75   0.75
 0.75   0.75   0.8125 0.8125 0.8125 0.8125 0.8125 0.8125 0.875  0.875
 0.875  0.9375 1.     1.     1.     1.     1.     1.     ]
ROC AUC: 0.650

Running evaluation with seed 49

Evaluating Decision Tree with seed 49...
Best parameters for Decision Tree: {'ccp_alpha': 0.002, 'criterion': 'entro
Training Metrics - Accuracy: 0.862, Sensitivity: 0.860, Specificity: 0.863,

Test Metrics for manual threshold 0.4:
Accuracy: 0.689, Sensitivity: 0.438, Specificity: 0.720, F1: 0.233, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.10810810810810811, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.20270270270270271, 'Sensitivity':
Threshold: 0.20, Metrics: {'Accuracy': 0.4391891891891892, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.5337837837837838, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6418918918918919, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6891891891891891, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7094594594594594, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Decision Tree
```



Confusion Matrix






```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.03787879 0.06818182 0.07575758
 0.08333333 0.09090909 0.09848485 0.11363636 0.12878788 0.15151515
```

```
 0.16666667 0.17424242 0.18939394 0.18939394 0.1969697  0.21969697
 0.25       0.28030303 0.28030303 0.29545455 0.3030303  0.32575758
 0.34848485 0.35606061 0.38636364 0.40909091 0.42424242 0.43939394
 0.43939394 0.4469697  0.47727273 0.50757576 0.51515152 0.59090909
 0.61363636 0.62878788 0.64393939 0.67424242 0.68181818 0.70454545
 0.75757576 0.75757576 0.77272727 0.79545455 0.82575758 0.85606061
 0.86363636 0.89393939 1.        ]
TPR: [0.     0.     0.     0.     0.     0.     0.0625 0.0625 0.125  0.125
 0.25   0.25   0.25   0.3125 0.3125 0.375  0.375  0.375  0.375  0.375
 0.4375 0.4375 0.5625 0.5625 0.625  0.625  0.625  0.6875 0.6875 0.6875
 0.75   0.75   0.75   0.75   0.75   0.8125 0.8125 0.8125 0.8125 0.8125
 0.8125 0.875  0.875  0.9375 0.9375 1.     1.     1.     1.     1.
 1.    ]
ROC AUC: 0.649

Aggregated Test Set Metrics Across Seeds:
   accuracy  sensitivity  specificity        f1   roc_auc
0  0.689189       0.4375     0.719697  0.233333  0.647491
1  0.682432       0.4375     0.712121  0.229508  0.645833
2  0.689189       0.4375     0.719697  0.233333  0.650805
3  0.689189       0.4375     0.719697  0.233333  0.648674
4  0.689189       0.4375     0.719697  0.233333  0.647491
5  0.689189       0.4375     0.719697  0.233333  0.649148
6  0.682432       0.4375     0.712121  0.229508  0.647964
7  0.689189       0.4375     0.719697  0.233333  0.646070
8  0.689189       0.4375     0.719697  0.233333  0.649858
9  0.689189       0.4375     0.719697  0.233333  0.648911

Summary of Test Set Metrics (Mean, Standard Error, 95% Confidence Interval)
Accuracy: Mean = 0.688, SE = 0.001, 95% CI = [0.686, 0.690]
Sensitivity: Mean = 0.438, SE = 0.000, 95% CI = [0.438, 0.438]
Specificity: Mean = 0.718, SE = 0.001, 95% CI = [0.716, 0.720]
F1: Mean = 0.233, SE = 0.001, 95% CI = [0.231, 0.234]
Roc_auc: Mean = 0.648, SE = 0.000, 95% CI = [0.647, 0.649]
```

```
def evaluate_model(model, name, grid, X_train, y_train, X_test, y_test, cv, scor
    print(f"\nEvaluating {name} with seed {seed}...")

    # Define inner and outer CV splits using the provided seed
    inner_cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=seed)
    outer_cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=seed)

    # Grid search using inner CV
    clf = GridSearchCV(model, grid, cv=inner_cv, scoring='roc_auc')
    nested_scores = cross_validate(clf, X=X_train, y=y_train, cv=outer_cv, scori

    # Fit grid search on full training set and extract the best estimator
    clf.fit(X_train, y_train)
    best_model = clf.best_estimator_
    best_params = clf.best_params_
```

```
print(f"Best parameters for {name}: {best_params}")

# Calibrate the best model
calibrated_clf = CalibratedClassifierCV(estimator=best_model, method='sigmoi
calibrated_clf.fit(X_train, y_train)

# Get predicted probabilities on the test set from the calibrated classifier
y_probs = calibrated_clf.predict_proba(X_test)[:, 1]

# --- Calculate Training Metrics ---
y_train_pred = best_model.predict(X_train)
y_train_probs = best_model.predict_proba(X_train)[:, 1]
train_acc    = accuracy_score(y_train, y_train_pred)
train_sens   = sensitivity(y_train, y_train_pred)
train_spec   = specificity(y_train, y_train_pred)
train_f1     = f1_score(y_train, y_train_pred)
train_roc_auc = roc_auc_score(y_train, y_train_probs)

print(f"Training Metrics - Accuracy: {train_acc:.3f}, Sensitivity: {train_se

# --- Calculate Test Metrics for the manually set threshold ---
y_pred_manual = (y_probs >= manual_threshold).astype(int)
manual_acc    = accuracy_score(y_test, y_pred_manual)
manual_sens   = sensitivity(y_test, y_pred_manual)
manual_spec   = specificity(y_test, y_pred_manual)
manual_f1     = f1_score(y_test, y_pred_manual)
manual_roc_auc = roc_auc_score(y_test, y_probs)

print(f"\nTest Metrics for manual threshold {manual_threshold}:")
print(f"Accuracy: {manual_acc:.3f}, Sensitivity: {manual_sens:.3f}, Specific

# --- Evaluate metrics across a range of thresholds ---
threshold_metrics = {}
for threshold in threshold_list:
    y_pred_threshold = (y_probs >= threshold).astype(int)
    threshold_acc    = accuracy_score(y_test, y_pred_threshold)
    threshold_sens   = sensitivity(y_test, y_pred_threshold)
    threshold_spec   = specificity(y_test, y_pred_threshold)
    threshold_f1     = f1_score(y_test, y_pred_threshold)
    threshold_metrics[threshold] = {
        'Accuracy': threshold_acc,
        'Sensitivity': threshold_sens,
        'Specificity': threshold_spec,
        'F1': threshold_f1,
        'ROC AUC': manual_roc_auc  # same ROC AUC regardless of threshold
    }
for threshold, metrics in threshold_metrics.items():
    print(f"Threshold: {threshold: 2f}, Metrics: {metrics}")
```

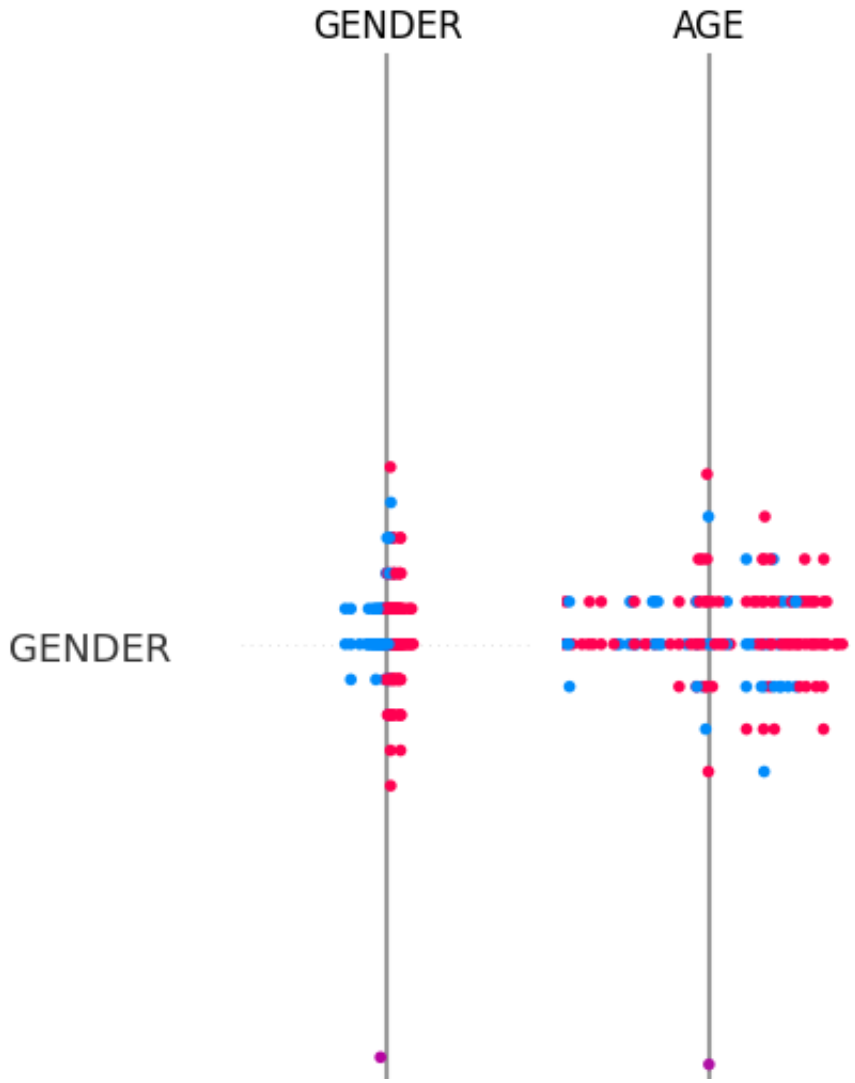
```
        print(f"Threshold: {threshold:.2f}, Metrics: {metrics}")

    # Plot SHAP summary
    calculate_and_plot_shap(best_model, X_train, X_test, name)

    # Prepare dictionary of test metrics at the manual threshold for aggregation
    test_metrics = {
        "accuracy": manual_acc,
        "sensitivity": manual_sens,
        "specificity": manual_spec,
        "f1": manual_f1,
        "roc_auc": manual_roc_auc
    }

    return best_model, manual_threshold, best_params, nested_scores, calibrated_

# --- SHAP Plotting Function ---
def calculate_and_plot_shap(model, X_train, X_test, model_name):
    # Use TreeExplainer if model is a Random Forest; otherwise use KernelExplain
    if isinstance(model, RandomForestClassifier):
        explainer = shap.TreeExplainer(model)
    else:
        explainer = shap.KernelExplainer(model.predict_proba, X_train.sample(100
    shap_values = explainer.shap_values(X_test)
    print(f"SHAP Summary for {model_name}")
    shap.summary_plot(shap_values, X_test, max_display=10)

# --- Plotting Functions ---
def plot_confusion_matrix(y_true, y_pred):
    matrix = confusion_matrix(y_true, y_pred)
    sns.heatmap(matrix, annot=True, fmt='d', cmap='Blues',
                xticklabels=['Predicted Success', 'Predicted Failure'],
                yticklabels=['Actual Success', 'Actual Failure'])
    plt.title('Confusion Matrix Random Forest')
    plt.show()

def plot_roc_curve(y_true, y_probs):
    fpr, tpr, thresholds = roc_curve(y_true, y_probs)
    roc_auc = auc(fpr, tpr)

    plt.figure()
    plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area = {roc_
    plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver Operating Characteristic Random Forest')
    plt.legend(loc="lower right")
```

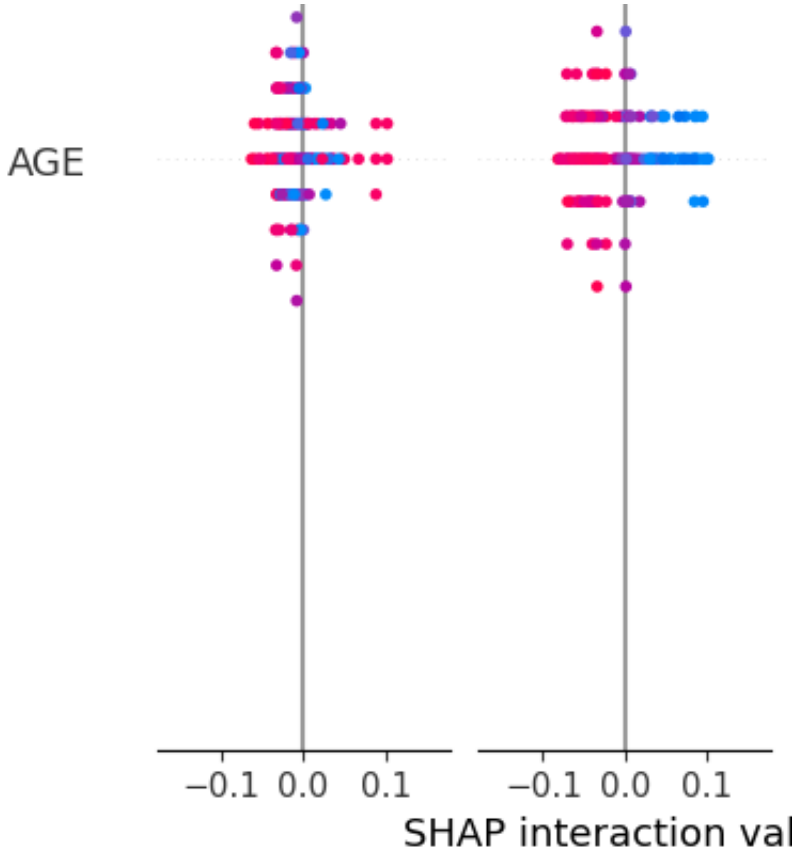
```
    plt.show()

    # --- Added code to output ROC curve metrics ---
    print("ROC Curve Metrics:")
    print("FPR:", fpr)
    print("TPR:", tpr)
    print("ROC AUC: {:.3f}".format(roc_auc))

    return fpr, tpr, roc_auc

# --- Evaluation Function for Random Forest ---
def evaluate_random_forest(X_train_resampled, y_train_resampled, X_test, y_test,
    model = RandomForestClassifier(n_jobs=-1, random_state=seed)
    grid = {
        'n_estimators': [300],
        'max_depth': [7],
        'min_samples_split': [2],
        'min_samples_leaf': [4],
        'max_features': ['sqrt'],
    }
    return evaluate_model(model, "Random Forest", grid, X_train_resampled, y_tra

# --- MAIN FUNCTION: AGGREGATING METRICS ACROSS SEEDS ---
def main(X_train_resampled, y_train_resampled, X_test, y_test):
    cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=10, random_state=42)
    scoring = {
        'accuracy': make_scorer(accuracy_score),
        'sensitivity': make_scorer(sensitivity),
        'specificity': make_scorer(specificity),
        'f1': make_scorer(f1_score),
        'roc_auc': make_scorer(roc_auc_score)
    }
    manual_threshold = 0.3
    threshold_list = np.arange(0.1, 1.05, 0.05)

    # List to collect test metrics from each seed iteration
    aggregated_metrics = []

    for seed in range(40, 50):
        print(f"\nRunning evaluation with seed {seed}")
        (best_model, manual_threshold, best_params, nested_scores,
         calibrated_clf, threshold_metrics, test_metrics) = evaluate_random_fore
            X_train_resampled, y_train_resampled, X_test, y_test, cv, scoring, m
        )

        # Use calibrated_clf for prediction probabilities (for plotting)
        y_probs = calibrated_clf.predict_proba(X_test)[:, 1]
        y_pred_manual = (y_probs >= manual_threshold).astype(int)
```

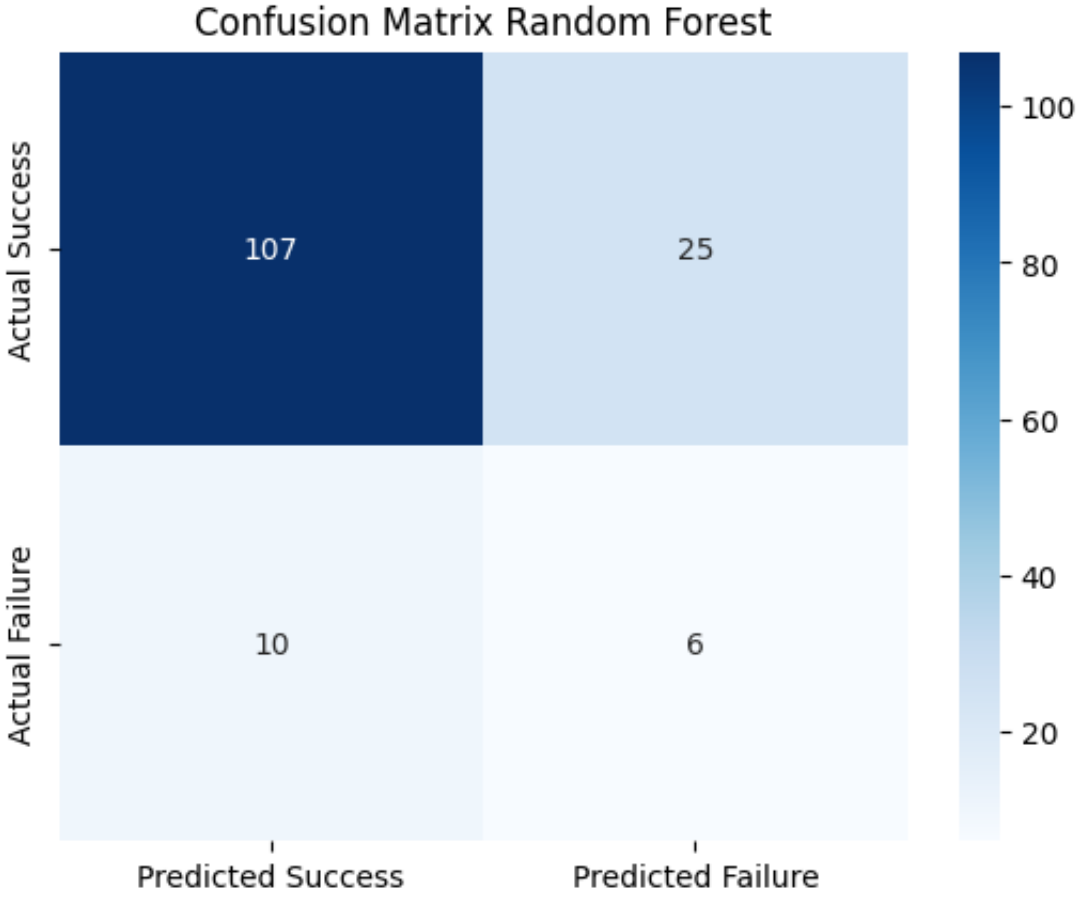
```
        # Plot confusion matrix and ROC curve for this seed
        plot_confusion_matrix(y_test, y_pred_manual)
        plot_roc_curve(y_test, y_probs)

        # Append the test set metrics from this seed for later aggregation
        aggregated_metrics.append(test_metrics)

    # --- Aggregate Results Across Seeds ---
    results_df = pd.DataFrame(aggregated_metrics)
    n = len(results_df)
    print("\nAggregated Test Set Metrics Across Seeds:")
    print(results_df)

    # Function to compute mean, standard error, and 95% confidence interval usin
    def summarize_metric(metric_values):
        mean_val = metric_values.mean()
        std_val = metric_values.std(ddof=1)
        se = std_val / np.sqrt(n)
        t_crit = stats.t.ppf(0.975, df=n-1)  # 95% confidence, two-tailed
        ci_lower = mean_val - t_crit * se
        ci_upper = mean_val + t_crit * se
        return mean_val, se, (ci_lower, ci_upper)

    metrics_summary = {}
    for metric in results_df.columns:
        mean_val, se, ci = summarize_metric(results_df[metric])
        metrics_summary[metric] = {
            "Mean": mean_val,
            "Standard Error": se,
            "95% CI": ci
        }

    print("\nSummary of Test Set Metrics (Mean, Standard Error, 95% Confidence I
    for metric, summary in metrics_summary.items():
        print(f"{metric.capitalize()}: Mean = {summary['Mean']:.3f}, SE = {summa
              f"95% CI = [{summary['95% CI'][0]:.3f}, {summary['95% CI'][1]:.3f}

# --- RUN THE MAIN FUNCTION ---
# It is assumed that X_train_resampled, y_train_resampled, X_test, and y_test ha
if __name__ == '__main__':
    main(X_train_resampled, y_train_resampled, X_test, y_test)
```

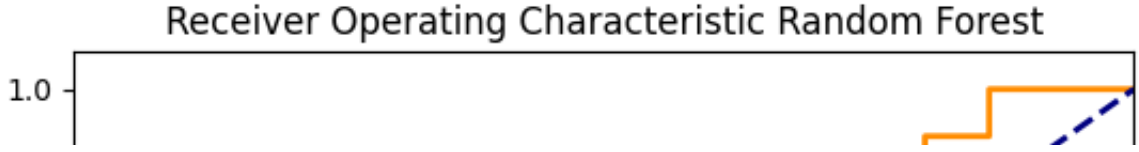
```
Running evaluation with seed 40

Evaluating Random Forest with seed 40...
Best parameters for Random Forest: {'max_depth': 7, 'max_features': 'sqrt',
```

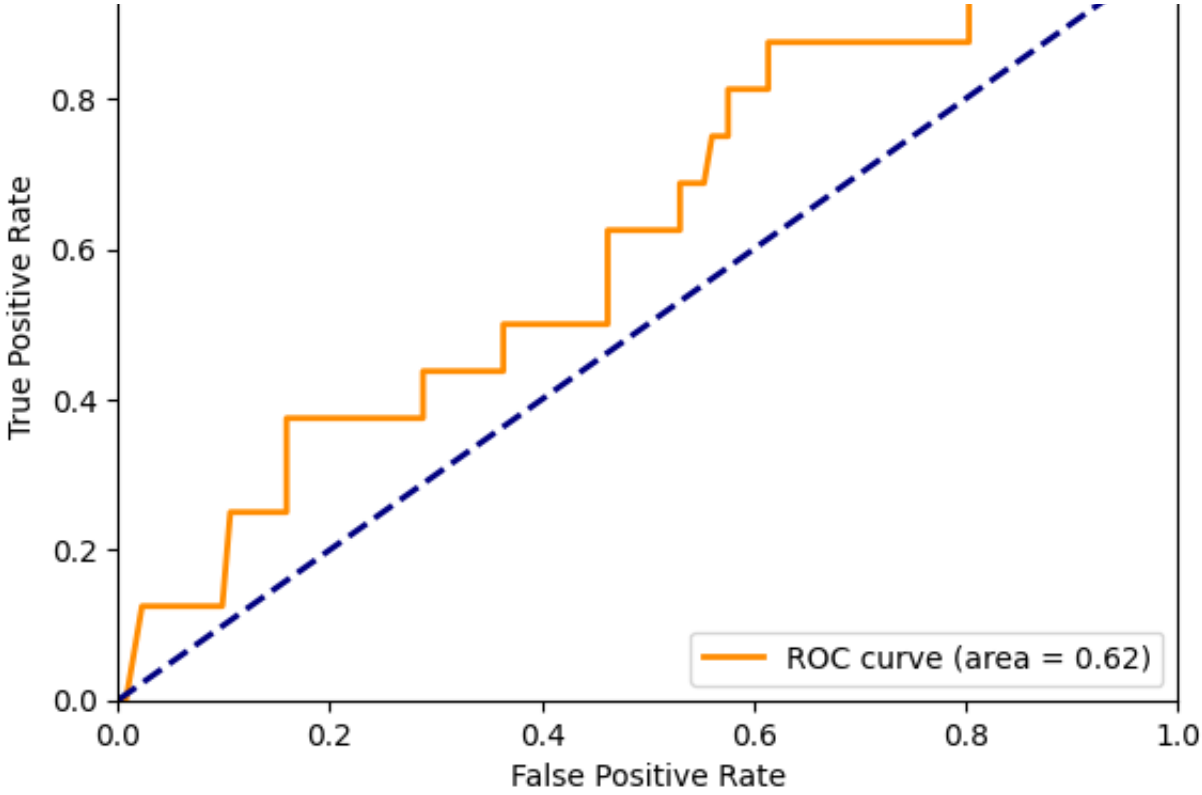
```
Training Metrics - Accuracy: 0.922, Sensitivity: 0.909, Specificity: 0.935,

Test Metrics for manual threshold 0.3:
Accuracy: 0.764, Sensitivity: 0.375, Specificity: 0.811, F1: 0.255, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.47297297297297297, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.6418918918918919, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.6959459459459459, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.777027027027027, 'Sensitivity': 0.
Threshold: 0.40, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Random Forest
```




AGE
−0.1 0.0 0.1
−0.1 0.0 0.1
SHAP interaction val

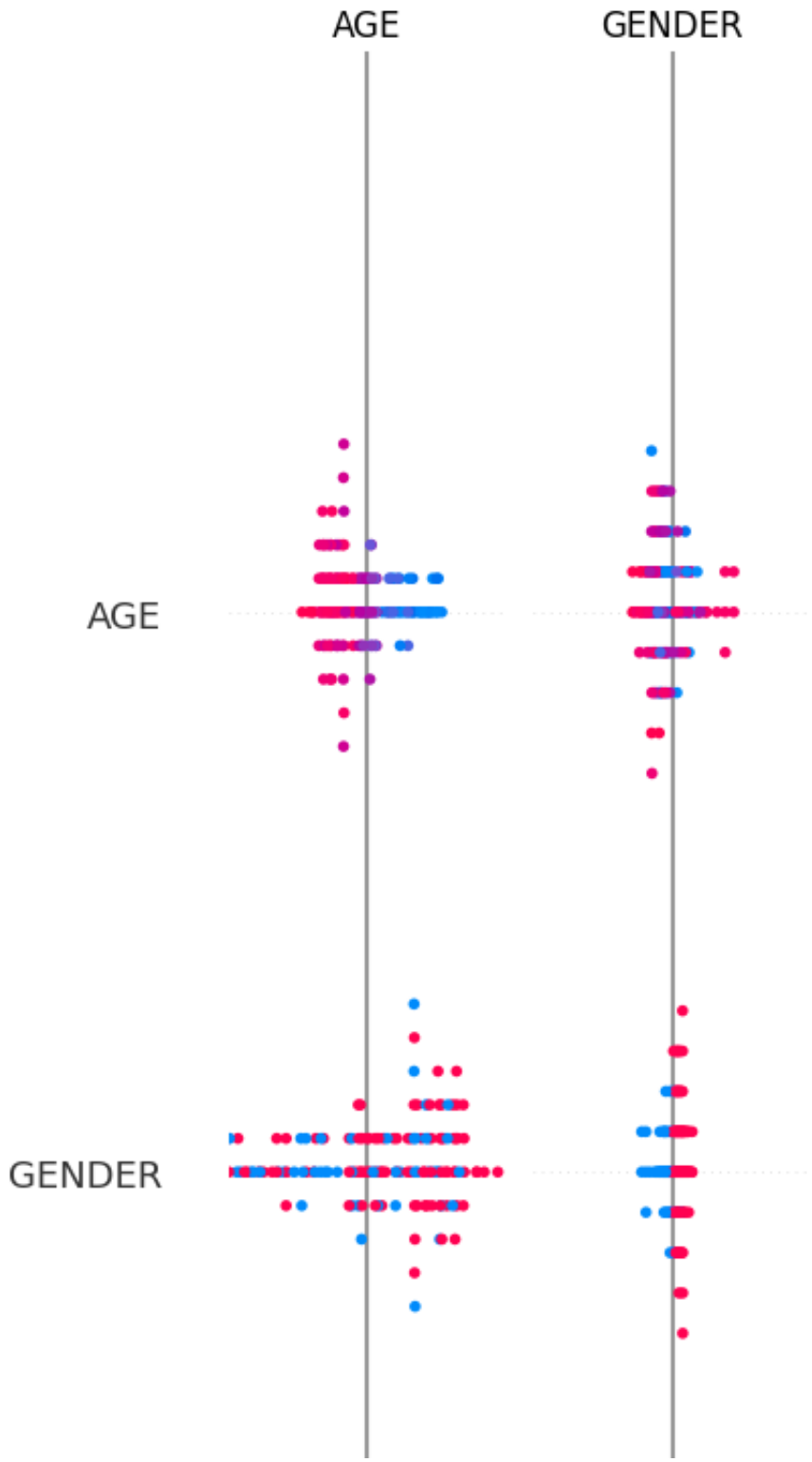

Confusion Matrix Random Forest
Actual Success
Actual Failure
107
25
10
6
Predicted Success
Predicted Failure
100
80
60
40
20

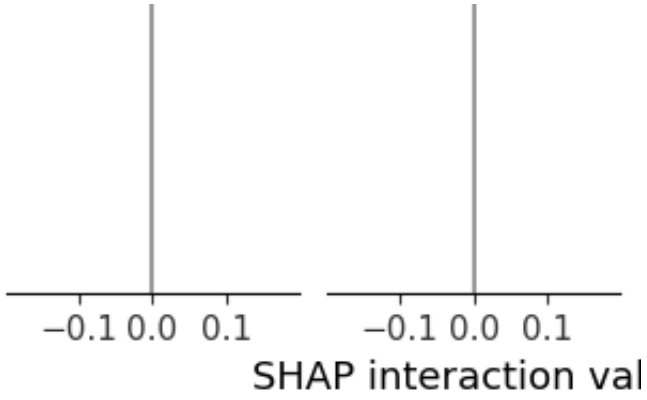

Receiver Operating Characteristic Random Forest
1.0

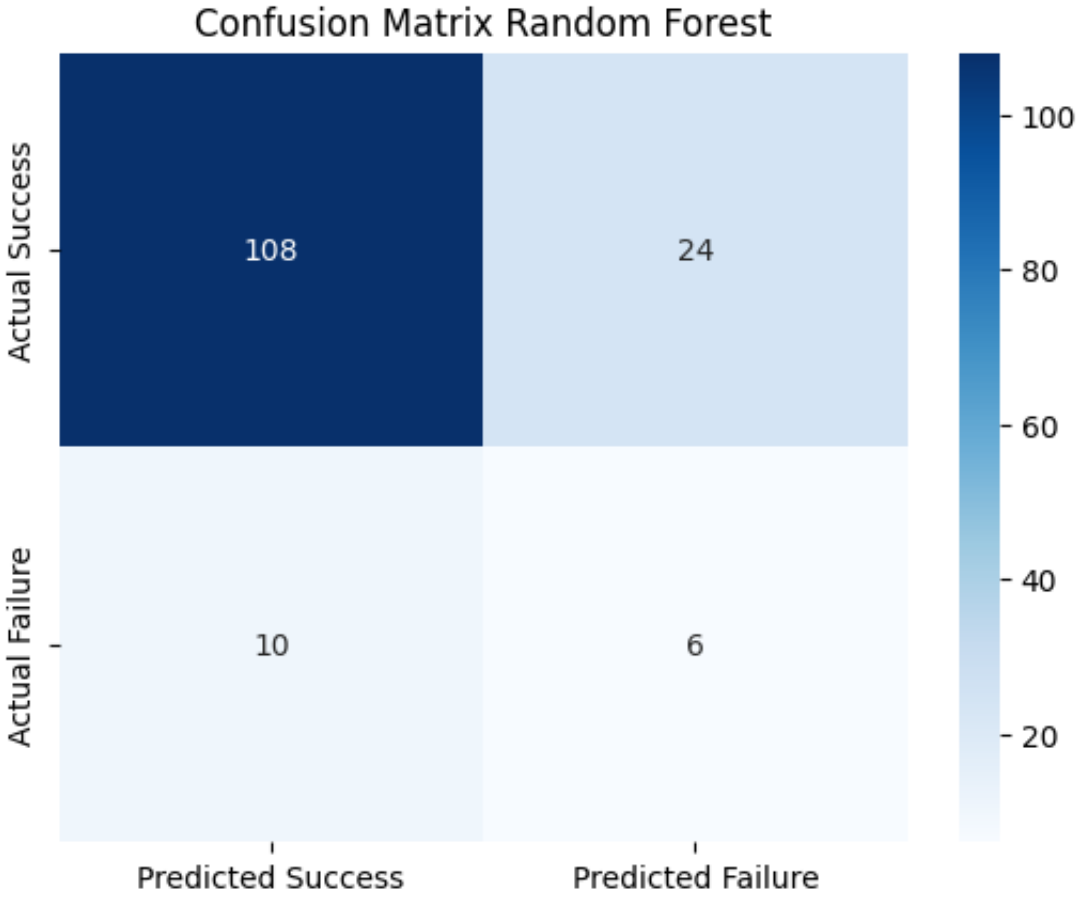


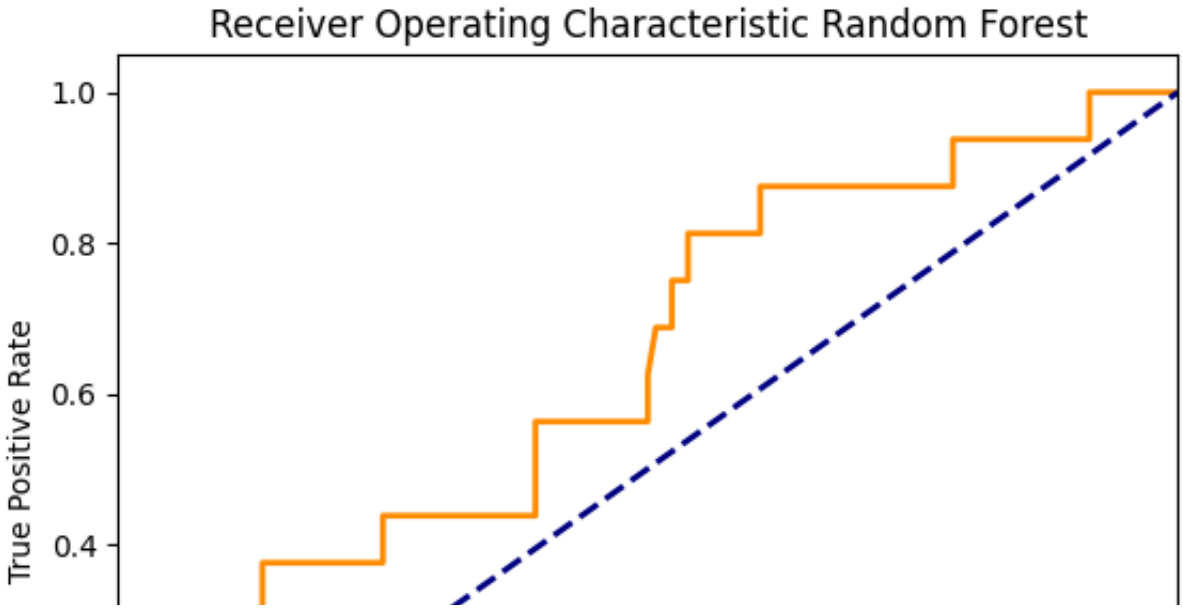

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.09848485 0.10606061 0.14393939
 0.15909091 0.15909091 0.1969697  0.21969697 0.27272727 0.28787879
 0.28787879 0.36363636 0.36363636 0.38636364 0.40909091 0.42424242
 0.46212121 0.46212121 0.47727273 0.48484848 0.5        0.53030303
 0.53030303 0.5530303  0.56060606 0.57575758 0.57575758 0.58333333
 0.59848485 0.61363636 0.61363636 0.75757576 0.77272727 0.8030303
 0.8030303  0.81818182 0.83333333 0.86363636 0.86363636 1.        ]
TPR: [0.     0.     0.125  0.125  0.25   0.25   0.25   0.375  0.375  0.375
 0.375  0.375  0.4375 0.4375 0.5    0.5    0.5    0.5    0.5    0.625
 0.625  0.625  0.625  0.625  0.6875 0.6875 0.75   0.75   0.8125 0.8125
 0.8125 0.8125 0.875  0.875  0.875  0.875  0.9375 0.9375 0.9375 0.9375
 1.     1.    ]
ROC AUC: 0.621

Running evaluation with seed 41

Evaluating Random Forest with seed 41...
Best parameters for Random Forest: {'max_depth': 7, 'max_features': 'sqrt',
Training Metrics - Accuracy: 0.915, Sensitivity: 0.896, Specificity: 0.935,

Test Metrics for manual threshold 0.3:
Accuracy: 0.770, Sensitivity: 0.375, Specificity: 0.818, F1: 0.261, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.49324324324324326, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.6621621621621622, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.7094594594594594, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.8040540540540541, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
```

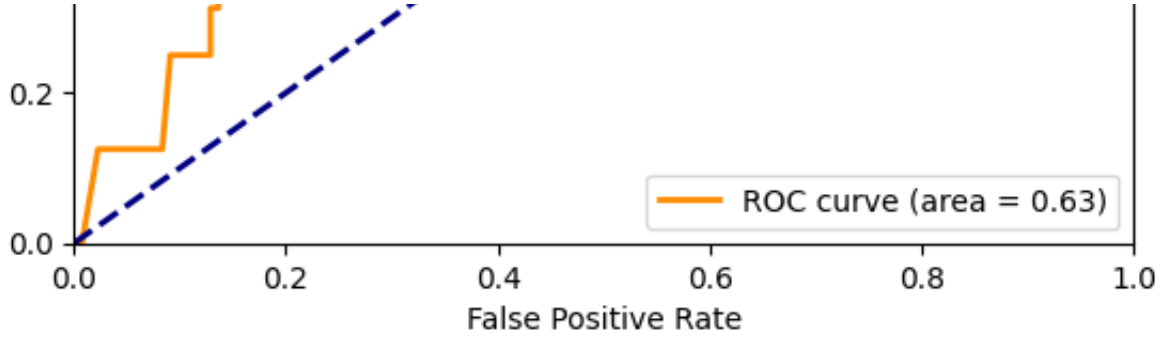
```
Threshold: 0.55, Metrics: {'Accuracy': 0.8513513513513513, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Random Forest
```




−0.1 0.0 0.1
−0.1 0.0 0.1
SHAP interaction val

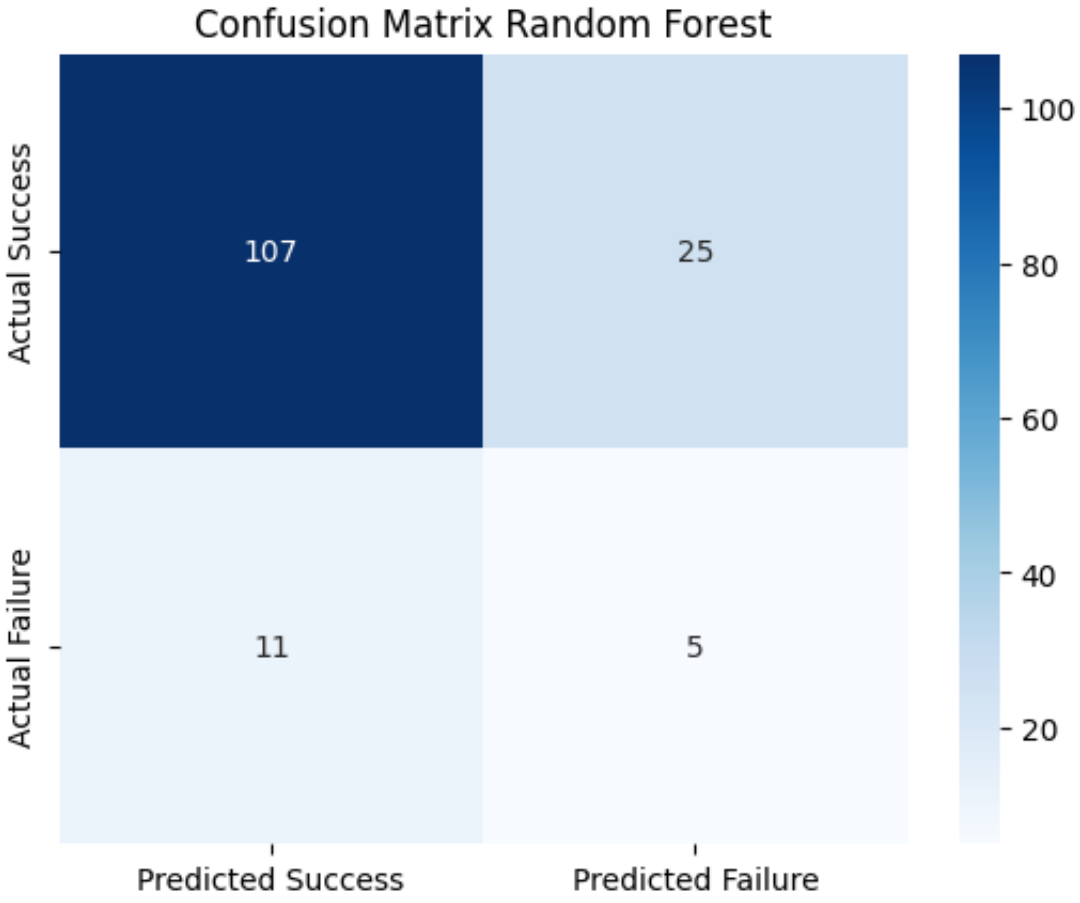

Confusion Matrix Random Forest
Actual Success
Actual Failure
108
24
10
6
100
80
60
40
20
Predicted Success
Predicted Failure

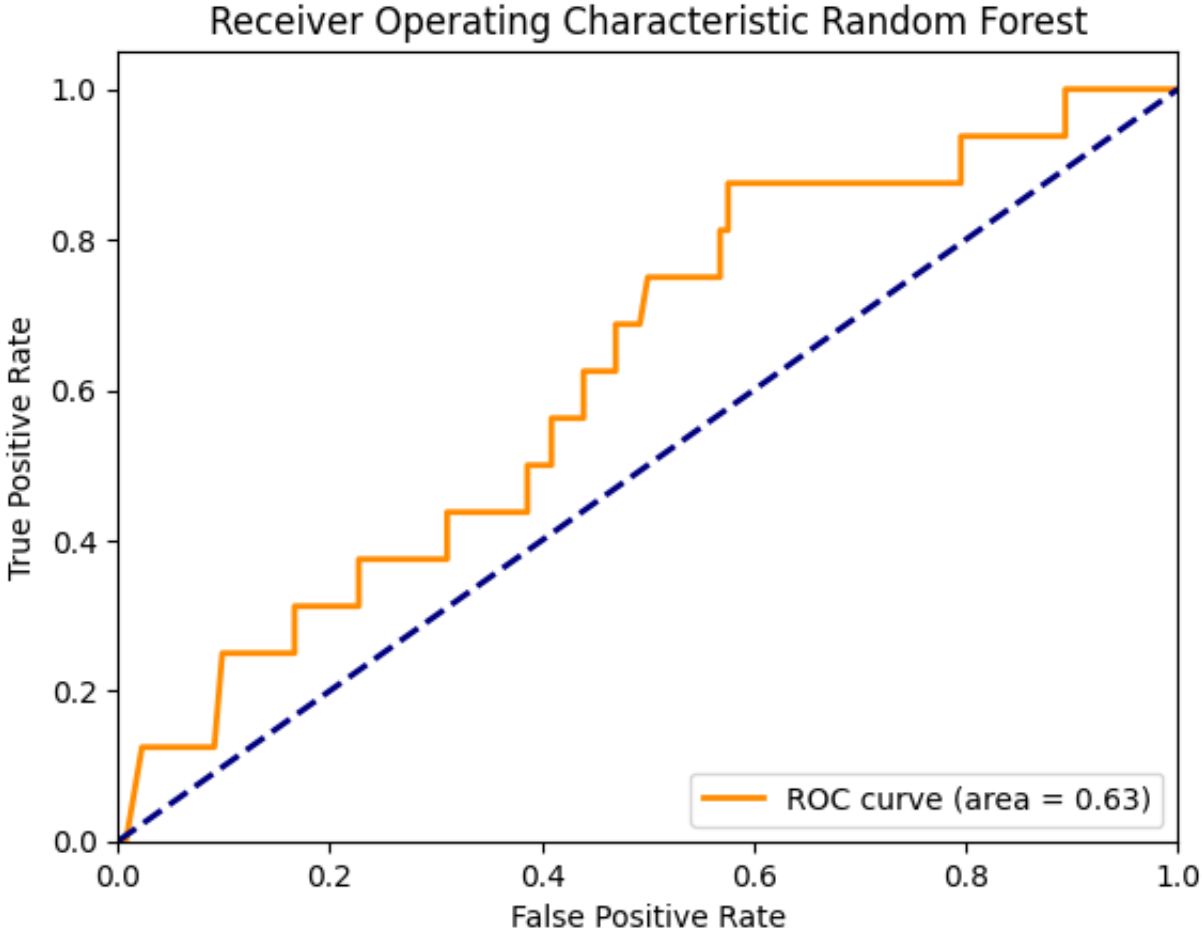

Receiver Operating Characteristic Random Forest
True Positive Rate
1.0
0.8
0.6
0.4




```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.08333333 0.09090909 0.12878788
 0.12878788 0.13636364 0.13636364 0.14393939 0.15909091 0.25
 0.25       0.26515152 0.28787879 0.31060606 0.33333333 0.34848485
 0.36363636 0.37878788 0.39393939 0.39393939 0.41666667 0.43181818
 0.43939394 0.45454545 0.5        0.5        0.50757576 0.52272727
 0.52272727 0.53787879 0.53787879 0.54545455 0.56060606 0.60606061
 0.60606061 0.74242424 0.75757576 0.78787879 0.78787879 0.84848485
 0.86363636 0.91666667 0.91666667 1.        ]
TPR: [0.     0.     0.125  0.125  0.25   0.25   0.3125 0.3125 0.375  0.375
 0.375  0.375  0.4375 0.4375 0.4375 0.4375 0.4375 0.4375 0.4375 0.4375
 0.4375 0.5625 0.5625 0.5625 0.5625 0.5625 0.5625 0.625  0.6875 0.6875
 0.75   0.75   0.8125 0.8125 0.8125 0.8125 0.875  0.875  0.875  0.875
 0.9375 0.9375 0.9375 0.9375 1.     1.    ]
ROC AUC: 0.632

Running evaluation with seed 42

Evaluating Random Forest with seed 42...
Best parameters for Random Forest: {'max_depth': 7, 'max_features': 'sqrt',
Training Metrics - Accuracy: 0.909, Sensitivity: 0.886, Specificity: 0.932,

Test Metrics for manual threshold 0.3:
Accuracy: 0.757, Sensitivity: 0.312, Specificity: 0.811, F1: 0.217, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.49324324324324326, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.6621621621621622, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.7027027027027027, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.7905405405405406, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Random Forest
```

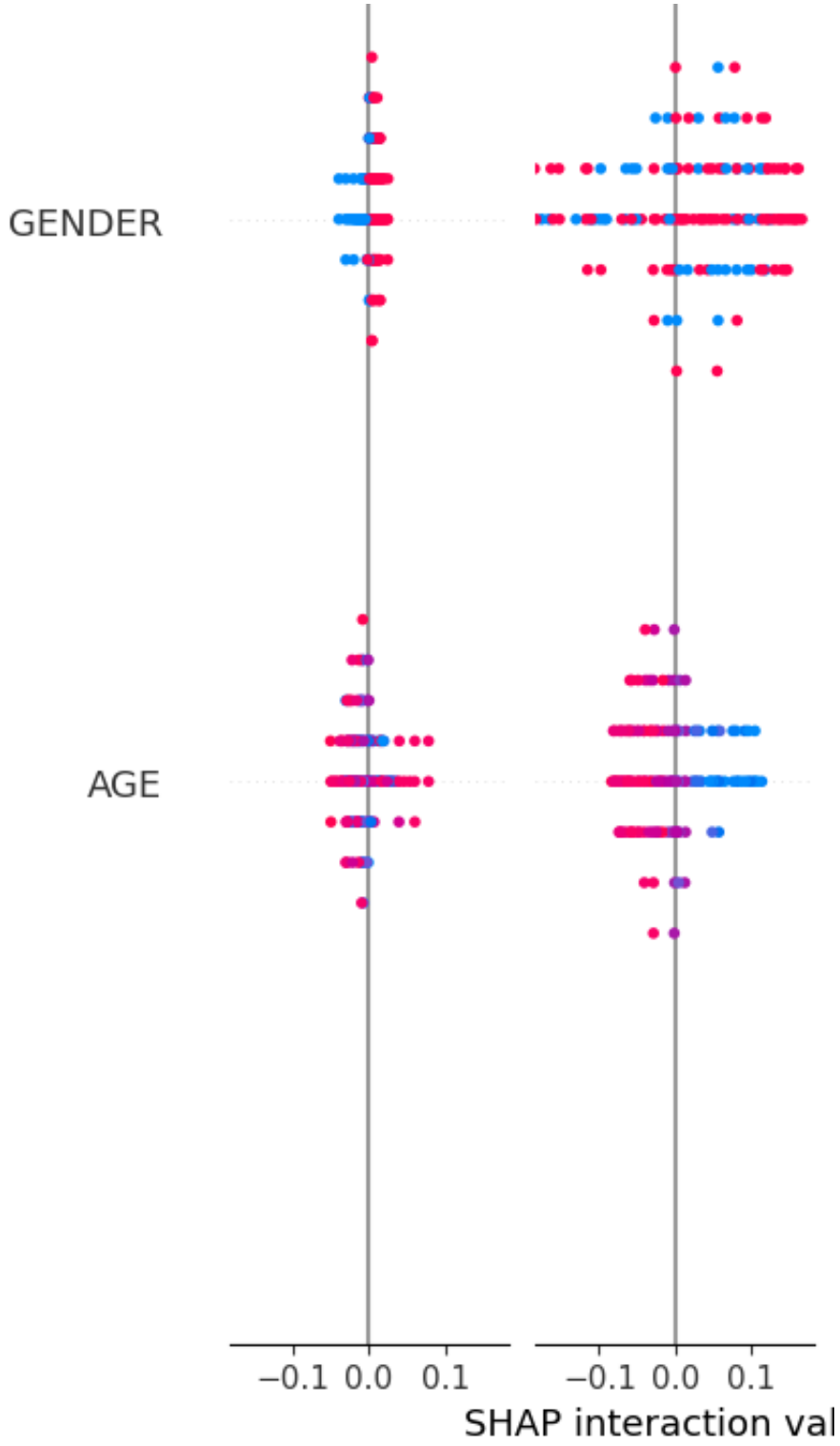

AGE
GENDER
AGE
GENDER
−0.1
0.0
0.1
−0.1
0.0
0.1
SHAP interaction val

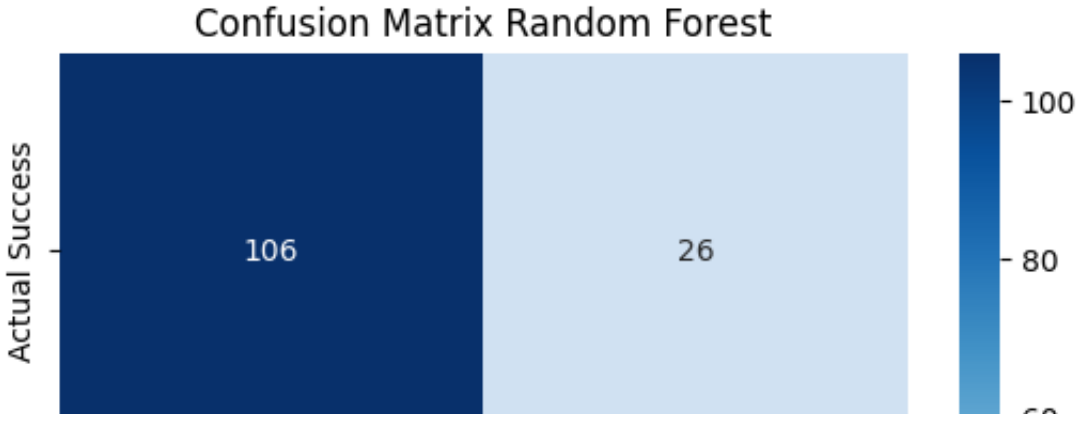


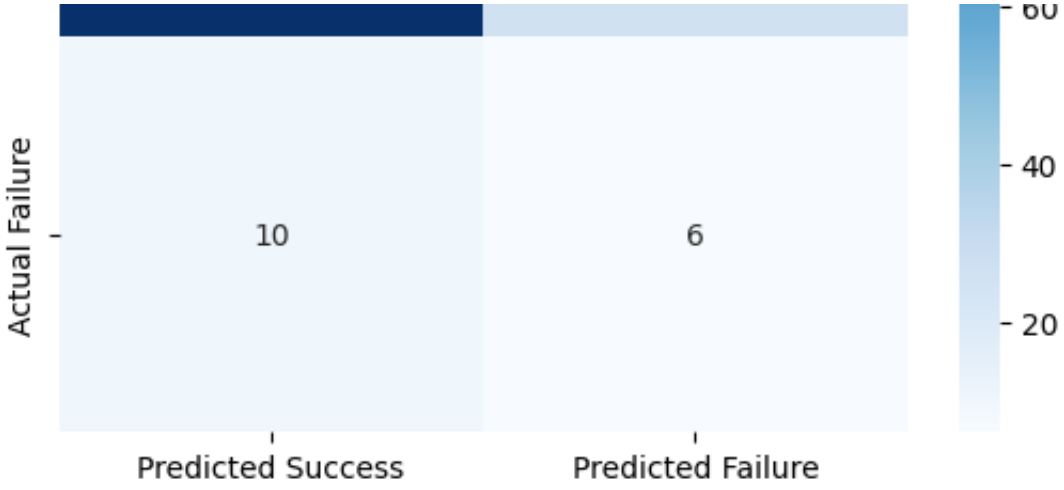


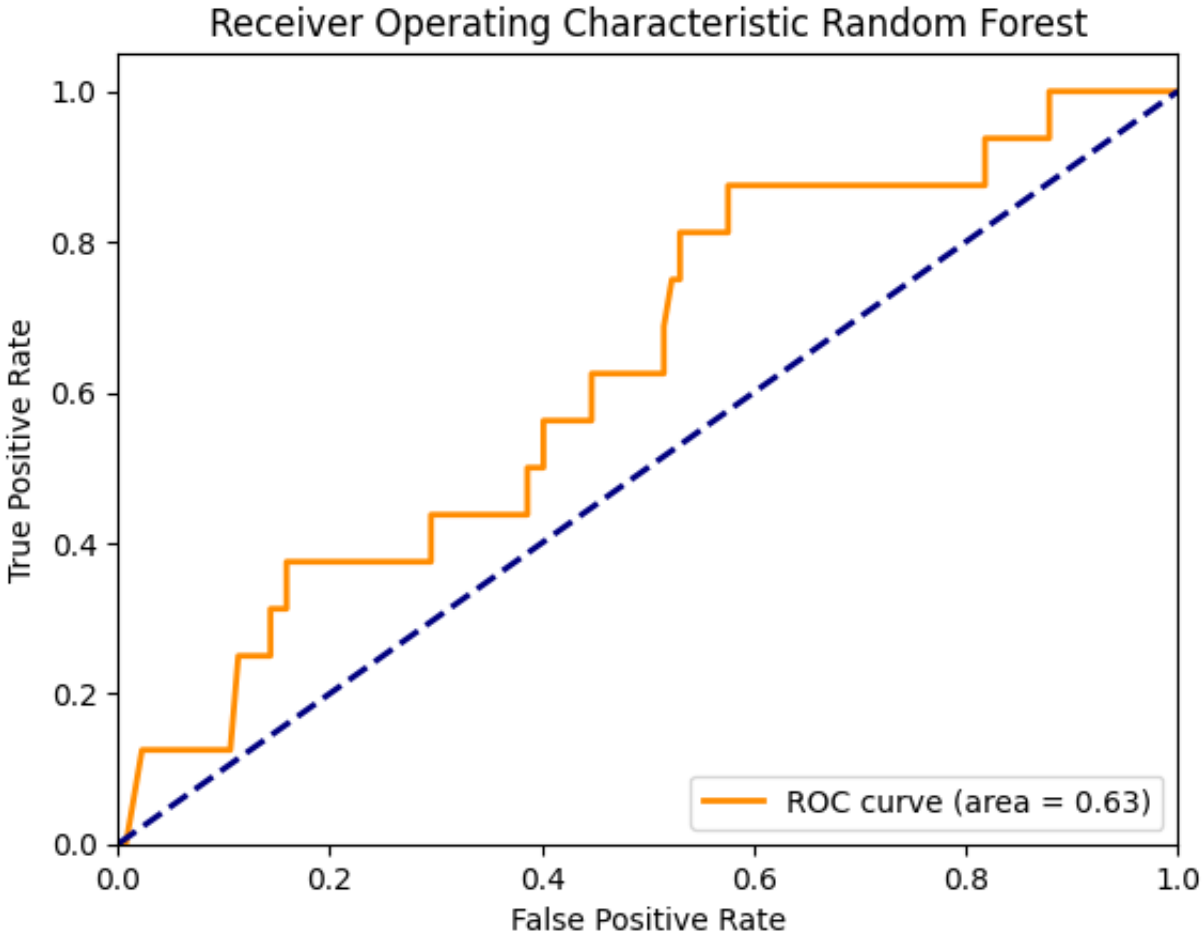

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.09090909 0.09848485 0.16666667
```

```
 0.16666667 0.18939394 0.20454545 0.22727273 0.22727273 0.26515152
 0.28787879 0.3030303  0.31060606 0.31060606 0.33333333 0.35606061
 0.38636364 0.38636364 0.40909091 0.40909091 0.42424242 0.43939394
 0.43939394 0.45454545 0.46969697 0.46969697 0.49242424 0.5
 0.51515152 0.56818182 0.56818182 0.57575758 0.57575758 0.62878788
 0.64393939 0.76515152 0.78030303 0.79545455 0.79545455 0.83333333
 0.84848485 0.89393939 0.89393939 1.         ]
TPR: [0.     0.     0.125  0.125  0.25   0.25   0.3125 0.3125 0.3125 0.3125
 0.375  0.375  0.375  0.375  0.375  0.4375 0.4375 0.4375 0.4375 0.5
 0.5    0.5625 0.5625 0.5625 0.625  0.625  0.625  0.6875 0.6875 0.75
 0.75   0.75   0.8125 0.8125 0.875  0.875  0.875  0.875  0.875  0.875
 0.9375 0.9375 0.9375 0.9375 1.     1.     ]
ROC AUC: 0.628

Running evaluation with seed 43

Evaluating Random Forest with seed 43...
Best parameters for Random Forest: {'max_depth': 7, 'max_features': 'sqrt',
Training Metrics - Accuracy: 0.907, Sensitivity: 0.876, Specificity: 0.938,

Test Metrics for manual threshold 0.3:
Accuracy: 0.757, Sensitivity: 0.375, Specificity: 0.803, F1: 0.250, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.5067567567567568, 'Sensitivity': 0
Threshold: 0.15, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.6554054054054054, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.7094594594594594, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.7905405405405406, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.8108108108108109, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Random Forest
```


GENDER
AGE
−0.1 0.0 0.1
−0.1 0.0 0.1
SHAP interaction val

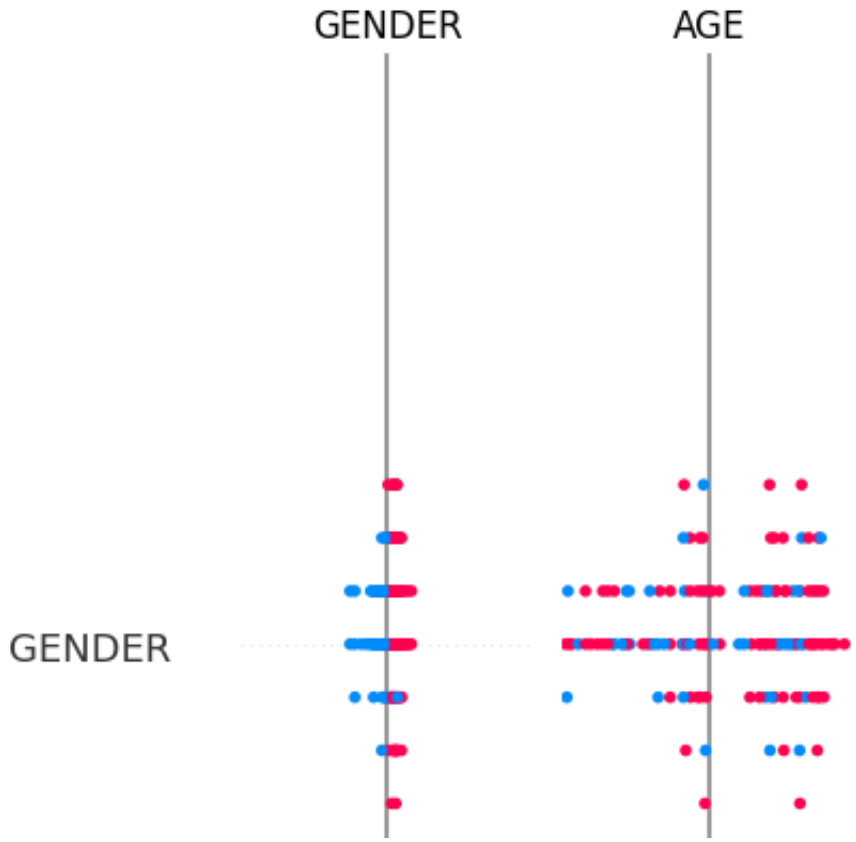

Confusion Matrix Random Forest
Actual Success
106
26
100
80

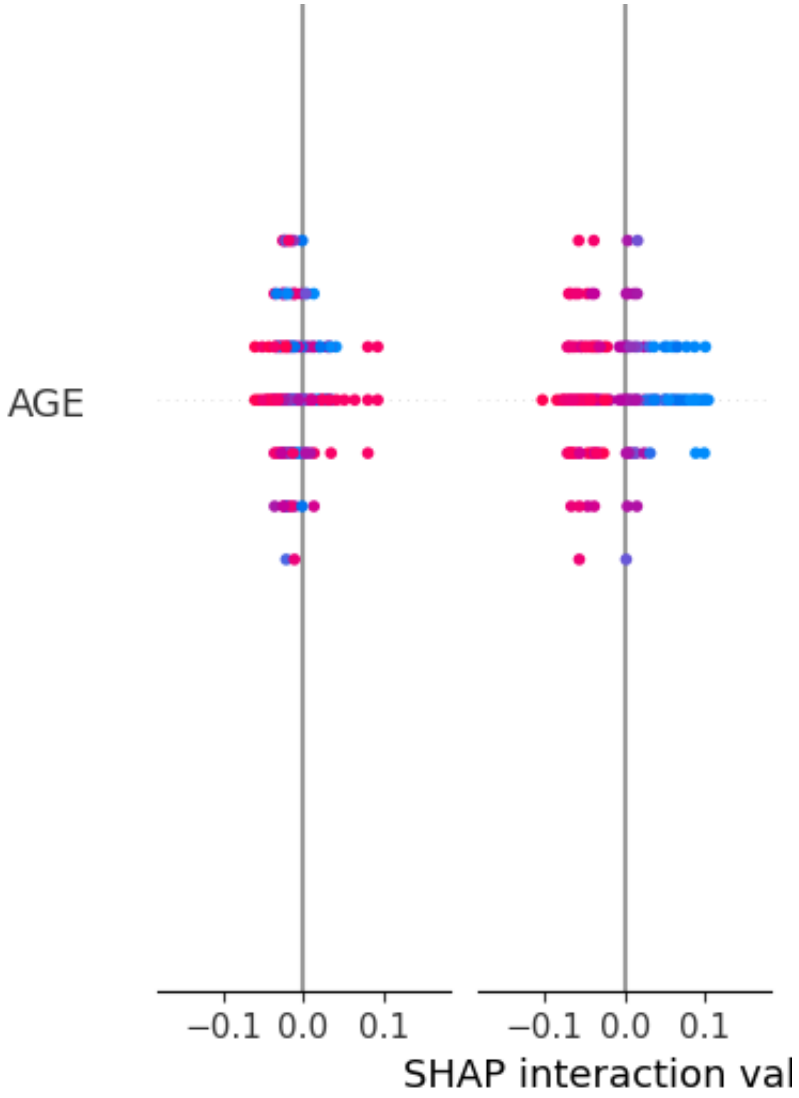


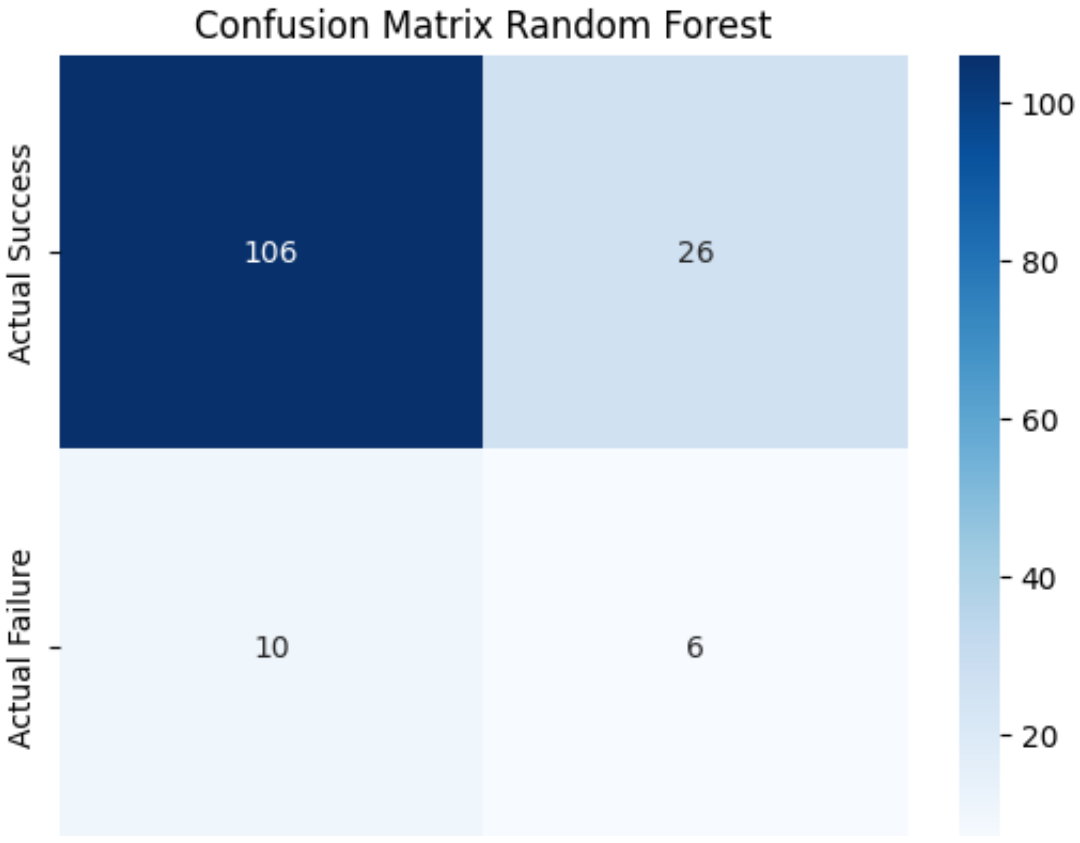


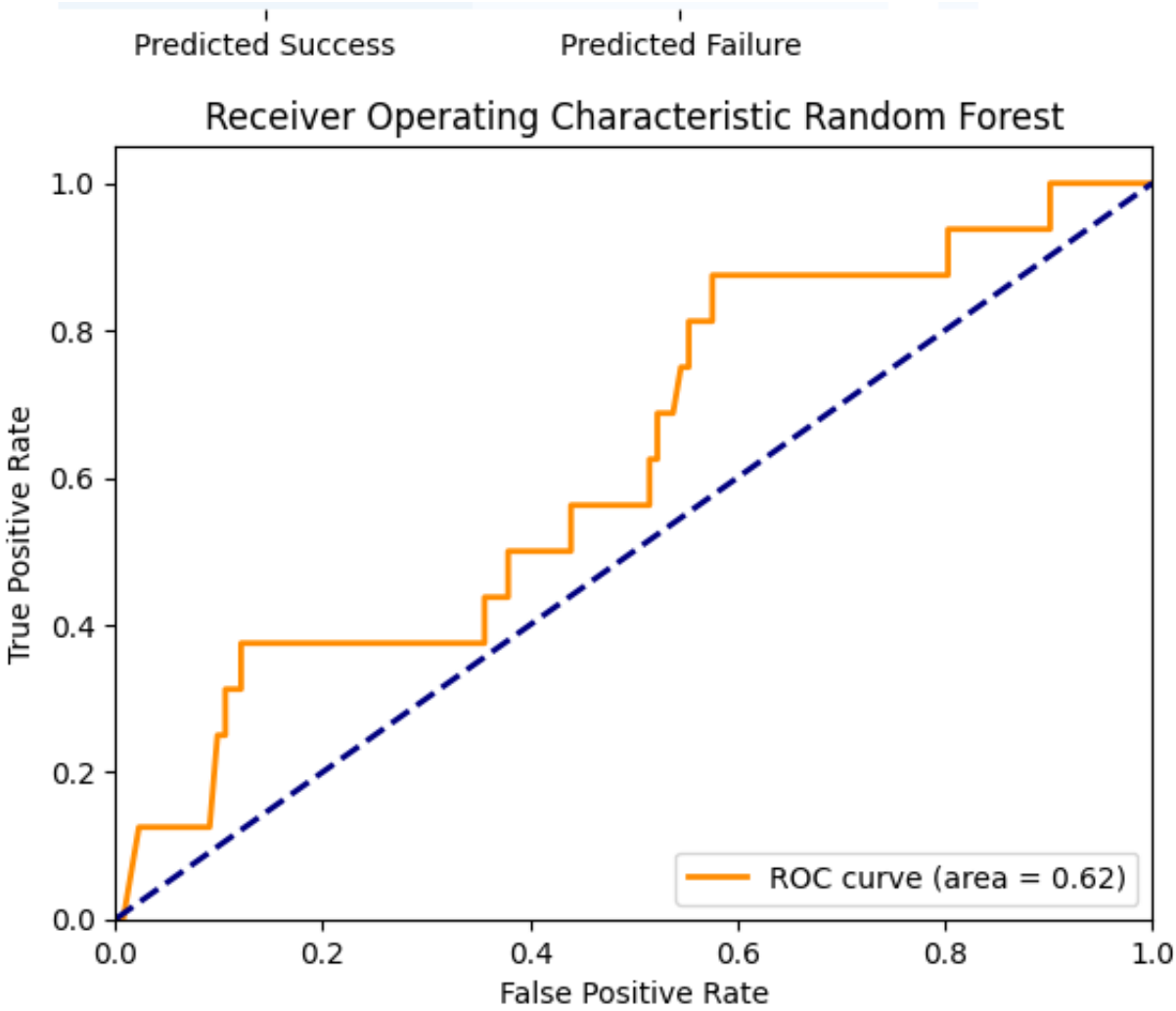

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.10606061 0.11363636 0.14393939
 0.14393939 0.15909091 0.15909091 0.17424242 0.25       0.27272727
 0.29545455 0.29545455 0.31818182 0.33333333 0.35606061 0.37121212
 0.38636364 0.38636364 0.40151515 0.40151515 0.42424242 0.43939394
 0.4469697  0.4469697  0.46212121 0.47727273 0.51515152 0.51515152
 0.52272727 0.53030303 0.53030303 0.57575758 0.57575758 0.61363636
 0.62878788 0.72727273 0.74242424 0.81818182 0.81818182 0.84090909
 0.85606061 0.87878788 0.87878788 1.        ]
TPR: [0.     0.     0.125  0.125  0.25   0.25   0.3125 0.3125 0.375  0.375
 0.375  0.375  0.375  0.4375 0.4375 0.4375 0.4375 0.4375 0.4375 0.5
 0.5    0.5625 0.5625 0.5625 0.5625 0.625  0.625  0.625  0.625  0.6875
 0.75   0.75   0.8125 0.8125 0.875  0.875  0.875  0.875  0.875  0.875
```

```
 0.9375 0.9375 0.9375 0.9375 1.     1.     ]
ROC AUC: 0.630

Running evaluation with seed 44

Evaluating Random Forest with seed 44...
Best parameters for Random Forest: {'max_depth': 7, 'max_features': 'sqrt',
Training Metrics - Accuracy: 0.906, Sensitivity: 0.889, Specificity: 0.922,

Test Metrics for manual threshold 0.3:
Accuracy: 0.757, Sensitivity: 0.375, Specificity: 0.803, F1: 0.250, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.4864864864864865, 'Sensitivity': 0
Threshold: 0.15, Metrics: {'Accuracy': 0.5945945945945946, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.6621621621621622, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.7027027027027027, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Random Forest
```



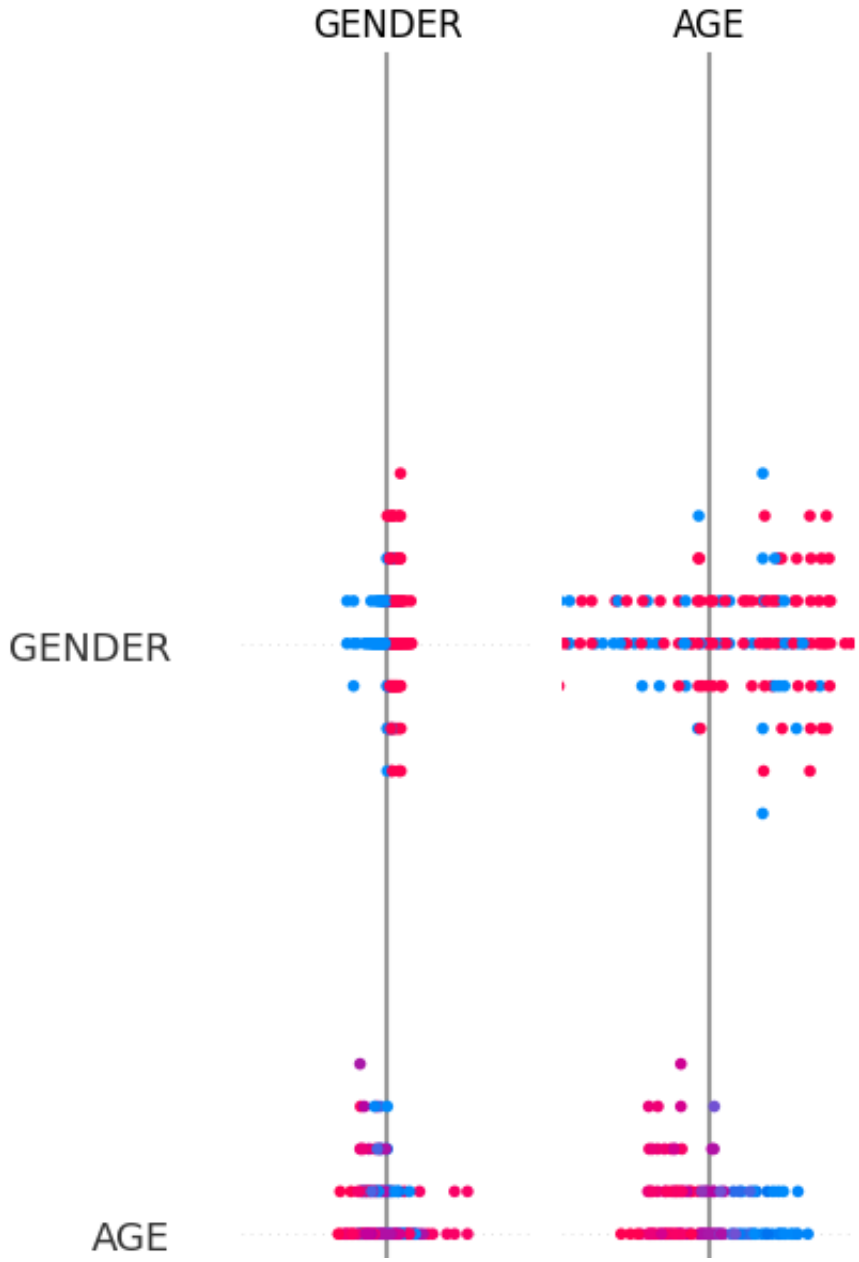

AGE
−0.1 0.0 0.1
−0.1 0.0 0.1
SHAP interaction val

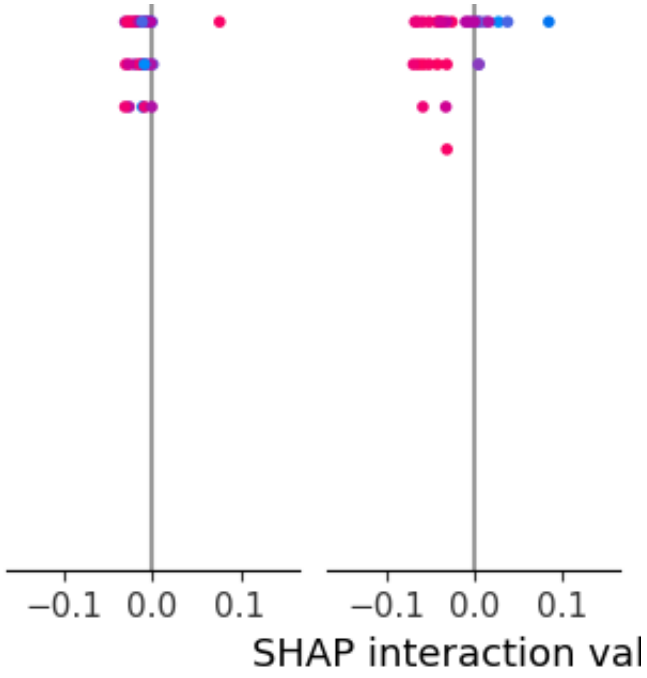

Confusion Matrix Random Forest
Actual Success
Actual Failure
106
26
10
6
100
80
60
40
20

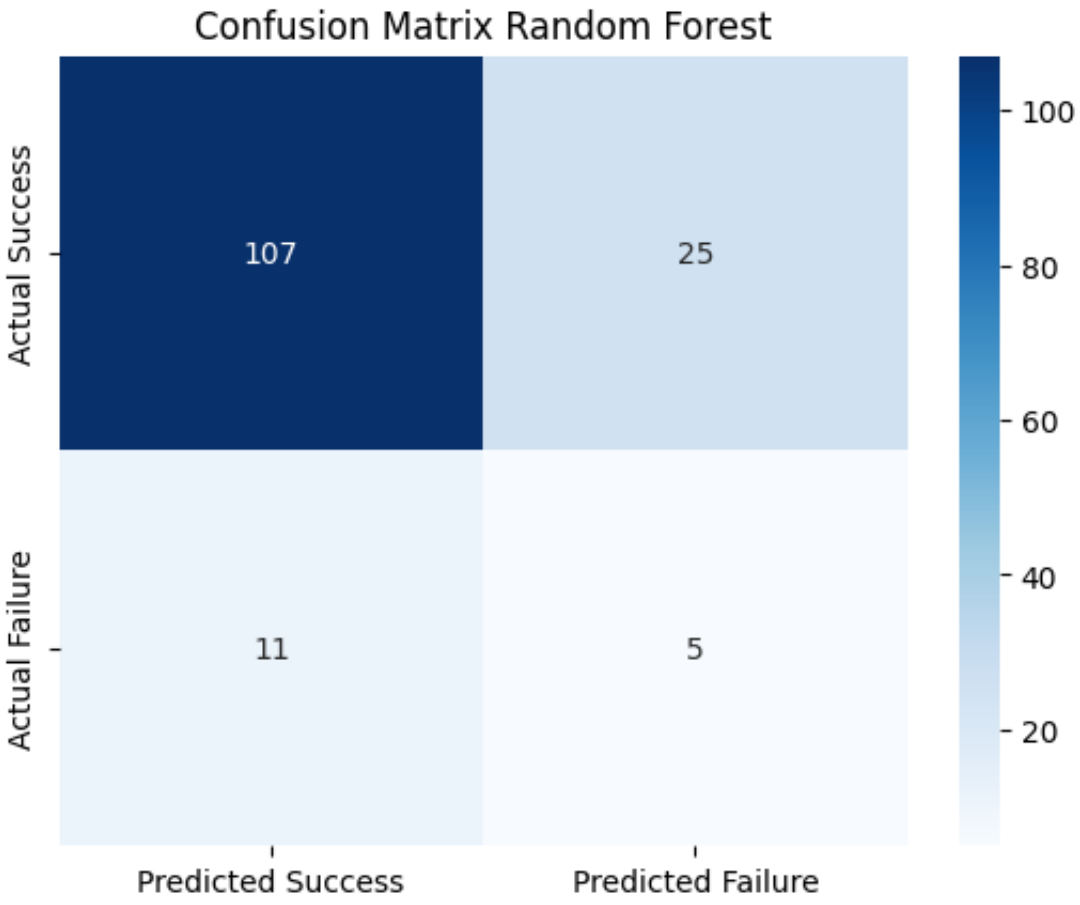


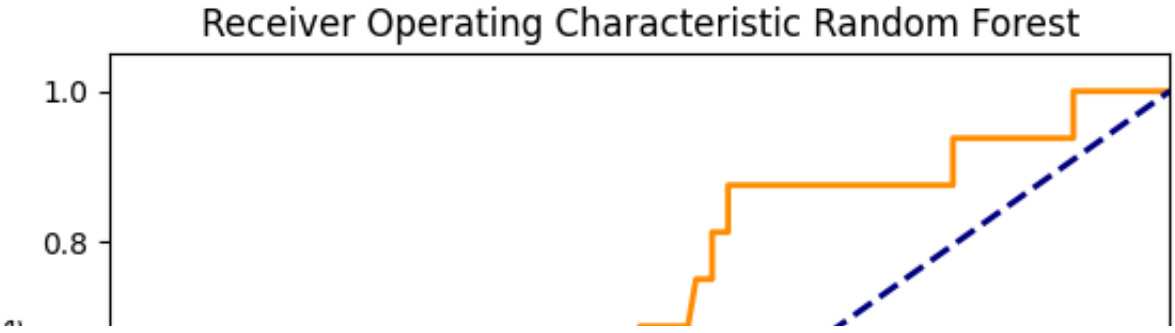

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.09090909 0.09848485 0.10606061
 0.10606061 0.12121212 0.12121212 0.14393939 0.15909091 0.22727273
 0.25       0.3030303  0.31818182 0.33333333 0.35606061 0.35606061
 0.37878788 0.37878788 0.39393939 0.40909091 0.42424242 0.43939394
 0.43939394 0.5        0.51515152 0.51515152 0.52272727 0.52272727
 0.53787879 0.54545455 0.5530303  0.5530303  0.56060606 0.57575758
 0.57575758 0.74242424 0.75757576 0.8030303  0.8030303  0.85606061
 0.87121212 0.90151515 0.90151515 1.        ]
TPR: [0.     0.     0.125  0.125  0.25   0.25   0.3125 0.3125 0.375  0.375
 0.375  0.375  0.375  0.375  0.375  0.375  0.375  0.4375 0.4375 0.5
 0.5    0.5    0.5    0.5    0.5625 0.5625 0.5625 0.625  0.625  0.6875
 0.6875 0.75   0.75   0.8125 0.8125 0.8125 0.875  0.875  0.875  0.875
 0.9375 0.9375 0.9375 0.9375 1.     1.     ]
ROC AUC: 0.623

Running evaluation with seed 45

Evaluating Random Forest with seed 45...
Best parameters for Random Forest: {'max_depth': 7, 'max_features': 'sqrt',
Training Metrics - Accuracy: 0.906, Sensitivity: 0.883, Specificity: 0.928,

Test Metrics for manual threshold 0.3:
Accuracy: 0.757, Sensitivity: 0.312, Specificity: 0.811, F1: 0.217, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.4864864864864865, 'Sensitivity': 0
```

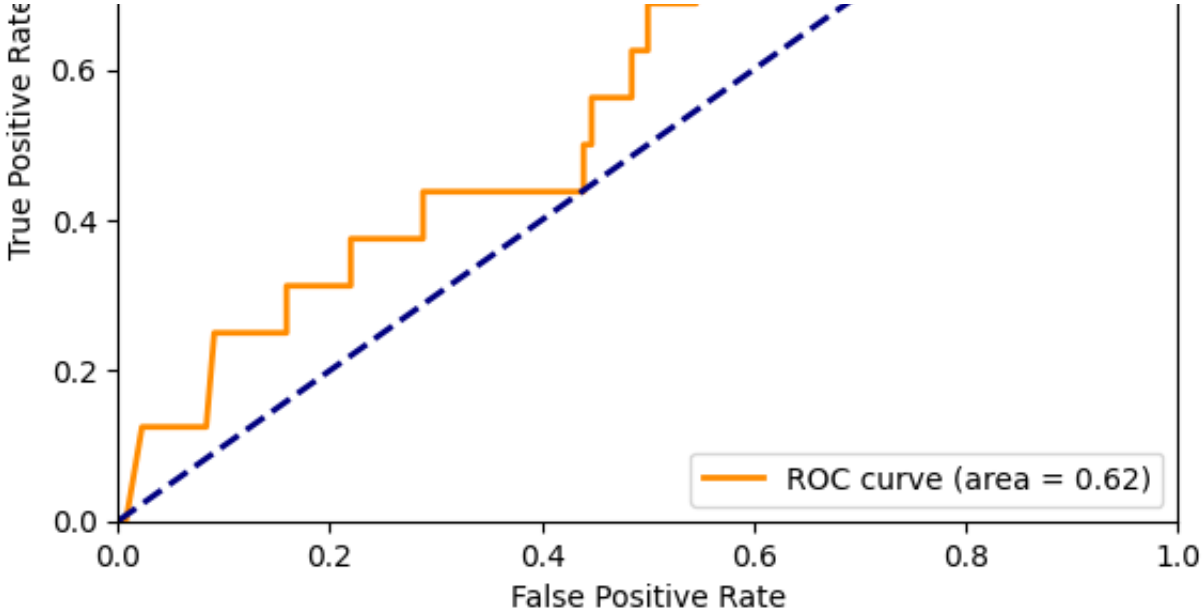
```
Threshold: 0.15, Metrics: {'Accuracy': 0.5743243243243243, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.6824324324324325, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.7162162162162162, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.8040540540540541, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Random Forest
```




−0.1 0.0 0.1
−0.1 0.0 0.1
SHAP interaction val

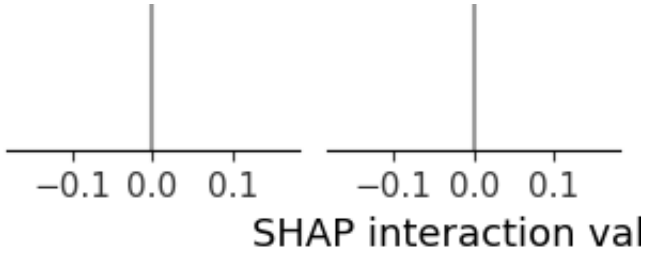

Confusion Matrix Random Forest
Actual Success
Actual Failure
107
25
11
5
Predicted Success
Predicted Failure
100
80
60
40
20

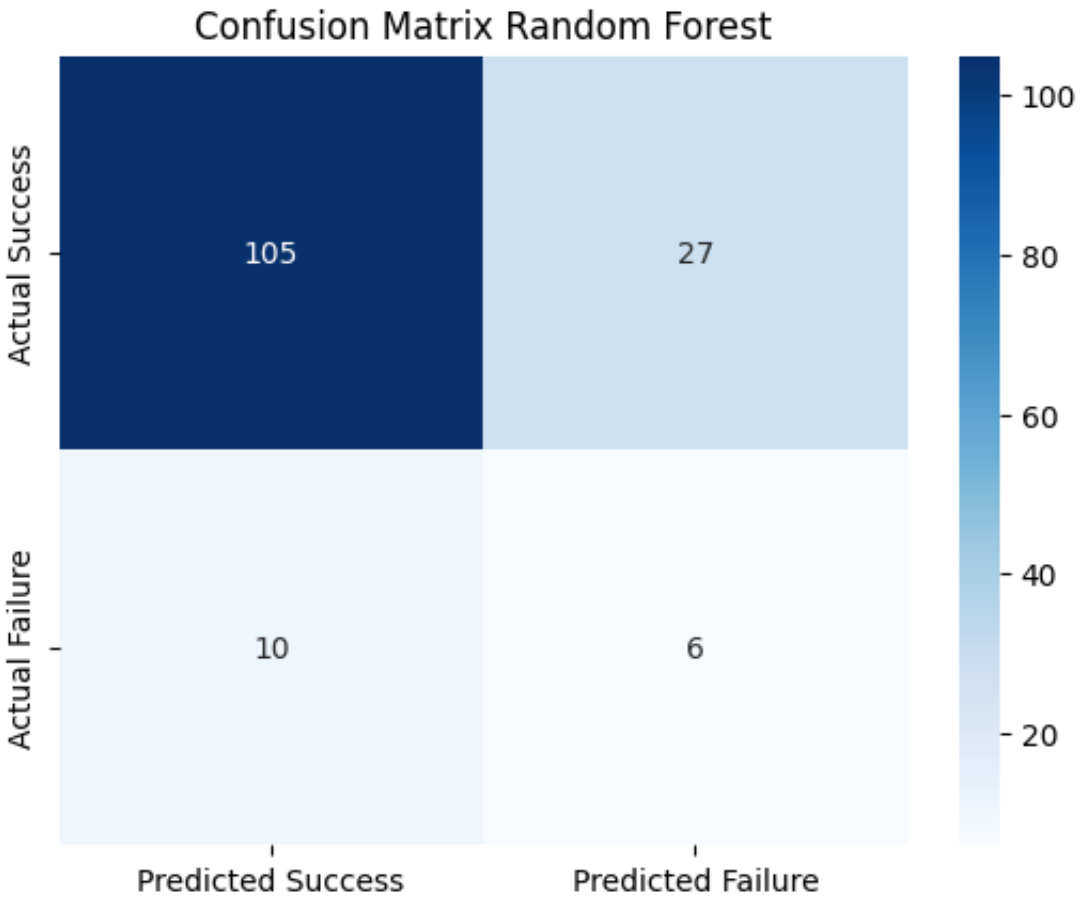

Receiver Operating Characteristic Random Forest
1.0
0.8

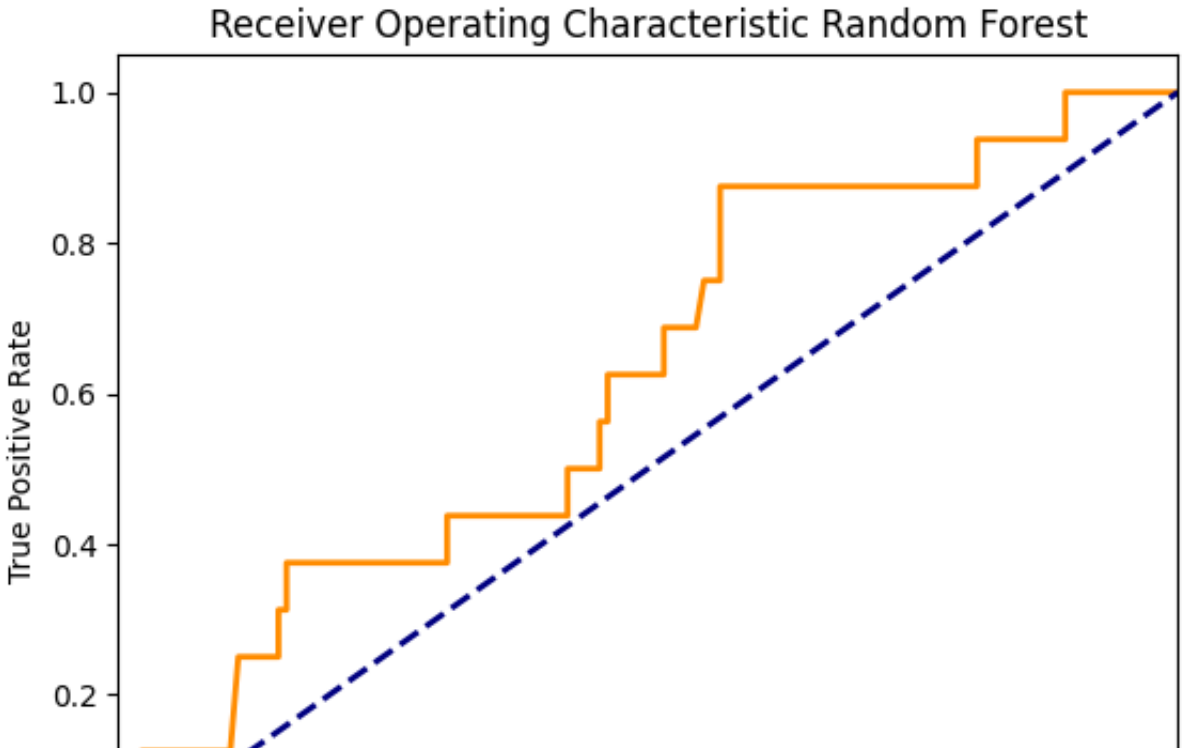


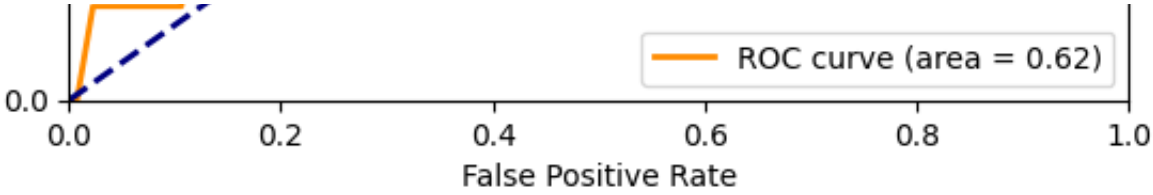

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.08333333 0.09090909 0.14393939
 0.15909091 0.15909091 0.21969697 0.21969697 0.24242424 0.26515152
 0.28787879 0.28787879 0.3030303  0.35606061 0.37878788 0.38636364
 0.40151515 0.43939394 0.43939394 0.4469697  0.4469697  0.46212121
 0.48484848 0.48484848 0.5        0.5        0.52272727 0.53787879
 0.54545455 0.5530303  0.56818182 0.56818182 0.58333333 0.58333333
 0.60606061 0.62121212 0.75       0.76515152 0.79545455 0.79545455
 0.87121212 0.88636364 0.90909091 0.90909091 1.        ]
TPR: [0.     0.     0.125  0.125  0.25   0.25   0.25   0.3125 0.3125 0.375
 0.375  0.375  0.375  0.4375 0.4375 0.4375 0.4375 0.4375 0.4375 0.4375
 0.5    0.5    0.5625 0.5625 0.5625 0.625  0.625  0.6875 0.6875 0.6875
 0.6875 0.75   0.75   0.8125 0.8125 0.875  0.875  0.875  0.875  0.875
 0.875  0.9375 0.9375 0.9375 0.9375 1.     1.     ]
ROC AUC: 0.616

Running evaluation with seed 46

Evaluating Random Forest with seed 46...
Best parameters for Random Forest: {'max_depth': 7, 'max_features': 'sqrt',
Training Metrics - Accuracy: 0.912, Sensitivity: 0.889, Specificity: 0.935,

Test Metrics for manual threshold 0.3:
Accuracy: 0.750, Sensitivity: 0.375, Specificity: 0.795, F1: 0.245, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.49324324324324326, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.5878378378378378, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.6621621621621622, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.722972972972973, 'Sensitivity': 0.
Threshold: 0.30, Metrics: {'Accuracy': 0.75, 'Sensitivity': 0.375, 'Specifi
Threshold: 0.35, Metrics: {'Accuracy': 0.7905405405405406, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
```

```
Threshold: 0.75, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Random Forest
```

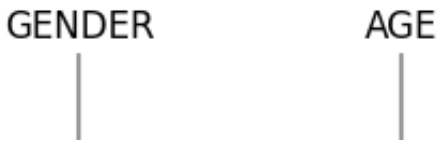

−0.1 0.0 0.1
−0.1 0.0 0.1
SHAP interaction val


Confusion Matrix Random Forest
Actual Success
Actual Failure
105
27
10
6
Predicted Success
Predicted Failure
100
80
60
40
20

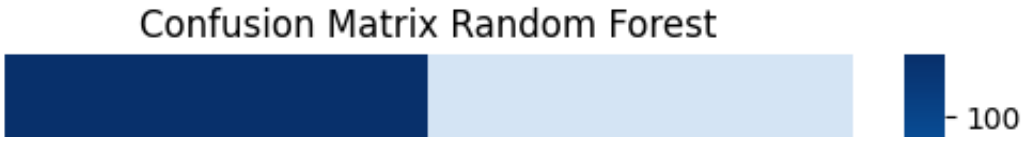

Receiver Operating Characteristic Random Forest
True Positive Rate
1.0
0.8
0.6
0.4
0.2

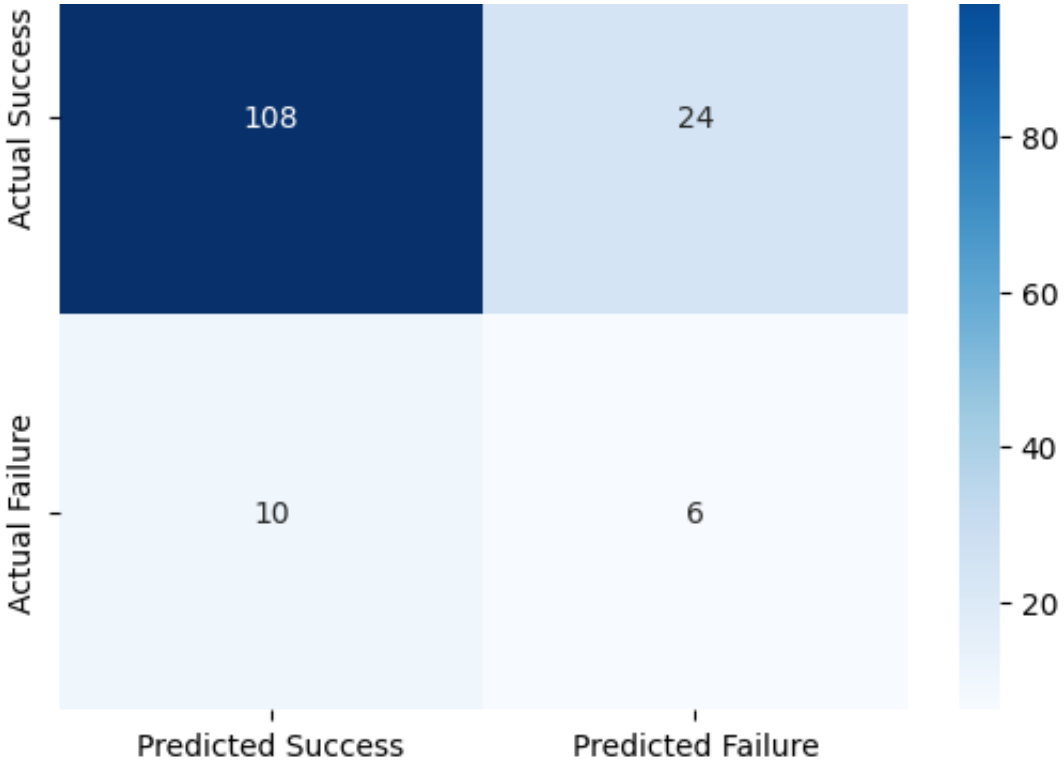


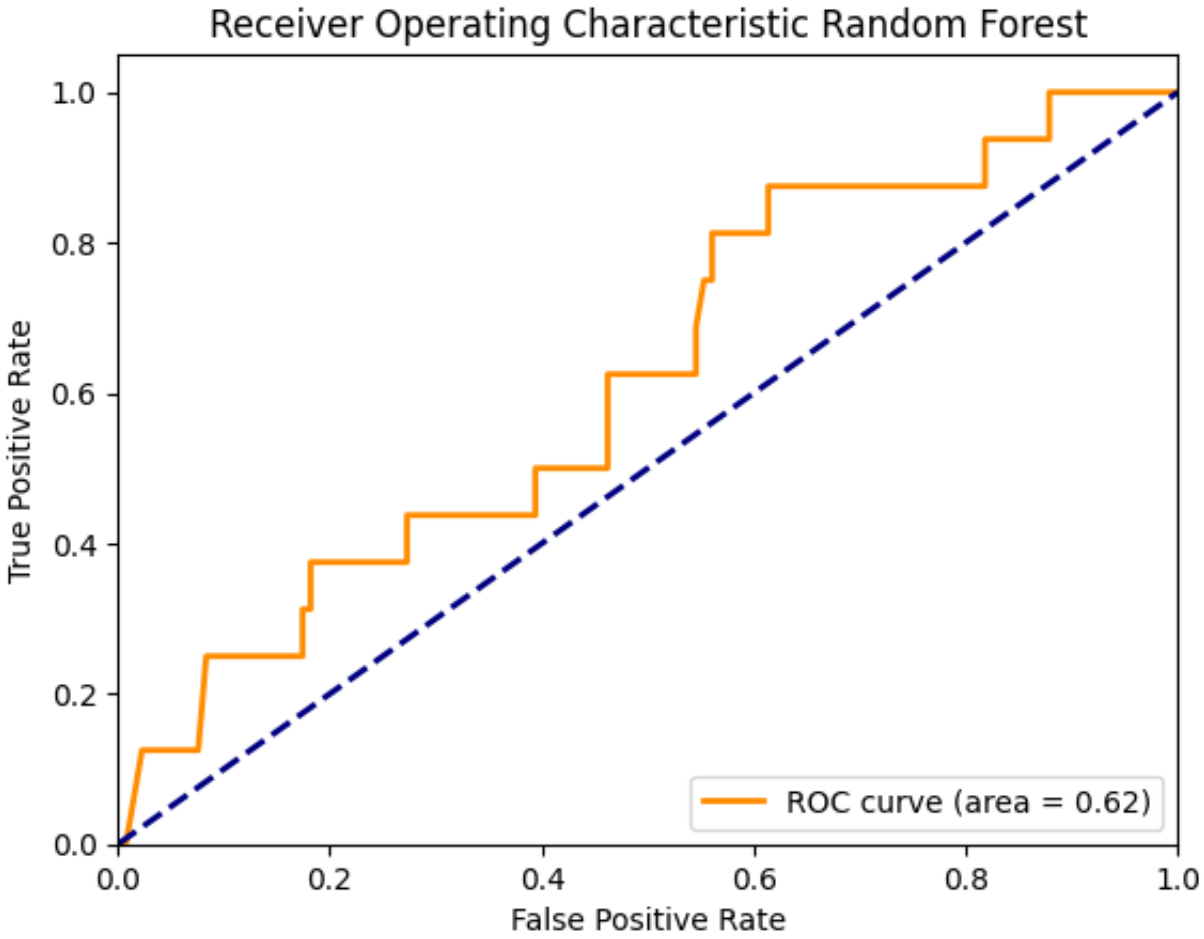

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.10606061 0.11363636 0.15151515
 0.15151515 0.15909091 0.15909091 0.16666667 0.18181818 0.26515152
 0.28787879 0.31060606 0.31060606 0.34090909 0.35606061 0.36363636
 0.38636364 0.40151515 0.41666667 0.42424242 0.42424242 0.43181818
 0.4469697  0.45454545 0.45454545 0.46212121 0.46212121 0.49242424
 0.50757576 0.51515152 0.51515152 0.54545455 0.5530303  0.56818182
 0.56818182 0.62878788 0.64393939 0.75757576 0.77272727 0.81060606
 0.81060606 0.81818182 0.83333333 0.89393939 0.89393939 1.        ]
TPR: [0.     0.     0.125  0.125  0.25   0.25   0.3125 0.3125 0.375  0.375
 0.375  0.375  0.375  0.375  0.4375 0.4375 0.4375 0.4375 0.4375 0.4375
 0.4375 0.4375 0.5    0.5    0.5    0.5    0.5625 0.5625 0.625  0.625
 0.625  0.625  0.6875 0.6875 0.75   0.75   0.875  0.875  0.875  0.875
 0.875  0.875  0.9375 0.9375 0.9375 0.9375 1.     1.     ]
ROC AUC: 0.618

Running evaluation with seed 47

Evaluating Random Forest with seed 47...
Best parameters for Random Forest: {'max_depth': 7, 'max_features': 'sqrt',
Training Metrics - Accuracy: 0.910, Sensitivity: 0.893, Specificity: 0.928,

Test Metrics for manual threshold 0.3:
Accuracy: 0.770, Sensitivity: 0.375, Specificity: 0.818, F1: 0.261, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.4797297297297297, 'Sensitivity': 0
Threshold: 0.15, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.6756756756756757, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.6959459459459459, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.8108108108108109, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Random Forest
```




GENDER
AGE
−0.1 0.0 0.1
−0.1 0.0 0.1
SHAP interaction val

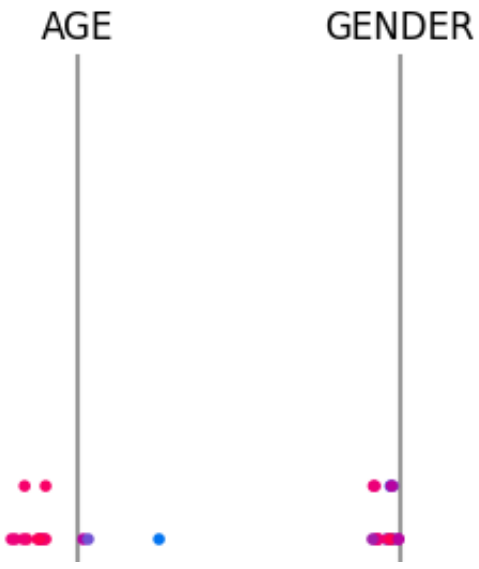

Confusion Matrix Random Forest
100

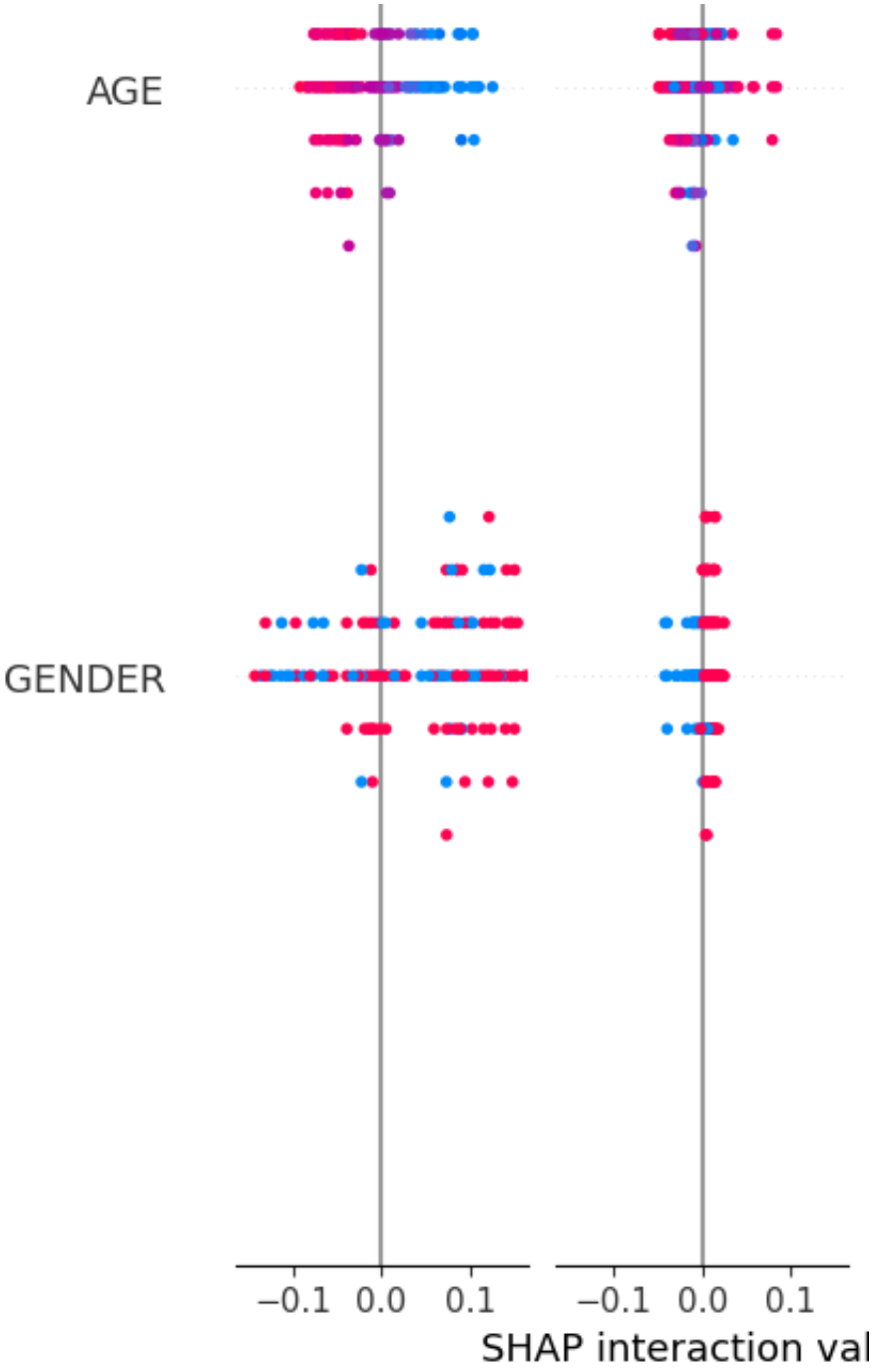


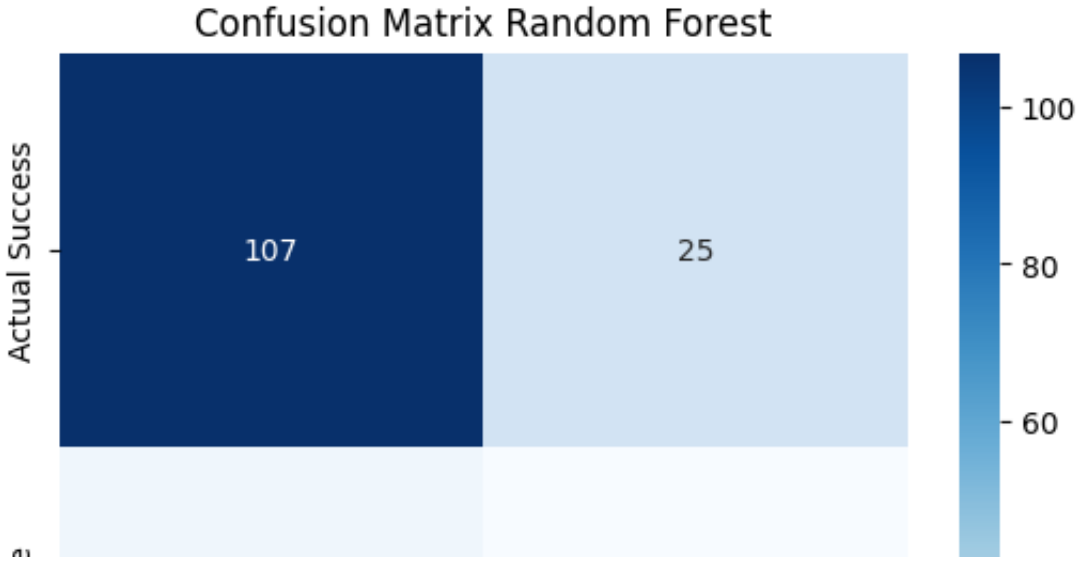


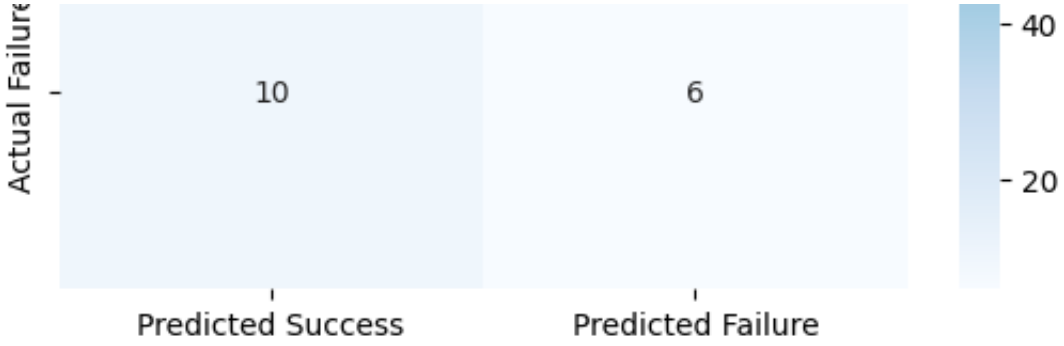

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.07575758 0.08333333 0.14393939
 0.15909091 0.17424242 0.17424242 0.18181818 0.18181818 0.21969697
 0.24242424 0.27272727 0.27272727 0.32575758 0.34090909 0.36363636
 0.39393939 0.39393939 0.42424242 0.43939394 0.46212121 0.46212121
 0.47727273 0.49242424 0.51515152 0.53030303 0.54545455 0.54545455
```

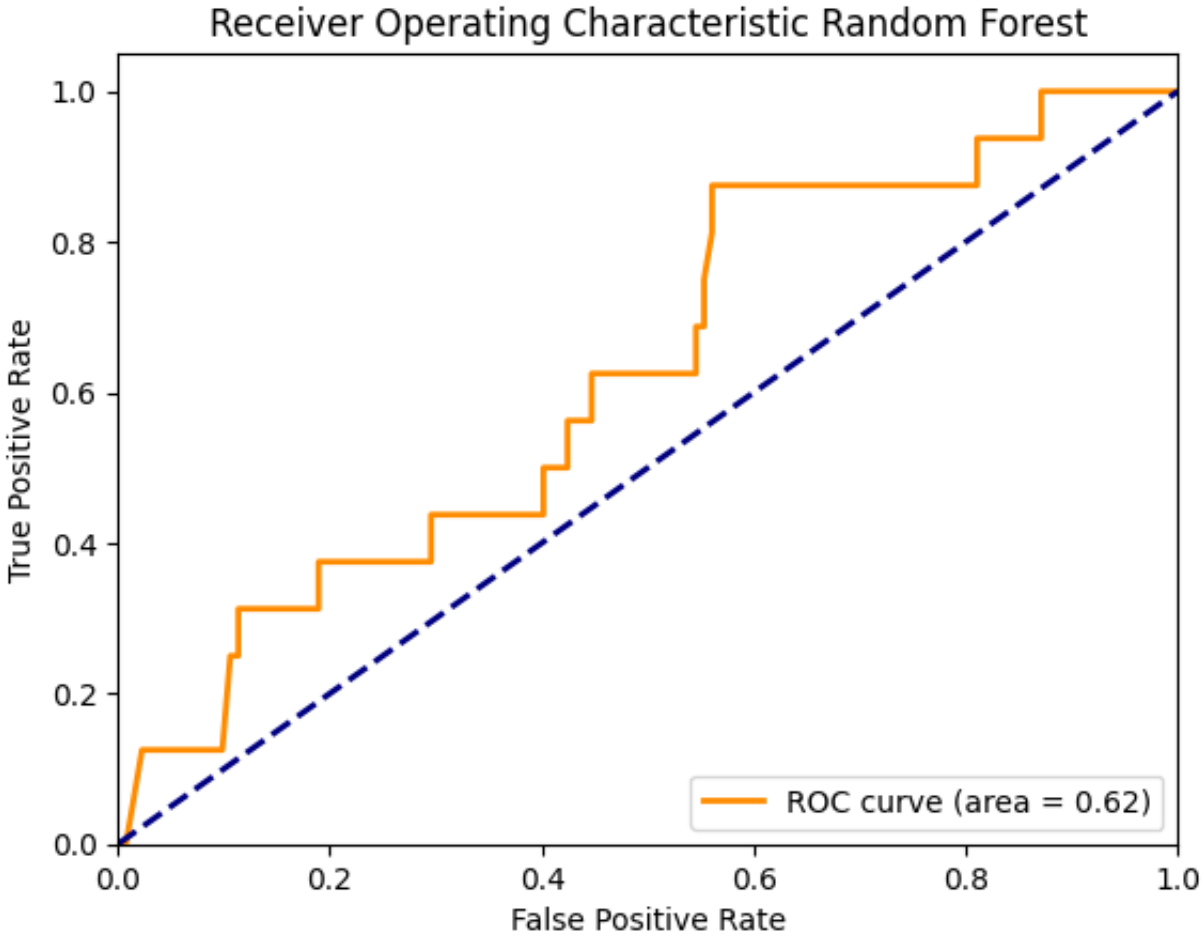
```
 0.5530303  0.56060606 0.56060606 0.61363636 0.61363636 0.62121212
 0.63636364 0.75757576 0.77272727 0.79545455 0.81060606 0.81818182
 0.81818182 0.87878788 0.87878788 1.        ]
TPR: [0.     0.     0.125  0.125  0.25   0.25   0.25   0.25   0.3125 0.3125
 0.375  0.375  0.375  0.375  0.4375 0.4375 0.4375 0.4375 0.4375 0.5
 0.5    0.5    0.5    0.625  0.625  0.625  0.625  0.625  0.625  0.6875
 0.75   0.75   0.8125 0.8125 0.875  0.875  0.875  0.875  0.875  0.875
 0.875  0.875  0.9375 0.9375 1.     1.    ]
ROC AUC: 0.619

Running evaluation with seed 48

Evaluating Random Forest with seed 48...
Best parameters for Random Forest: {'max_depth': 7, 'max_features': 'sqrt',
Training Metrics - Accuracy: 0.920, Sensitivity: 0.906, Specificity: 0.935,

Test Metrics for manual threshold 0.3:
Accuracy: 0.764, Sensitivity: 0.375, Specificity: 0.811, F1: 0.255, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.4864864864864865, 'Sensitivity': 0
Threshold: 0.15, Metrics: {'Accuracy': 0.5945945945945946, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.6621621621621622, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.7027027027027027, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.7972972972972973, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8513513513513513, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Random Forest
```




AGE
GENDER
−0.1 0.0 0.1
−0.1 0.0 0.1
SHAP interaction val

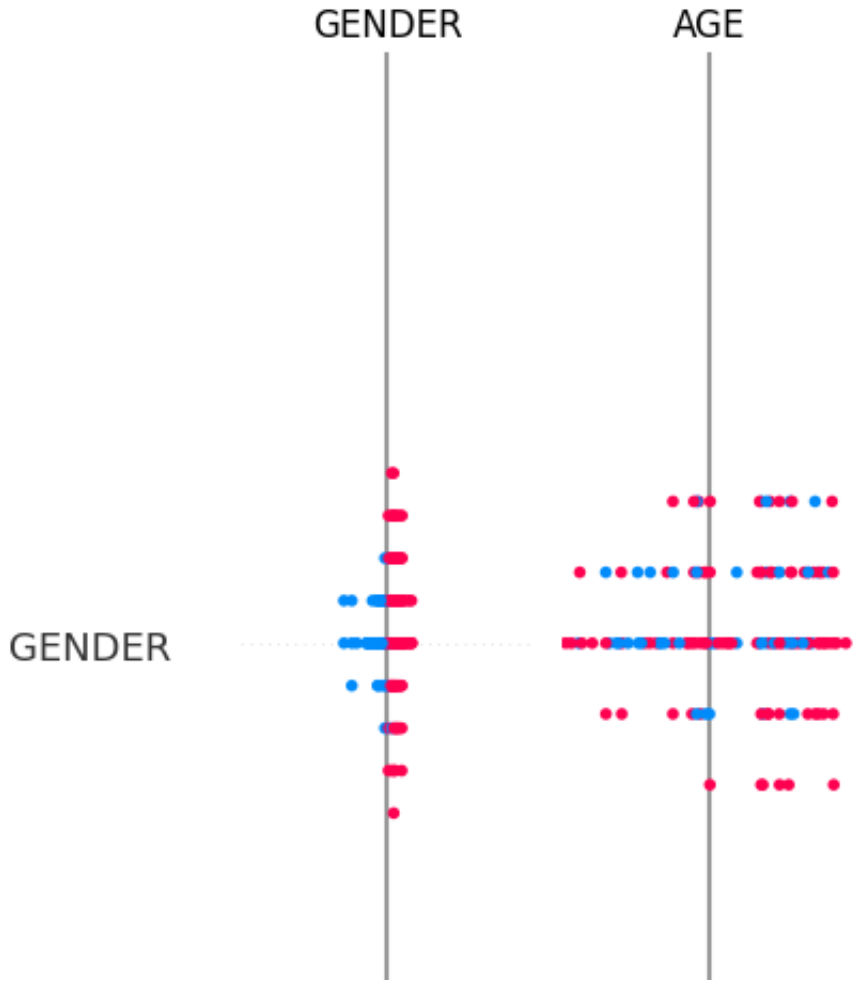

Confusion Matrix Random Forest
Actual Success
107
25
100
80
60

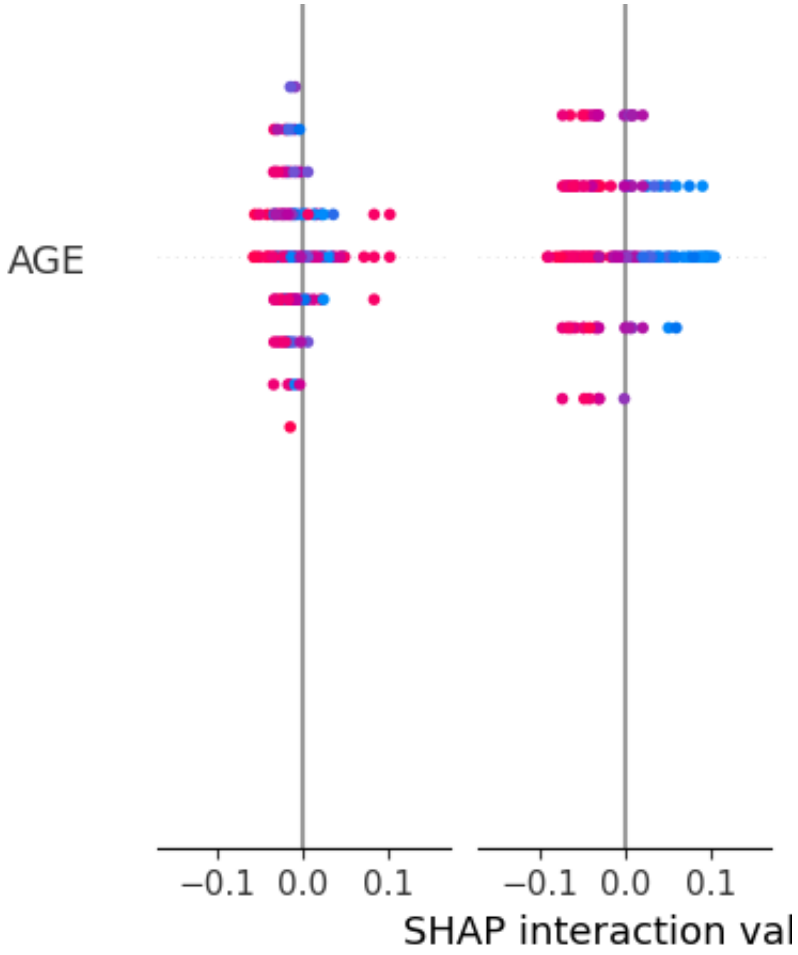


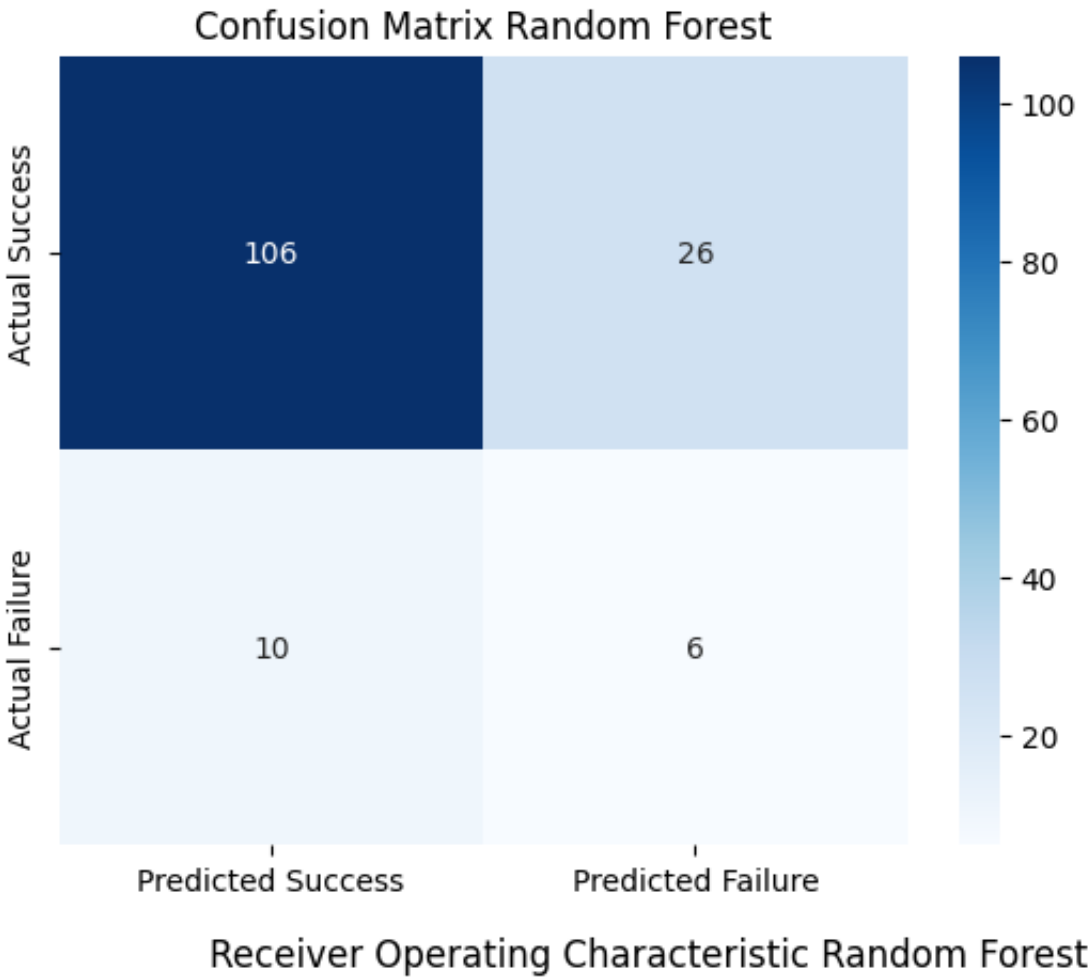



```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.09848485 0.10606061 0.11363636
 0.11363636 0.16666667 0.18181818 0.18939394 0.18939394 0.21969697
 0.24242424 0.29545455 0.29545455 0.32575758 0.34090909 0.34848485
 0.37121212 0.40151515 0.40151515 0.42424242 0.42424242 0.43181818
 0.4469697  0.4469697  0.46212121 0.5        0.51515152 0.54545455
 0.54545455 0.5530303  0.5530303  0.56060606 0.56060606 0.59848485
 0.61363636 0.76515152 0.78030303 0.79545455 0.81060606 0.81060606
 0.87121212 0.87121212 1.        ]
TPR: [0.     0.     0.125  0.125  0.25   0.25   0.3125 0.3125 0.3125 0.3125
 0.375  0.375  0.375  0.375  0.4375 0.4375 0.4375 0.4375 0.4375 0.4375
 0.5    0.5    0.5625 0.5625 0.5625 0.625  0.625  0.625  0.625  0.625
 0.6875 0.6875 0.75   0.8125 0.875  0.875  0.875  0.875  0.875  0.875
 0.875  0.9375 0.9375 1.     1.     ]
ROC AUC: 0.625

Running evaluation with seed 49
```

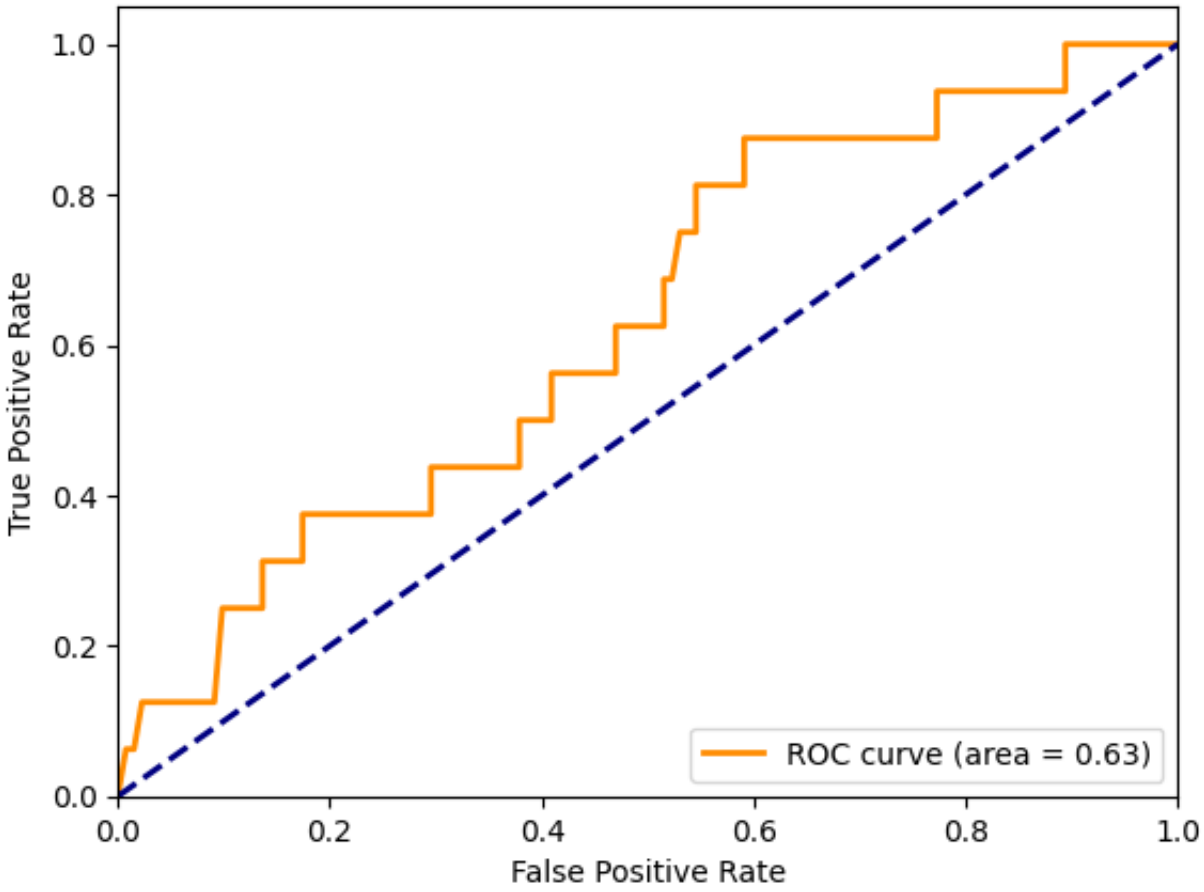
```
Evaluating Random Forest with seed 49...
Best parameters for Random Forest: {'max_depth': 7, 'max_features': 'sqrt',
Training Metrics - Accuracy: 0.914, Sensitivity: 0.896, Specificity: 0.932,

Test Metrics for manual threshold 0.3:
Accuracy: 0.757, Sensitivity: 0.375, Specificity: 0.803, F1: 0.250, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.49324324324324326, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.581081081081081, 'Sensitivity': 0.
Threshold: 0.20, Metrics: {'Accuracy': 0.6621621621621622, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.7094594594594594, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.7972972972972973, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.45, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8513513513513513, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for Random Forest
```







Receiver Operating Characteristic Random Forest




```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.01515152 0.02272727 0.09090909 0.09848485
 0.13636364 0.13636364 0.15151515 0.16666667 0.17424242 0.17424242
 0.22727273 0.25       0.29545455 0.29545455 0.31060606 0.33333333
 0.34090909 0.35606061 0.36363636 0.37878788 0.37878788 0.40909091
 0.40909091 0.46969697 0.46969697 0.48484848 0.49242424 0.50757576
 0.51515152 0.51515152 0.52272727 0.53030303 0.54545455 0.54545455
 0.59090909 0.59090909 0.59848485 0.61363636 0.73484848 0.75
 0.77272727 0.77272727 0.84848485 0.86363636 0.89393939 0.89393939
 1.        ]
TPR: [0.     0.0625 0.0625 0.125  0.125  0.25   0.25   0.3125 0.3125 0.3125
 0.3125 0.375  0.375  0.375  0.375  0.4375 0.4375 0.4375 0.4375 0.4375
 0.4375 0.4375 0.5    0.5    0.5625 0.5625 0.625  0.625  0.625  0.625
 0.625  0.6875 0.6875 0.75   0.75   0.8125 0.8125 0.875  0.875  0.875
 0.875  0.875  0.875  0.9375 0.9375 0.9375 0.9375 1.     1.     ]
ROC AUC: 0.630

Aggregated Test Set Metrics Across Seeds:
   accuracy  sensitivity  specificity        f1   roc_auc
0  0.763514       0.3750     0.810606  0.255319  0.620502
1  0.770270       0.3750     0.818182  0.260870  0.632339
2  0.756757       0.3125     0.810606  0.217391  0.627604
3  0.756757       0.3750     0.803030  0.250000  0.629972
4  0.756757       0.3750     0.803030  0.250000  0.622869
5  0.756757       0.3125     0.810606  0.217391  0.615767
6  0.750000       0.3750     0.795455  0.244898  0.617661
7  0.770270       0.3750     0.818182  0.260870  0.618608
8  0.763514       0.3750     0.810606  0.255319  0.624763
9  0.756757       0.3750     0.803030  0.250000  0.629972
```

```
Summary of Test Set Metrics (Mean, Standard Error, 95% Confidence Interval)
Accuracy: Mean = 0.760, SE = 0.002, 95% CI = [0.755, 0.765]
Sensitivity: Mean = 0.362, SE = 0.008, 95% CI = [0.344, 0.381]
Specificity: Mean = 0.808, SE = 0.002, 95% CI = [0.803, 0.813]
F1: Mean = 0.246, SE = 0.005, 95% CI = [0.235, 0.258]
Roc_auc: Mean = 0.624, SE = 0.002, 95% CI = [0.620, 0.628]
```

```
def evaluate_model(model, name, grid, X_train, y_train, X_test, y_test, cv, scor
    print(f"\nEvaluating {name} with seed {seed}...")

    # Define inner and outer CV splits using the provided seed
    inner_cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=seed)
    outer_cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=seed)

    # Grid search using inner CV
    clf = GridSearchCV(model, grid, cv=inner_cv, scoring='roc_auc')
    nested_scores = cross_validate(clf, X=X_train, y=y_train, cv=outer_cv, scori

    # Fit grid search on full training set and extract the best estimator
    clf.fit(X_train, y_train)
    best_model = clf.best_estimator_
    best_params = clf.best_params_
    print(f"Best parameters for {name}: {best_params}")

    # Calibrate the best model
    calibrated_clf = CalibratedClassifierCV(estimator=best_model, method='sigmoi
    calibrated_clf.fit(X_train, y_train)

    # Get predicted probabilities on the test set from the calibrated classifier
    y_probs = calibrated_clf.predict_proba(X_test)[:, 1]

    # --- Calculate Training Metrics ---
    y_train_pred = best_model.predict(X_train)
    y_train_probs = best_model.predict_proba(X_train)[:, 1]
    train_acc    = accuracy_score(y_train, y_train_pred)
    train_sens   = sensitivity(y_train, y_train_pred)
    train_spec   = specificity(y_train, y_train_pred)
    train_f1     = f1_score(y_train, y_train_pred)
    train_roc_auc = roc_auc_score(y_train, y_train_probs)

    print(f"Training - Accuracy: {train_acc:.3f}, Sensitivity: {train_sens:.3f},
          f"Specificity: {train_spec:.3f}, F1: {train_f1:.3f}, ROC AUC: {train_r

    # --- Calculate Test Metrics for the manually set threshold ---
    y_pred_manual = (y_probs >= manual_threshold).astype(int)
    manual_acc    = accuracy_score(y_test, y_pred_manual)
```

```
    manual_sens   = sensitivity(y_test, y_pred_manual)
    manual_spec   = specificity(y_test, y_pred_manual)
    manual_f1     = f1_score(y_test, y_pred_manual)
    manual_roc_auc = roc_auc_score(y_test, y_probs)

    print(f"\nTest Metrics for manual threshold {manual_threshold}:")
    print(f"Accuracy: {manual_acc:.3f}, Sensitivity: {manual_sens:.3f}, "
          f"Specificity: {manual_spec:.3f}, F1: {manual_f1:.3f}, ROC AUC: {manua

    # --- Evaluate metrics across a range of thresholds ---
    threshold_metrics = {}
    for threshold in threshold_list:
        y_pred_threshold = (y_probs >= threshold).astype(int)
        threshold_acc    = accuracy_score(y_test, y_pred_threshold)
        threshold_sens   = sensitivity(y_test, y_pred_threshold)
        threshold_spec   = specificity(y_test, y_pred_threshold)
        threshold_f1     = f1_score(y_test, y_pred_threshold)
        threshold_metrics[threshold] = {
            'Accuracy': threshold_acc,
            'Sensitivity': threshold_sens,
            'Specificity': threshold_spec,
            'F1': threshold_f1,
            'ROC AUC': manual_roc_auc  # same ROC AUC regardless of threshold
        }
    for threshold, metrics in threshold_metrics.items():
        print(f"Threshold: {threshold:.2f}, Metrics: {metrics}")

    # Plot SHAP summary (using X_train as the background sample)
    calculate_and_plot_shap(best_model, X_train, X_test, name)

    # Prepare dictionary of test metrics at the manual threshold for aggregation
    test_metrics = {
        "accuracy": manual_acc,
        "sensitivity": manual_sens,
        "specificity": manual_spec,
        "f1": manual_f1,
        "roc_auc": manual_roc_auc
    }

    return best_model, manual_threshold, best_params, nested_scores, calibrated_

# --- SHAP Plotting Function ---
def calculate_and_plot_shap(model, X_train, X_test, model_name):
    # Use TreeExplainer if model is an XGBClassifier; otherwise, use KernelExpla
    if isinstance(model, XGBClassifier):
        explainer = shap.TreeExplainer(model)
    else:
        explainer = shap.KernelExplainer(model.predict_proba, X_train.sample(100
    shap_values = explainer.shap_values(X_test)
```

```
    shap_values = explainer.shap_values(X_test)
    print(f"SHAP Summary for {model_name}")
    shap.summary_plot(shap_values, X_test, max_display=10)

# --- Plotting Functions ---
def plot_confusion_matrix(y_true, y_pred):
    matrix = confusion_matrix(y_true, y_pred)
    sns.heatmap(matrix, annot=True, fmt='d', cmap='Blues',
                xticklabels=['Predicted Success', 'Predicted Failure'],
                yticklabels=['Actual Success', 'Actual Failure'])
    plt.title('Confusion Matrix XGBoosting')
    plt.show()

def plot_roc_curve(y_true, y_probs):
    fpr, tpr, thresholds = roc_curve(y_true, y_probs)
    roc_auc = auc(fpr, tpr)

    plt.figure()
    plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area = {roc_
    plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver Operating Characteristic XGBoosting')
    plt.legend(loc="lower right")
    plt.show()

    # --- Added code to output ROC curve metrics ---
    print("ROC Curve Metrics:")
    print("FPR:", fpr)
    print("TPR:", tpr)
    print("ROC AUC: {:.3f}".format(roc_auc))

    return fpr, tpr, roc_auc

# --- Evaluation Function for XGBoost ---
def evaluate_xgboost(X_train_resampled, y_train_resampled, X_test, y_test, cv, s
    print("Inside evaluate_xgboost function")
    model = XGBClassifier(use_label_encoder=False, eval_metric='logloss', random
    grid = {
        'max_depth': [12],
        'gamma': [2],
        'learning_rate': [1.2],
        'subsample': [0.8],
        'colsample_bytree': [1],
        'reg_alpha': [0],
        'reg_lambda': [2],
        'n_estimators': [120]
```

```
    }

    return evaluate_model(model, "XGBoost", grid, X_train_resampled, y_train_res

# --- MAIN FUNCTION: AGGREGATING METRICS ACROSS SEEDS ---
def main(X_train_resampled, y_train_resampled, X_test, y_test):
    cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=10, random_state=42)
    scoring = {
        'accuracy': make_scorer(accuracy_score),
        'sensitivity': make_scorer(sensitivity),
        'specificity': make_scorer(specificity),
        'f1': make_scorer(f1_score),
        'roc_auc': make_scorer(roc_auc_score)
    }
    manual_threshold = 0.50
    threshold_list = np.arange(0.1, 1.05, 0.05)

    # List to collect test metrics from each seed iteration
    aggregated_metrics = []

    for seed in range(40, 50):
        print(f"\nRunning evaluation with seed {seed}")
        (best_model, manual_threshold, best_params, nested_scores,
         calibrated_clf, threshold_metrics, test_metrics) = evaluate_xgboost(
            X_train_resampled, y_train_resampled, X_test, y_test, cv, scoring, m
        )

        # Use calibrated_clf for prediction probabilities (for plotting)
        y_probs = calibrated_clf.predict_proba(X_test)[:, 1]
        y_pred_manual = (y_probs >= manual_threshold).astype(int)

        # Plot confusion matrix and ROC curve for this seed
        plot_confusion_matrix(y_test, y_pred_manual)
        plot_roc_curve(y_test, y_probs)

        # Append the test set metrics from this seed for later aggregation
        aggregated_metrics.append(test_metrics)

    # --- Aggregate Results Across Seeds ---
    results_df = pd.DataFrame(aggregated_metrics)
    n = len(results_df)
    print("\nAggregated Test Set Metrics Across Seeds:")
    print(results_df)

    # Function to compute mean, standard error, and 95% confidence interval usin
    def summarize_metric(metric_values):
        mean_val = metric_values.mean()
        std_val = metric_values.std(ddof=1)
```

```
        se = std_val / np.sqrt(n)
        t_crit = stats.t.ppf(0.975, df=n-1)  # 95% confidence, two-tailed
        ci_lower = mean_val - t_crit * se
        ci_upper = mean_val + t_crit * se
        return mean_val, se, (ci_lower, ci_upper)

    metrics_summary = {}
    for metric in results_df.columns:
        mean_val, se, ci = summarize_metric(results_df[metric])
        metrics_summary[metric] = {
            "Mean": mean_val,
            "Standard Error": se,
            "95% CI": ci
        }

    print("\nSummary of Test Set Metrics (Mean, Standard Error, 95% Confidence I
    for metric, summary in metrics_summary.items():
        print(f"{metric.capitalize()}: Mean = {summary['Mean']:.3f}, SE = {summa
              f"95% CI = [{summary['95% CI'][0]:.3f}, {summary['95% CI'][1]:.3f}

# --- RUN THE MAIN FUNCTION ---
# It is assumed that X_train_resampled, y_train_resampled, X_test, and y_test ar
if __name__ == '__main__':
    main(X_train_resampled, y_train_resampled, X_test, y_test)
```

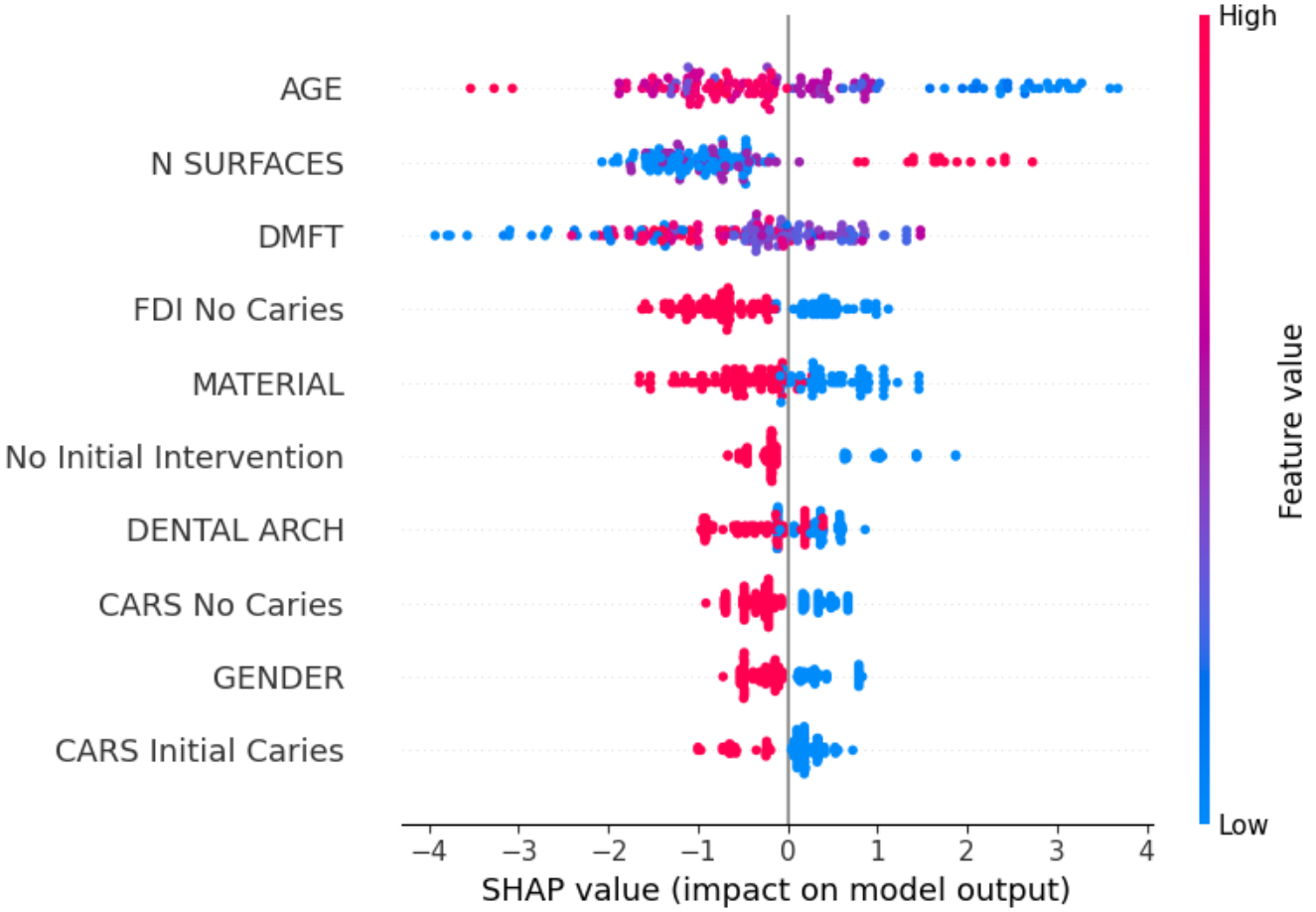
```
Running evaluation with seed 40
Inside evaluate_xgboost function

Evaluating XGBoost with seed 40...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
```

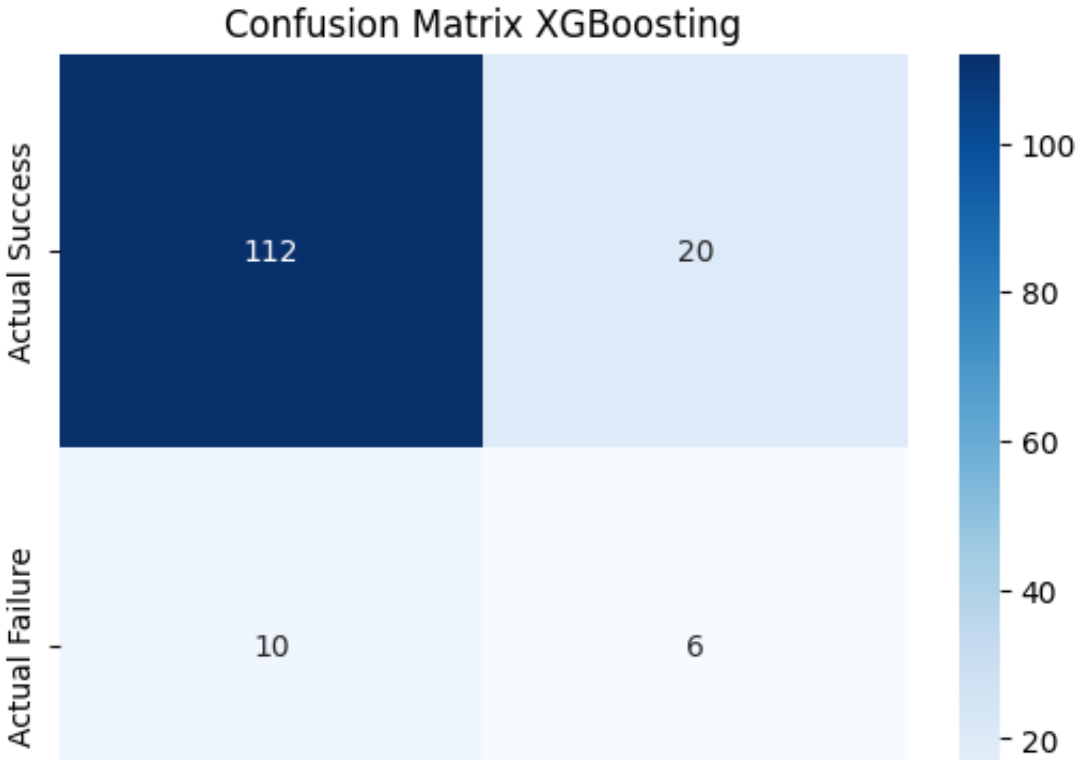
```
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

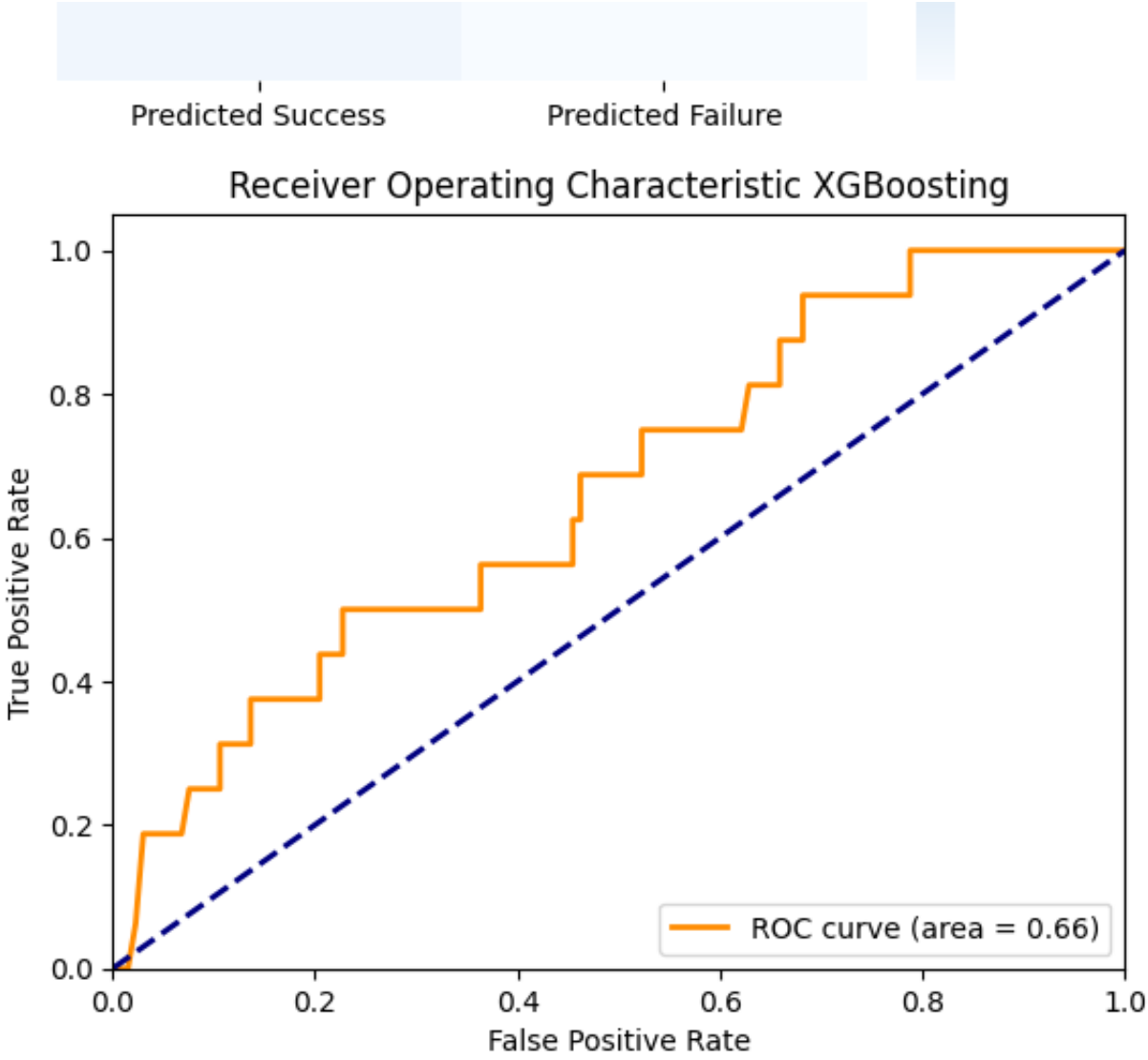
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
```

```
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Best parameters for XGBoost: {'colsample_bytree': 1, 'gamma': 2, 'learning_
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Training - Accuracy: 0.969, Sensitivity: 0.967, Specificity: 0.971, F1: 0.9

Test Metrics for manual threshold 0.5:
Accuracy: 0.797, Sensitivity: 0.375, Specificity: 0.848, F1: 0.286, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.38513513513513514, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.4864864864864865, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.5405405405405406, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.5743243243243243, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6824324324324325, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.722972972972973, 'Sensitivity': 0.
Threshold: 0.40, Metrics: {'Accuracy': 0.7432432432432432, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7972972972972973, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8040540540540541, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
```

```
Threshold: 0.75, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for XGBoost
```







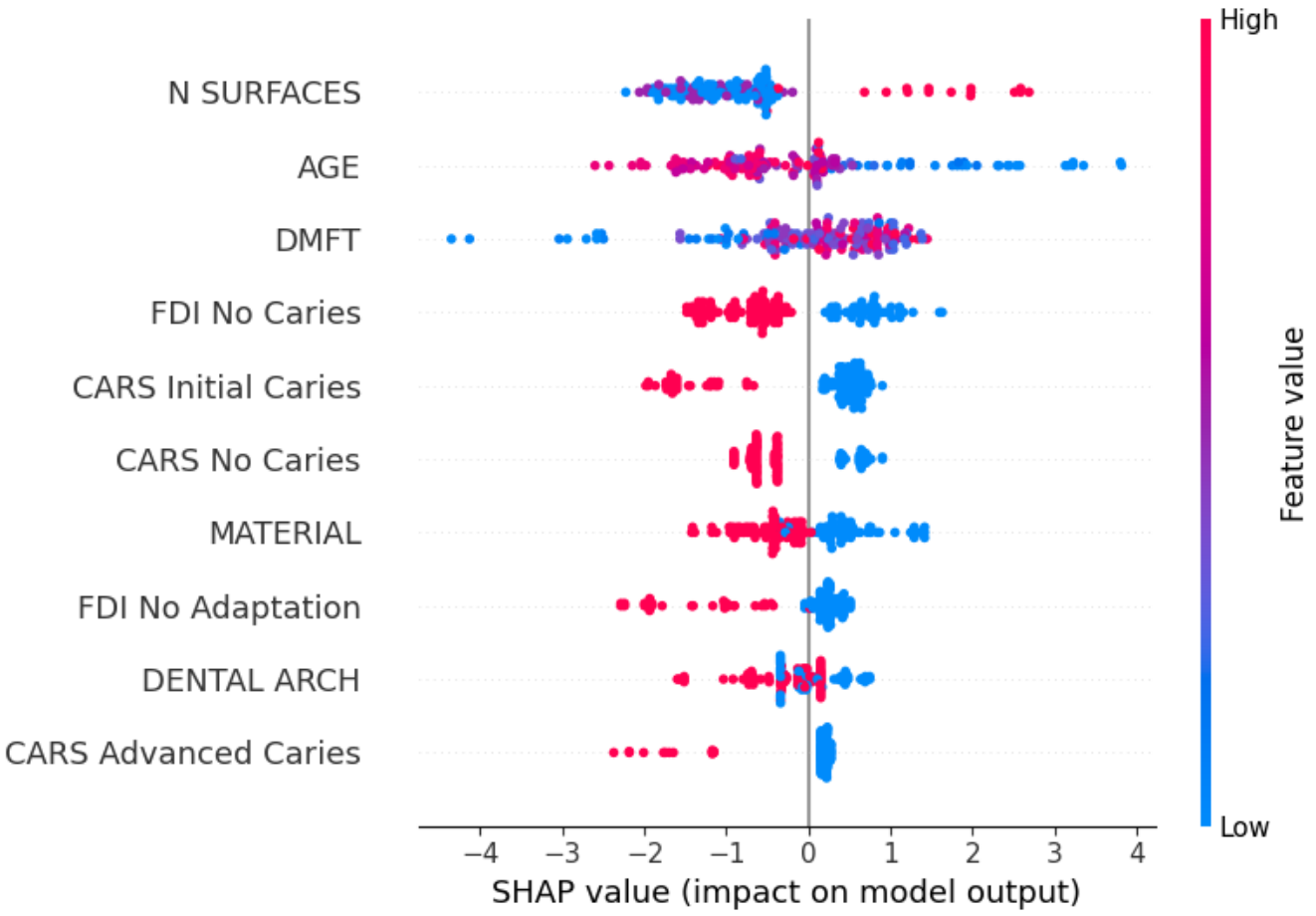

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.01515152 0.02272727 0.03030303 0.06818182
 0.07575758 0.10606061 0.10606061 0.12121212 0.13636364 0.13636364
 0.15151515 0.17424242 0.18181818 0.1969697  0.20454545 0.20454545
 0.22727273 0.22727273 0.3030303  0.31818182 0.36363636 0.36363636
 0.38636364 0.41666667 0.45454545 0.45454545 0.46212121 0.46212121
 0.52272727 0.52272727 0.53030303 0.54545455 0.62121212 0.62878788
 0.65909091 0.65909091 0.68181818 0.68181818 0.78787879 0.78787879
 0.87121212 0.88636364 0.89393939 0.90909091 1.        ]
TPR: [0.     0.     0.     0.0625 0.1875 0.1875 0.25   0.25   0.3125 0.3125
 0.3125 0.375  0.375  0.375  0.375  0.375  0.375  0.4375 0.4375 0.5
 0.5    0.5    0.5    0.5625 0.5625 0.5625 0.5625 0.625  0.625  0.6875
 0.6875 0.75   0.75   0.75   0.75   0.8125 0.8125 0.875  0.875  0.9375
 0.9375 1.     1.     1.     1.     1.     1.    ]
ROC AUC: 0.664

Running evaluation with seed 41
Inside evaluate_xgboost function

Evaluating XGBoost with seed 41...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

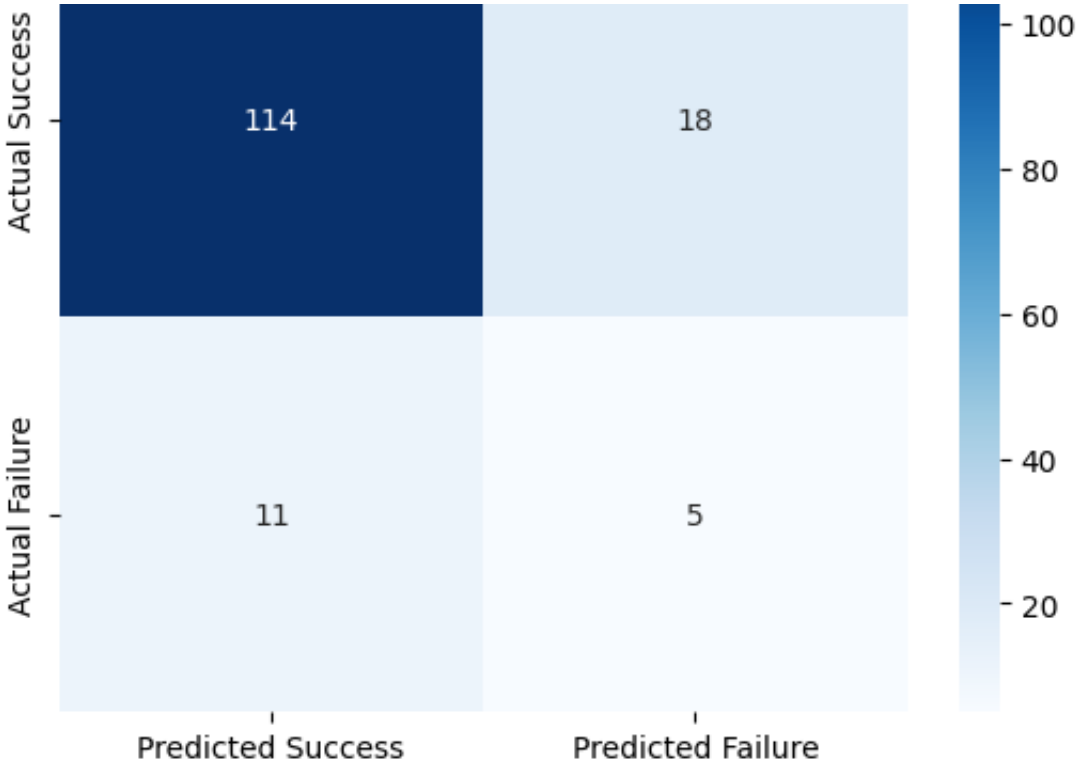
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
```

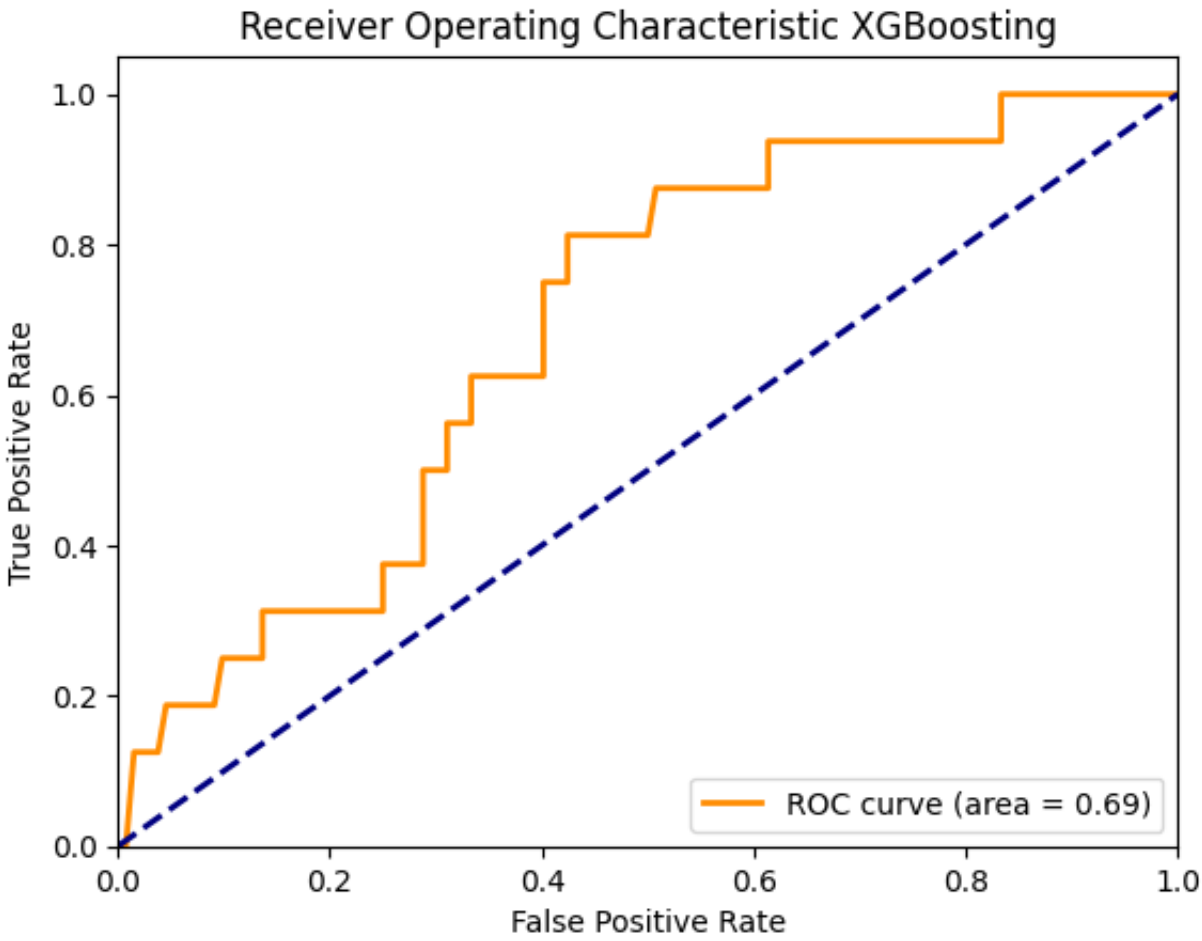
```
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Best parameters for XGBoost: {'colsample_bytree': 1, 'gamma': 2, 'learning_
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
```

```
Training - Accuracy: 0.953, Sensitivity: 0.951, Specificity: 0.954, F1: 0.9

Test Metrics for manual threshold 0.5:
Accuracy: 0.804, Sensitivity: 0.312, Specificity: 0.864, F1: 0.256, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.4391891891891892, 'Sensitivity': 0
Threshold: 0.15, Metrics: {'Accuracy': 0.5675675675675675, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.6216216216216216, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.6621621621621622, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6824324324324325, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.722972972972973, 'Sensitivity': 0.
Threshold: 0.40, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7837837837837838, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.8040540540540541, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8108108108108109, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for XGBoost
```







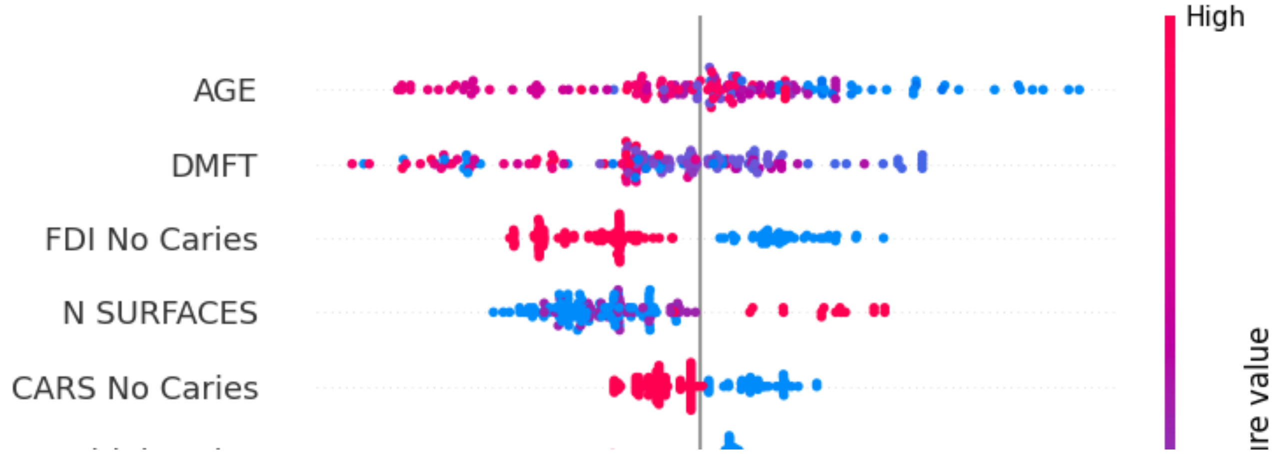

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.01515152 0.03787879 0.04545455 0.09090909
 0.09848485 0.13636364 0.13636364 0.15909091 0.17424242 0.18939394
 0.21212121 0.22727273 0.24242424 0.25       0.25       0.28787879
 0.28787879 0.31060606 0.31060606 0.33333333 0.33333333 0.35606061
 0.36363636 0.39393939 0.40151515 0.40151515 0.42424242 0.42424242
```

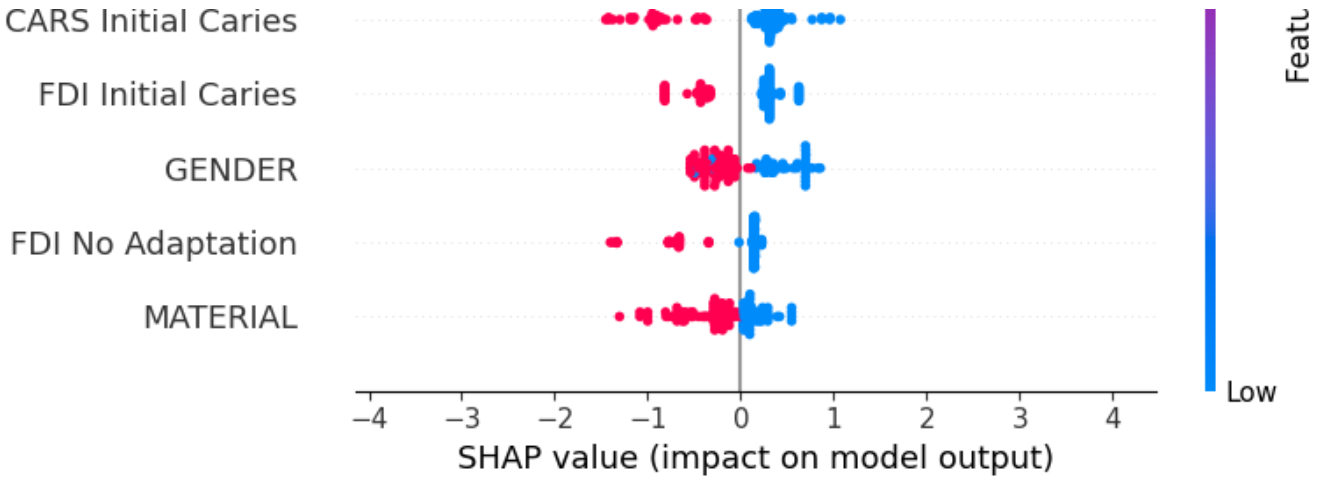
```
 0.43939394 0.5        0.50757576 0.53030303 0.54545455 0.61363636
 0.61363636 0.75       0.76515152 0.83333333 0.83333333 0.85606061
 0.87121212 1.        ]
TPR: [0.     0.     0.125  0.125  0.1875 0.1875 0.25   0.25   0.3125 0.3125
 0.3125 0.3125 0.3125 0.3125 0.3125 0.3125 0.375  0.375  0.5    0.5
 0.5625 0.5625 0.625  0.625  0.625  0.625  0.625  0.75   0.75   0.8125
 0.8125 0.8125 0.875  0.875  0.875  0.875  0.9375 0.9375 0.9375 0.9375
 1.     1.     1.     1.    ]
ROC AUC: 0.691

Running evaluation with seed 42
Inside evaluate_xgboost function

Evaluating XGBoost with seed 42...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
```

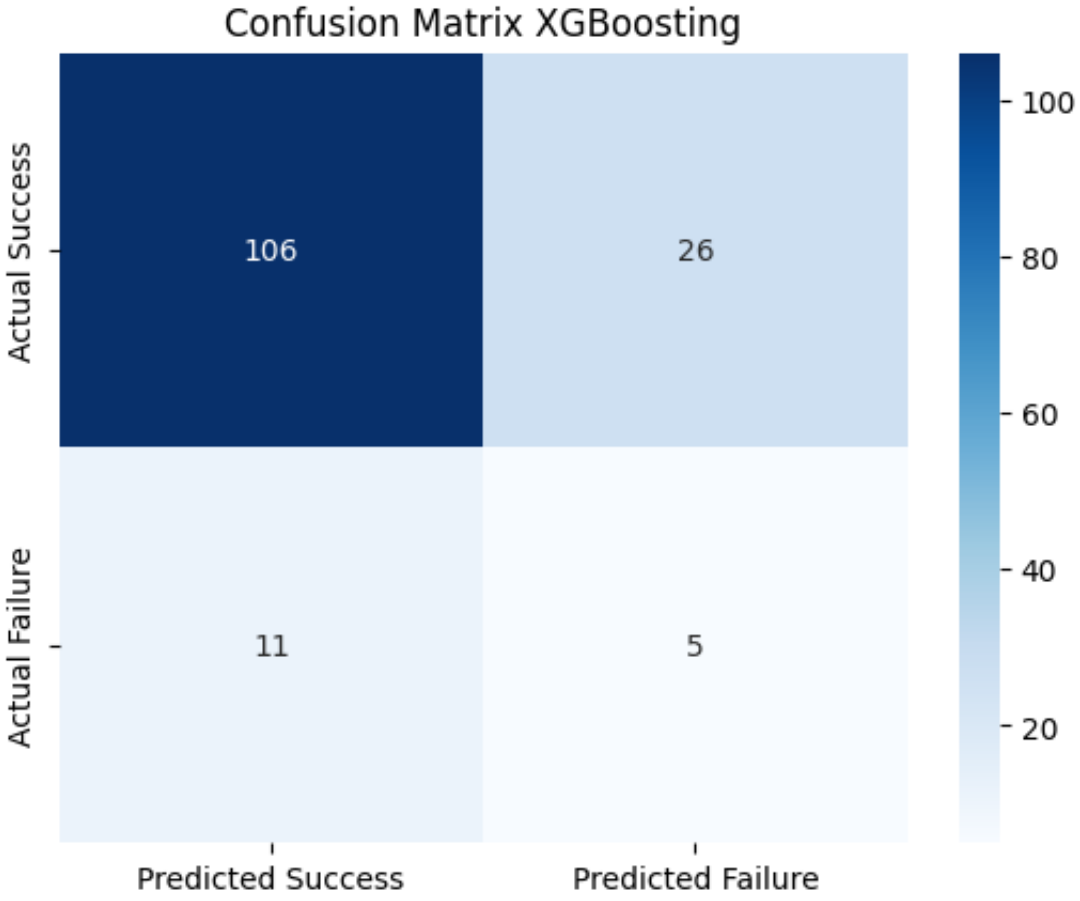
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

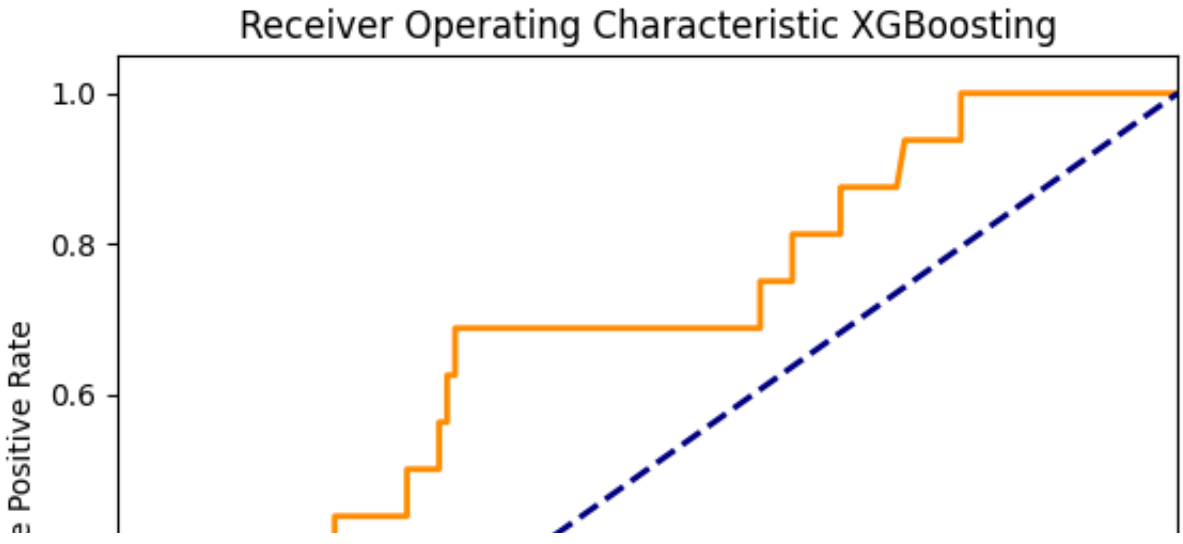
```
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Best parameters for XGBoost: {'colsample_bytree': 1, 'gamma': 2, 'learning_
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

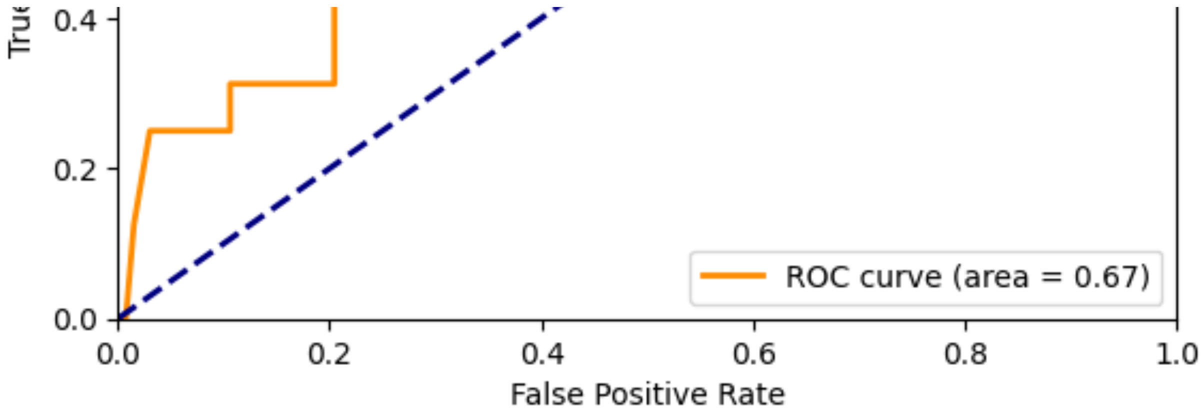
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Training - Accuracy: 0.954, Sensitivity: 0.951, Specificity: 0.958, F1: 0.9

Test Metrics for manual threshold 0.5:
Accuracy: 0.750, Sensitivity: 0.312, Specificity: 0.803, F1: 0.213, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.41216216216216217, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.5135135135135135, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.581081081081081, 'Sensitivity': 0.
Threshold: 0.25, Metrics: {'Accuracy': 0.6216216216216216, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6756756756756757, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.7027027027027027, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7027027027027027, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7432432432432432, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.75, 'Sensitivity': 0.3125, 'Specif
Threshold: 0.55, Metrics: {'Accuracy': 0.777027027027027, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.8108108108108109, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for XGBoost
```




CARS Initial Caries
FDI Initial Caries
GENDER
FDI No Adaptation
MATERIAL
−4
−3
−2
−1
0
1
2
3
4
SHAP value (impact on model output)
Featu
Low


Confusion Matrix XGBoosting
Actual Success
Actual Failure
106
26
11
5
Predicted Success
Predicted Failure
100
80
60
40
20


Receiver Operating Characteristic XGBoosting
1.0
0.8
0.6
Positive Rate



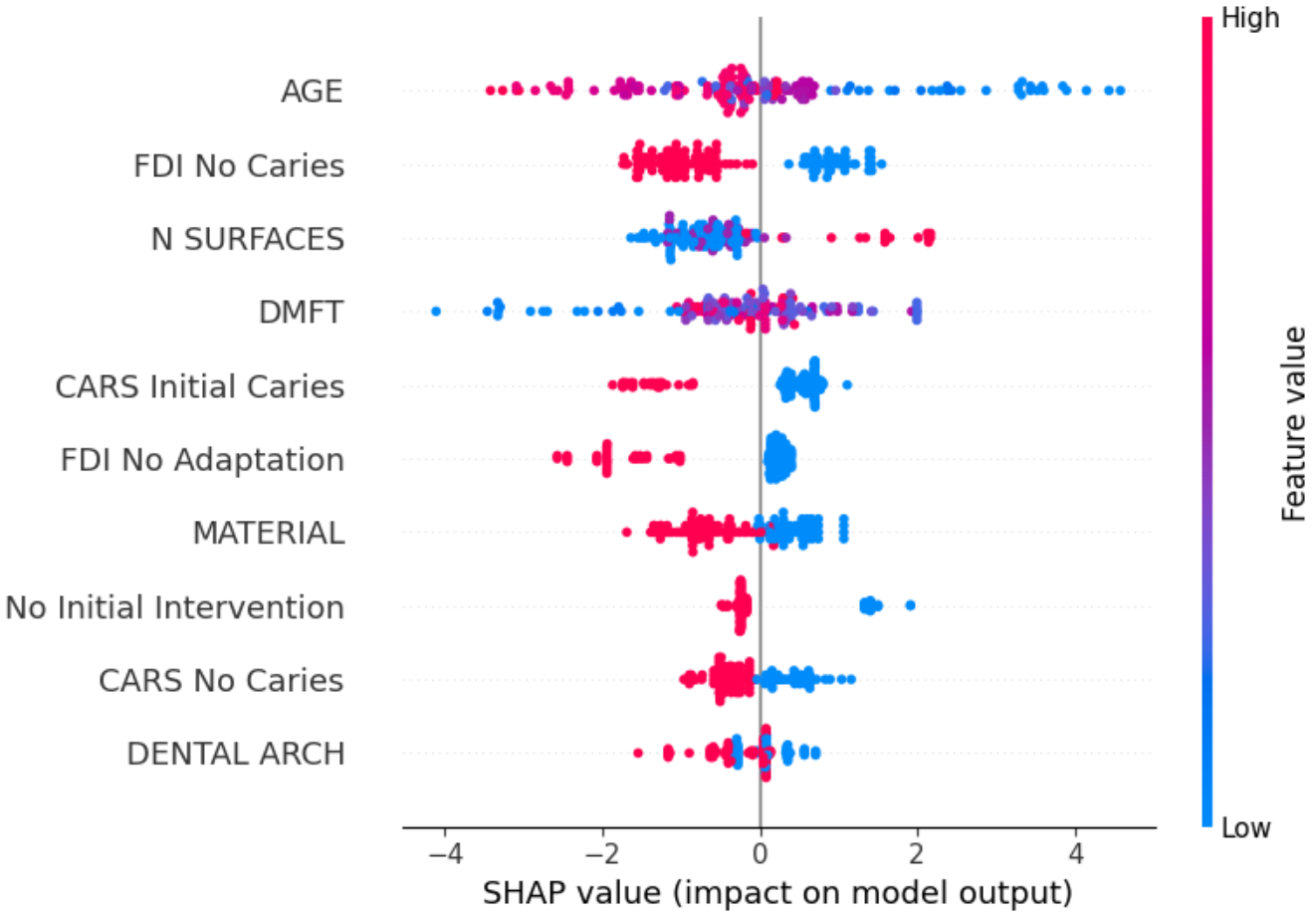

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.01515152 0.03030303 0.07575758 0.09090909
 0.10606061 0.10606061 0.14393939 0.16666667 0.17424242 0.18939394
 0.20454545 0.20454545 0.21969697 0.23484848 0.27272727 0.27272727
 0.28030303 0.29545455 0.3030303  0.3030303  0.31060606 0.31060606
 0.31818182 0.31818182 0.34090909 0.49242424 0.50757576 0.52272727
 0.53787879 0.60606061 0.60606061 0.62121212 0.63636364 0.63636364
 0.68181818 0.68181818 0.73484848 0.74242424 0.79545455 0.79545455
 0.87121212 0.88636364 1.        ]
TPR: [0.     0.     0.125  0.25   0.25   0.25   0.25   0.3125 0.3125 0.3125
 0.3125 0.3125 0.3125 0.4375 0.4375 0.4375 0.4375 0.5    0.5    0.5
 0.5    0.5625 0.5625 0.625  0.625  0.6875 0.6875 0.6875 0.6875 0.6875
 0.6875 0.6875 0.75   0.75   0.75   0.8125 0.8125 0.875  0.875  0.9375
 0.9375 1.     1.     1.     1.    ]
ROC AUC: 0.672

Running evaluation with seed 43
Inside evaluate_xgboost function

Evaluating XGBoost with seed 43...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

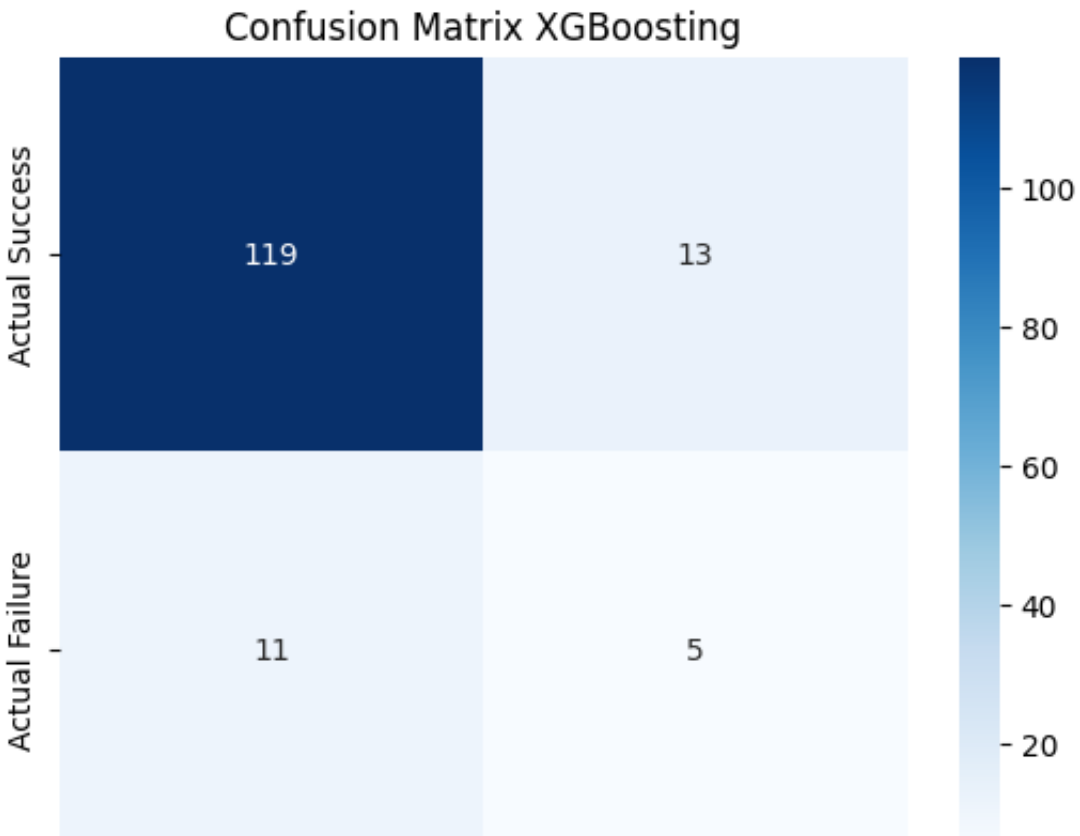
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
```

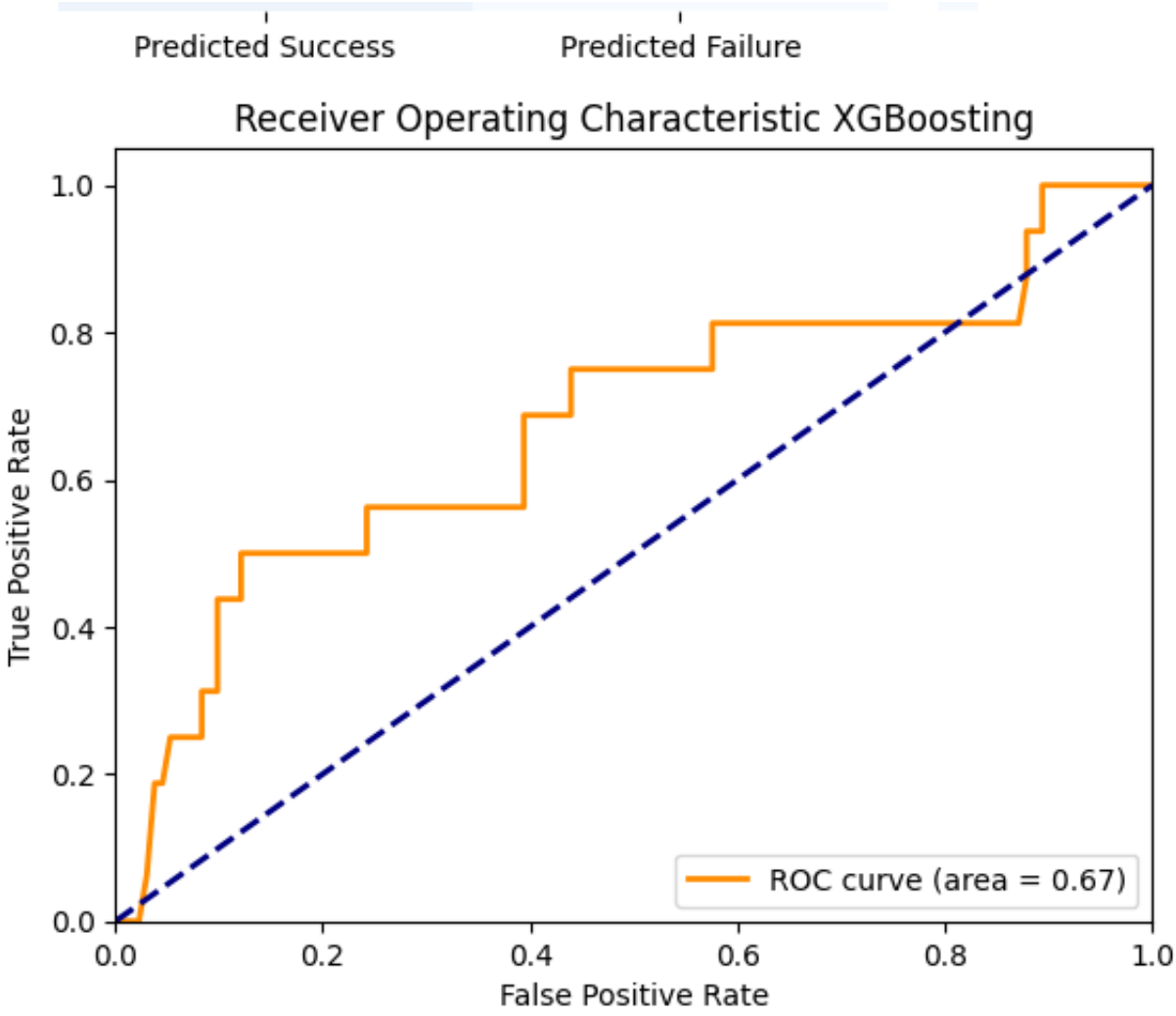
```
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Best parameters for XGBoost: {'colsample_bytree': 1, 'gamma': 2, 'learning_
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Training - Accuracy: 0.956, Sensitivity: 0.958, Specificity: 0.954, F1: 0.9

Test Metrics for manual threshold 0.5:
Accuracy: 0.838, Sensitivity: 0.312, Specificity: 0.902, F1: 0.294, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.46621621621621623, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.581081081081081, 'Sensitivity': 0.
Threshold: 0.20, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.6621621621621622, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7297297297297297, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.45, Metrics: {'Accuracy': 0.8513513513513513, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
```

```
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for XGBoost
```







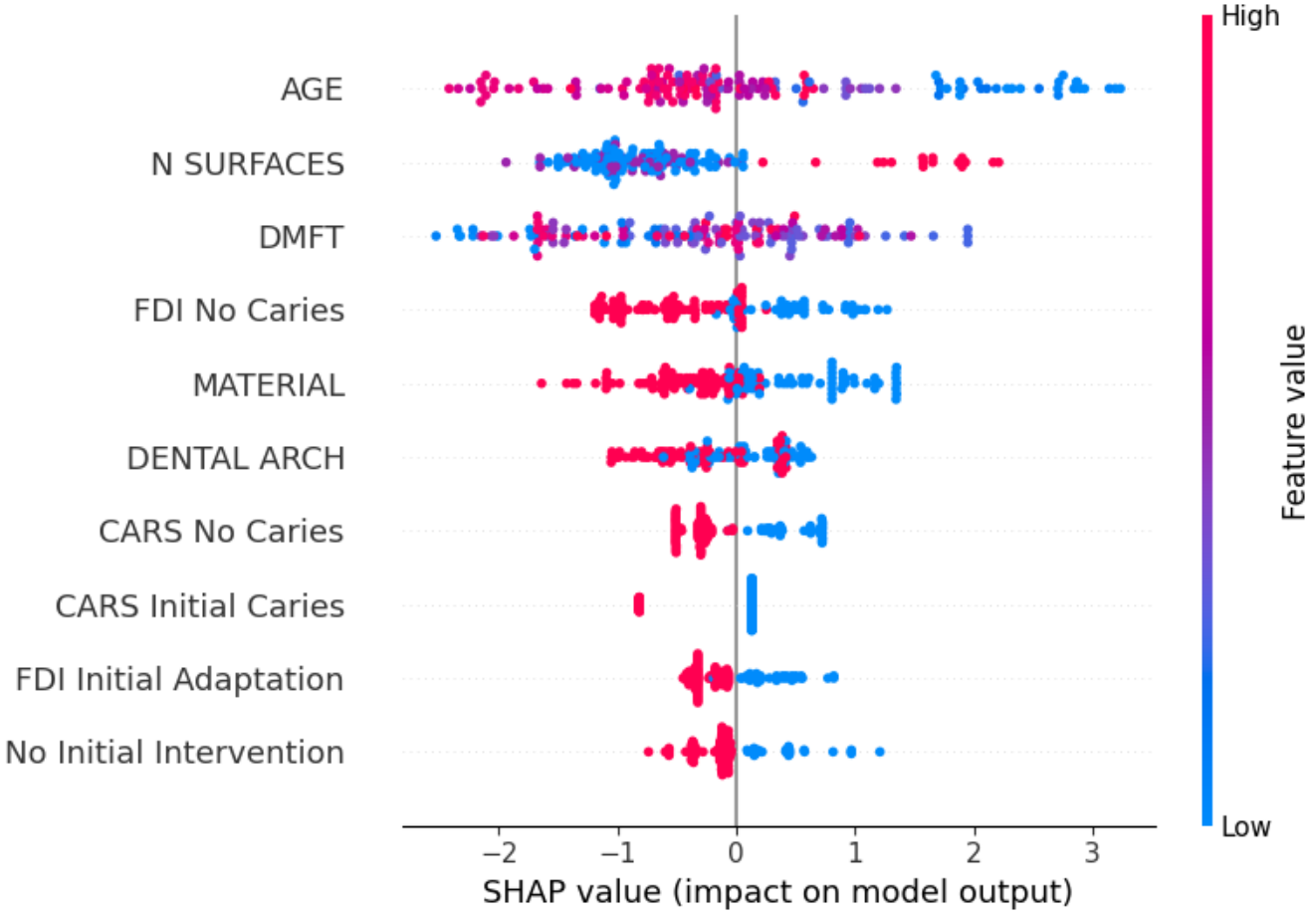

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.03030303 0.03787879 0.04545455
 0.0530303  0.08333333 0.08333333 0.09848485 0.09848485 0.12121212
 0.12121212 0.15151515 0.16666667 0.24242424 0.24242424 0.26515152
 0.27272727 0.29545455 0.31060606 0.32575758 0.34090909 0.35606061
 0.39393939 0.39393939 0.43939394 0.43939394 0.45454545 0.48484848
 0.5        0.54545455 0.56060606 0.57575758 0.57575758 0.87121212
 0.87878788 0.87878788 0.89393939 0.89393939 0.90909091 0.92424242
 0.96212121 0.97727273 1.        ]
TPR: [0.     0.     0.     0.0625 0.1875 0.1875 0.25   0.25   0.3125 0.3125
 0.4375 0.4375 0.5    0.5    0.5    0.5    0.5625 0.5625 0.5625 0.5625
 0.5625 0.5625 0.5625 0.5625 0.5625 0.6875 0.6875 0.75   0.75   0.75
 0.75   0.75   0.75   0.75   0.8125 0.8125 0.875  0.9375 0.9375 1.
 1.     1.     1.     1.     1.    ]
ROC AUC: 0.673

Running evaluation with seed 44
Inside evaluate_xgboost function

Evaluating XGBoost with seed 44...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
```

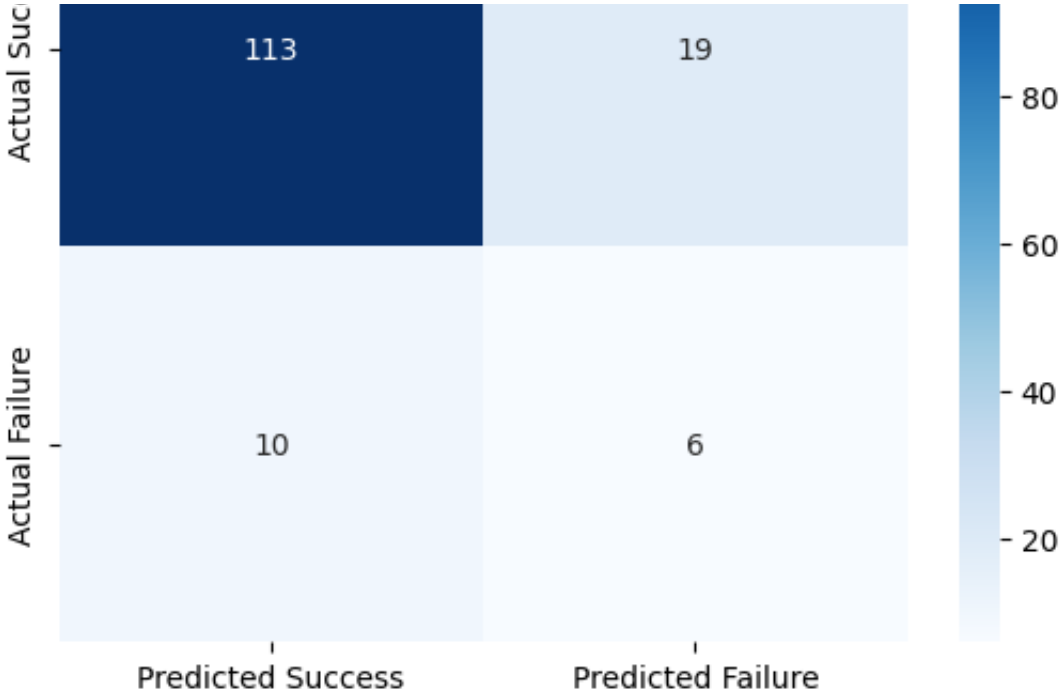
```
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

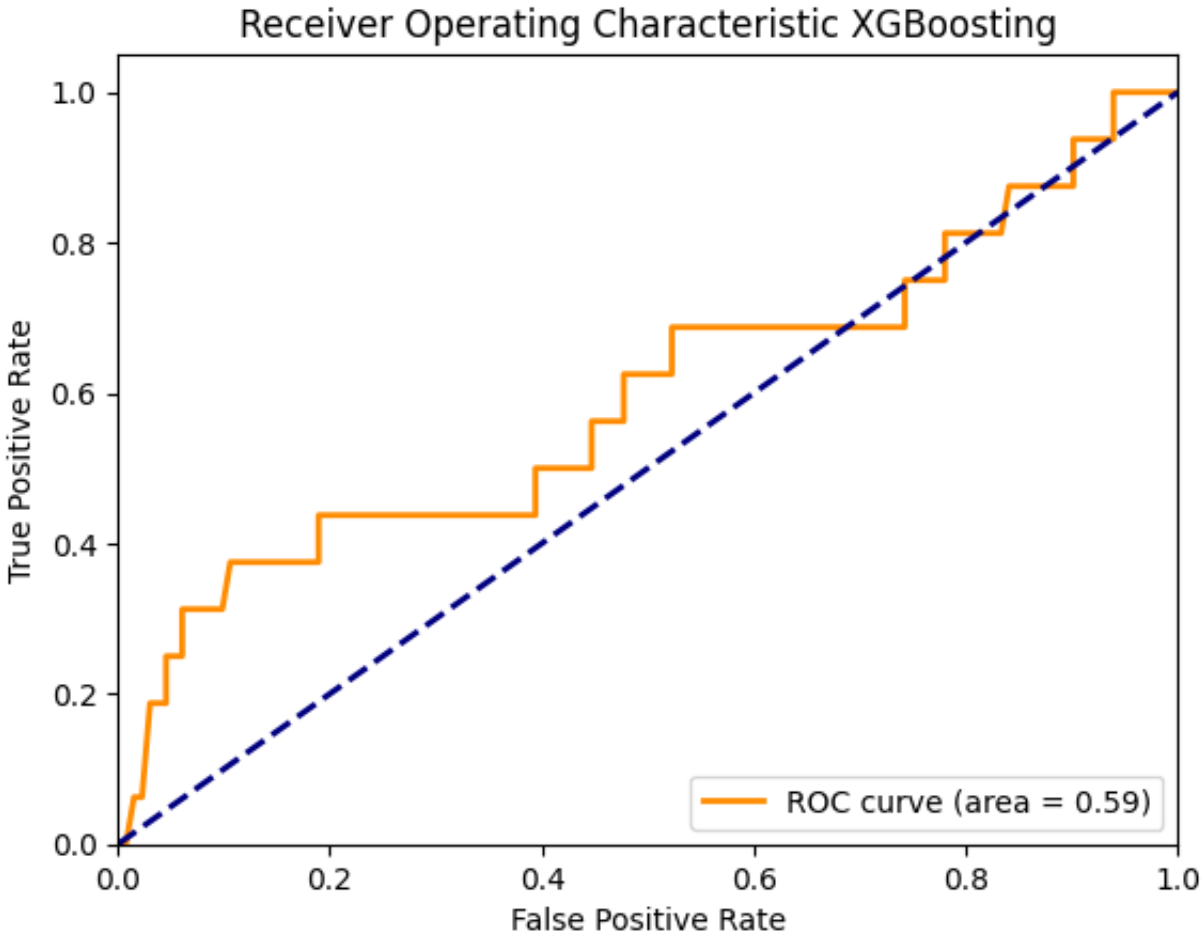
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
```

```
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Best parameters for XGBoost: {'colsample_bytree': 1, 'gamma': 2, 'learning_
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Training - Accuracy: 0.964, Sensitivity: 0.958, Specificity: 0.971, F1: 0.9
```

```
Test Metrics for manual threshold 0.5:
Accuracy: 0.804, Sensitivity: 0.375, Specificity: 0.856, F1: 0.293, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.47297297297297297, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.5472972972972973, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.5945945945945946, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6554054054054054, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.7162162162162162, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.8040540540540541, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8513513513513513, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for XGBoost
```







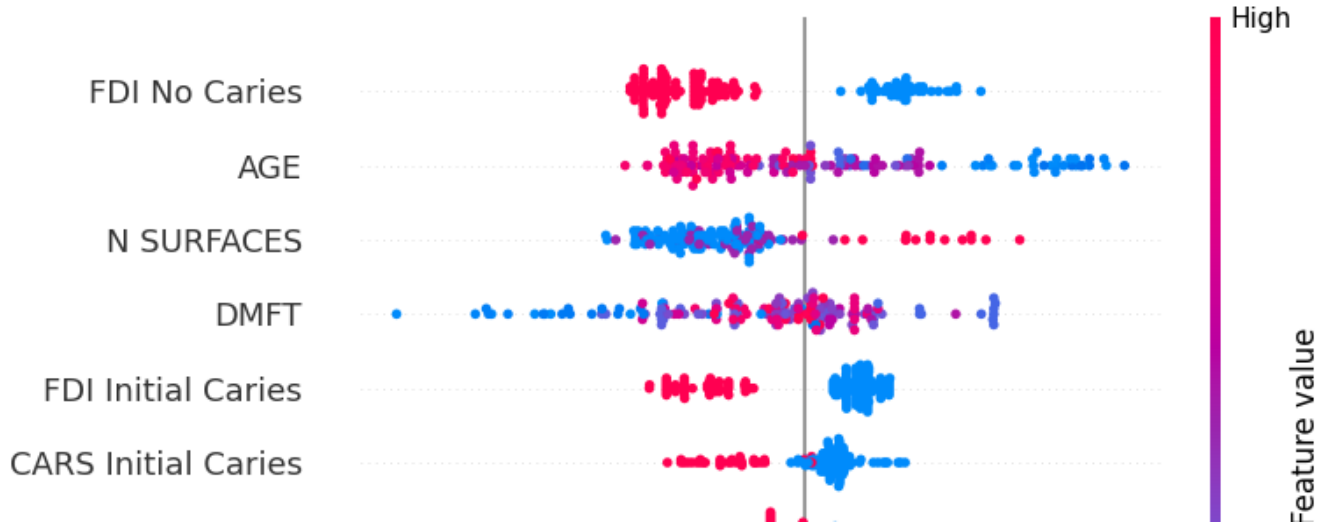

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.01515152 0.02272727 0.03030303 0.04545455
 0.04545455 0.06060606 0.06060606 0.09848485 0.10606061 0.16666667
 0.18181818 0.18939394 0.18939394 0.21969697 0.27272727 0.29545455
 0.31060606 0.35606061 0.37878788 0.39393939 0.39393939 0.4469697
 0.4469697  0.45454545 0.46969697 0.47727273 0.47727273 0.49242424
 0.52272727 0.52272727 0.74242424 0.74242424 0.78030303 0.78030303
 0.83333333 0.84090909 0.87878788 0.89393939 0.90151515 0.90151515
```

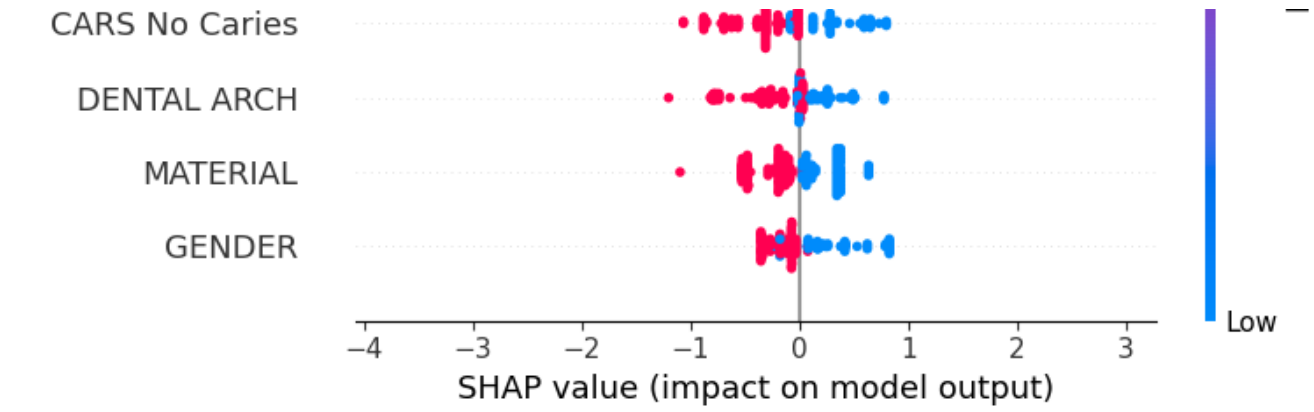
```
 0.93939394 0.93939394 0.9469697  0.96212121 1.        ]
TPR: [0.     0.     0.0625 0.0625 0.1875 0.1875 0.25   0.25   0.3125 0.3125
 0.375  0.375  0.375  0.375  0.4375 0.4375 0.4375 0.4375 0.4375 0.4375
 0.4375 0.4375 0.5    0.5    0.5625 0.5625 0.5625 0.5625 0.625  0.625
 0.625  0.6875 0.6875 0.75   0.75   0.8125 0.8125 0.875  0.875  0.875
 0.875  0.9375 0.9375 1.     1.     1.     1.    ]
ROC AUC: 0.594

Running evaluation with seed 45
Inside evaluate_xgboost function

Evaluating XGBoost with seed 45...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

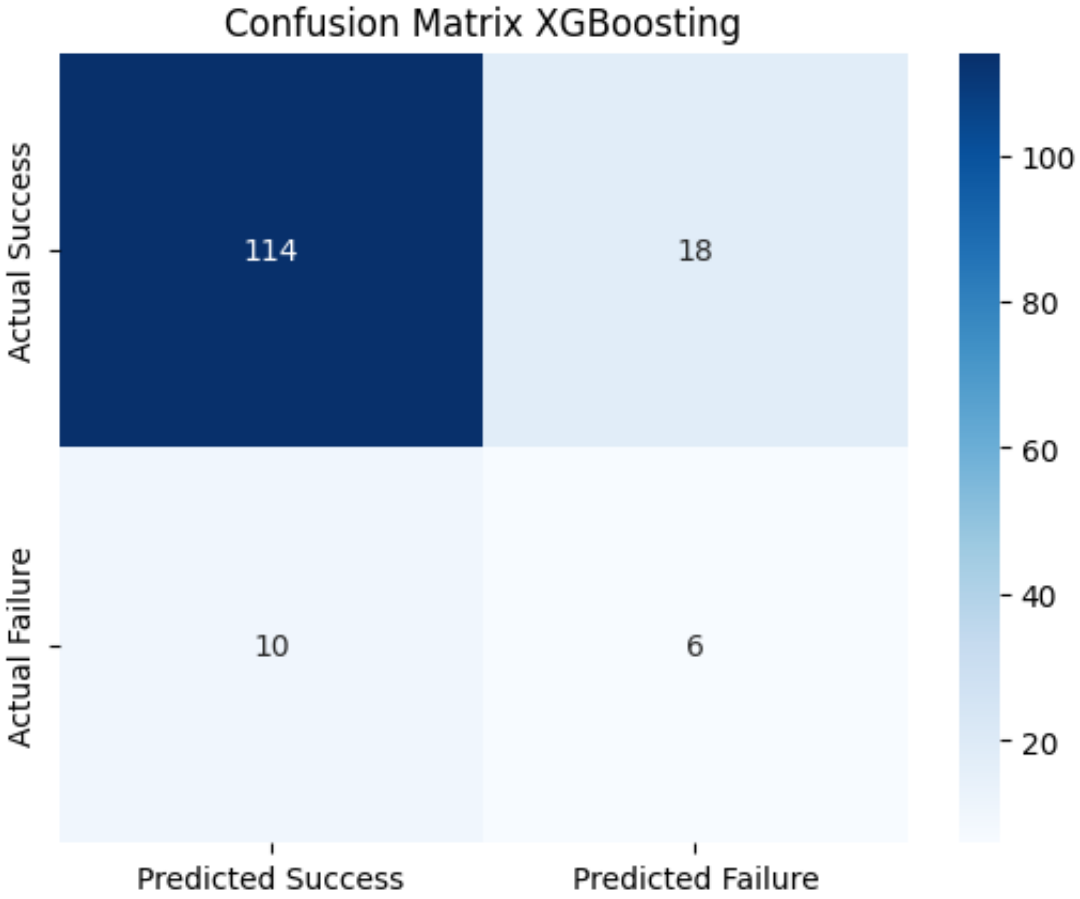
```
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
```

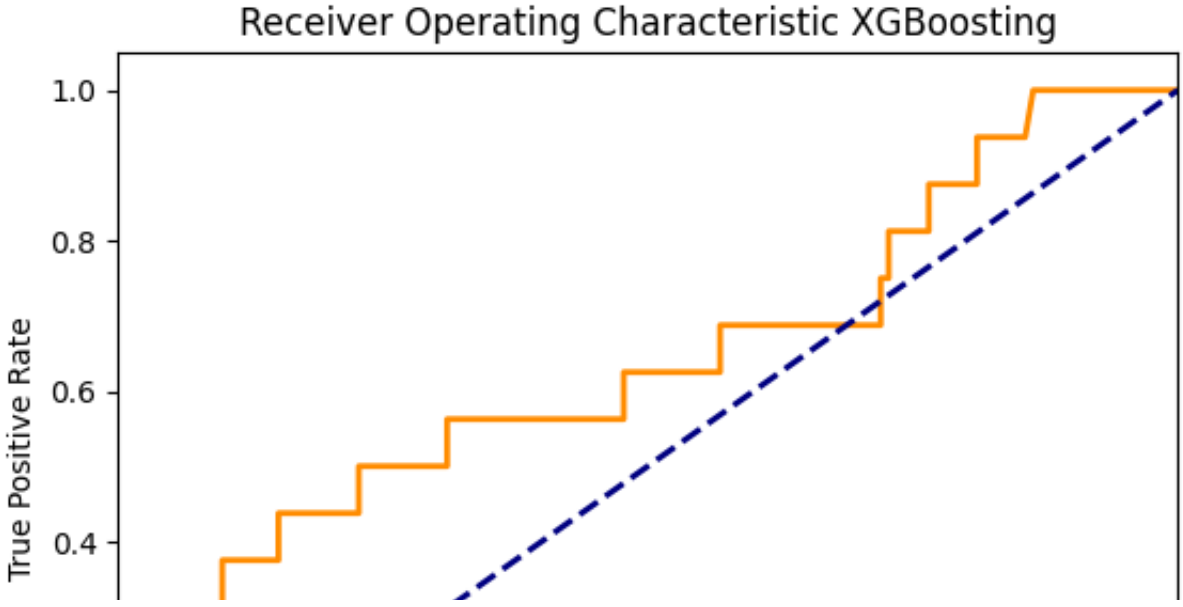
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

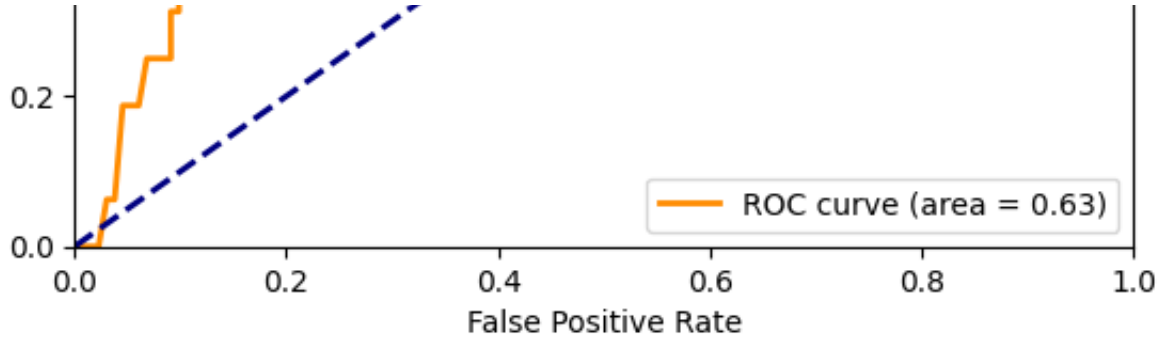
```
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Best parameters for XGBoost: {'colsample_bytree': 1, 'gamma': 2, 'learning_
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Training - Accuracy: 0.959, Sensitivity: 0.954, Specificity: 0.964, F1: 0.9

Test Metrics for manual threshold 0.5:
Accuracy: 0.811, Sensitivity: 0.375, Specificity: 0.864, F1: 0.300, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.4594594594594595, 'Sensitivity': 0
Threshold: 0.15, Metrics: {'Accuracy': 0.527027027027027, 'Sensitivity': 0.
Threshold: 0.20, Metrics: {'Accuracy': 0.581081081081081, 'Sensitivity': 0.
Threshold: 0.25, Metrics: {'Accuracy': 0.6351351351351351, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7094594594594594, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.75, 'Sensitivity': 0.4375, 'Specif
Threshold: 0.40, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.8040540540540541, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.8108108108108109, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8513513513513513, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for XGBoost
```




CARS No Caries
DENTAL ARCH
MATERIAL
GENDER
−4
−3
−2
−1
0
1
2
3
SHAP value (impact on model output)
Low


Confusion Matrix XGBoosting
Actual Success
Actual Failure
114
18
10
6
Predicted Success
Predicted Failure
100
80
60
40
20


Receiver Operating Characteristic XGBoosting
True Positive Rate
1.0
0.8
0.6
0.4



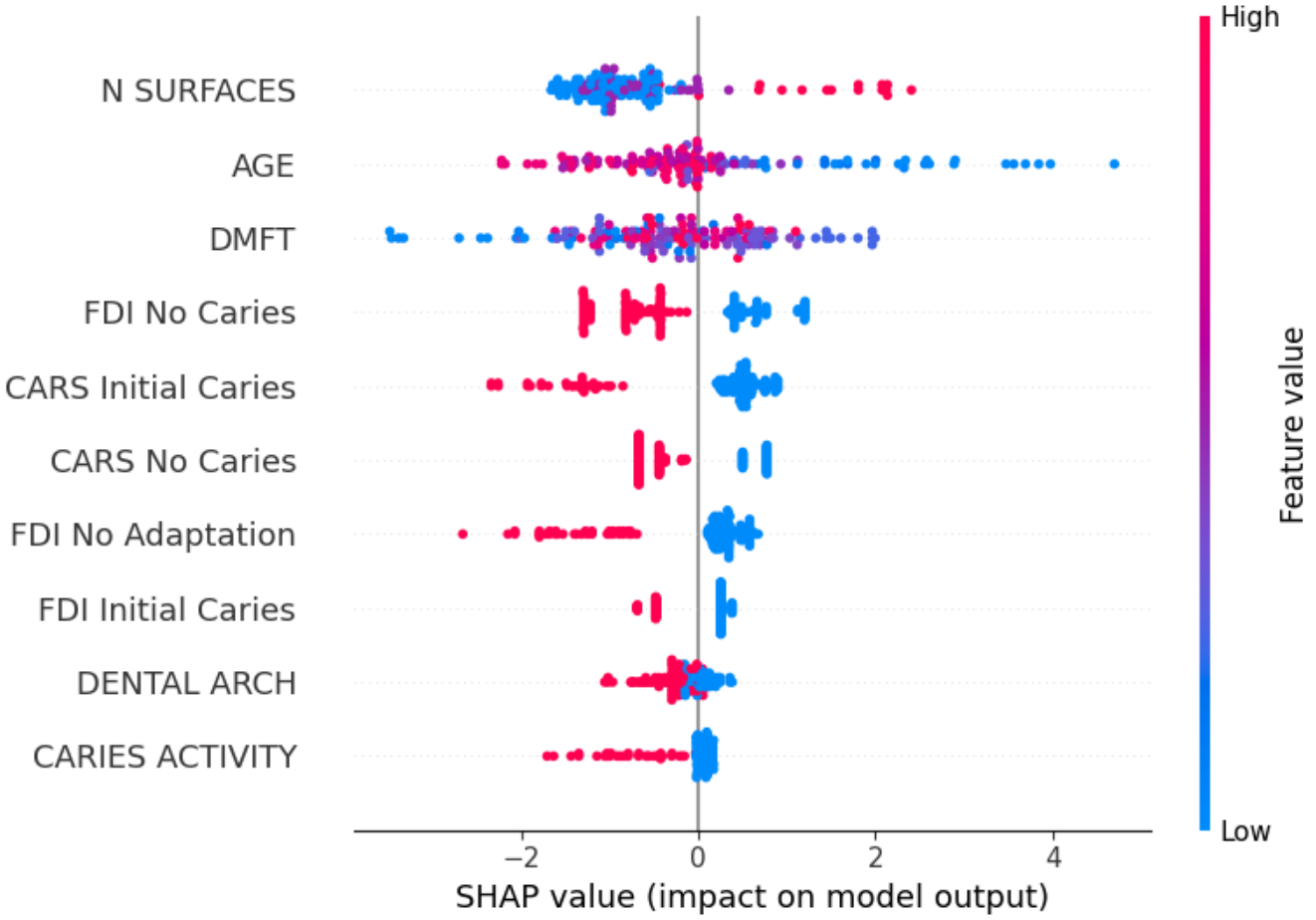

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.02272727 0.03030303 0.03787879 0.04545455
 0.06060606 0.06818182 0.09090909 0.09090909 0.09848485 0.09848485
 0.11363636 0.13636364 0.15151515 0.15151515 0.22727273 0.22727273
 0.23484848 0.25757576 0.26515152 0.28030303 0.29545455 0.31060606
 0.31060606 0.33333333 0.35606061 0.37121212 0.47727273 0.47727273
 0.56818182 0.56818182 0.59090909 0.60606061 0.71969697 0.71969697
 0.72727273 0.72727273 0.76515152 0.76515152 0.79545455 0.81060606
 0.81060606 0.85606061 0.86363636 0.90151515 0.91666667 1.        ]
TPR: [0.     0.     0.     0.0625 0.0625 0.1875 0.1875 0.25   0.25   0.3125
 0.3125 0.375  0.375  0.375  0.375  0.4375 0.4375 0.5    0.5    0.5
 0.5    0.5    0.5    0.5    0.5625 0.5625 0.5625 0.5625 0.5625 0.625
 0.625  0.6875 0.6875 0.6875 0.6875 0.75   0.75   0.8125 0.8125 0.875
 0.875  0.875  0.9375 0.9375 1.     1.     1.     1.    ]
ROC AUC: 0.626

Running evaluation with seed 46
Inside evaluate_xgboost function

Evaluating XGBoost with seed 46...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
```

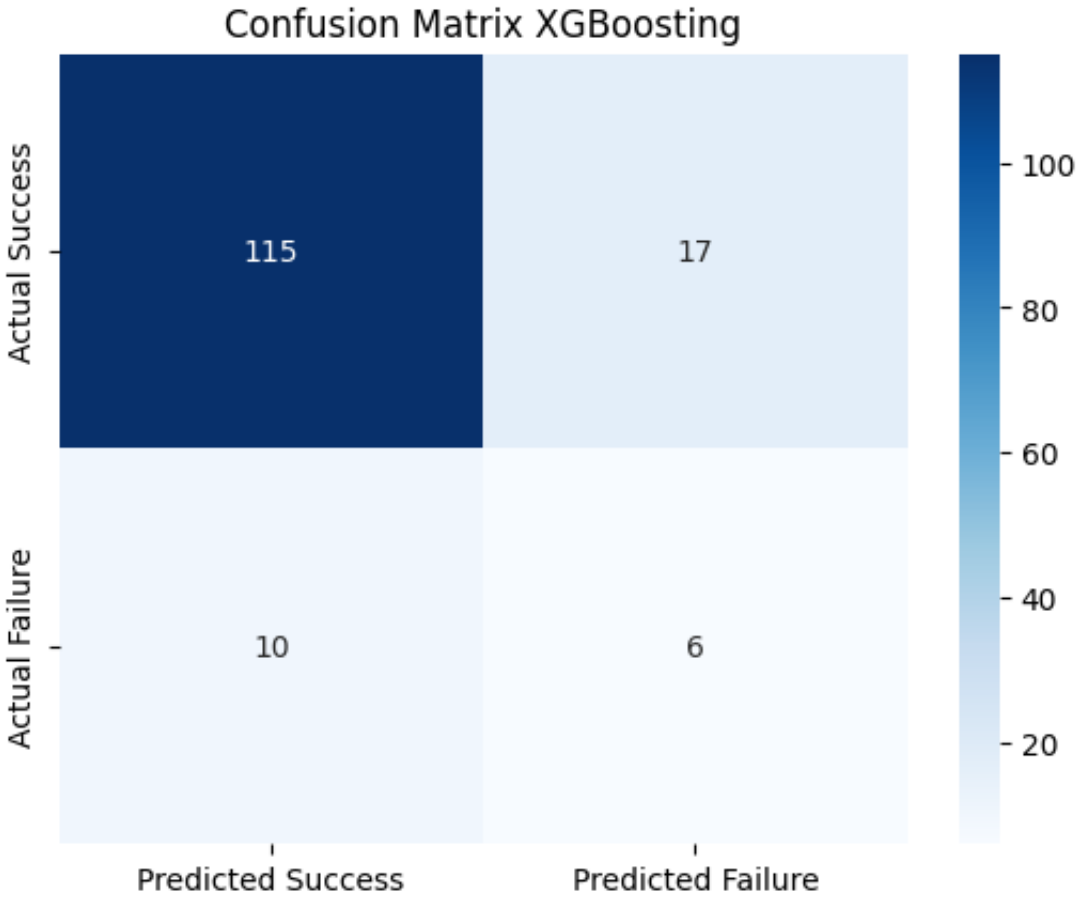
```
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

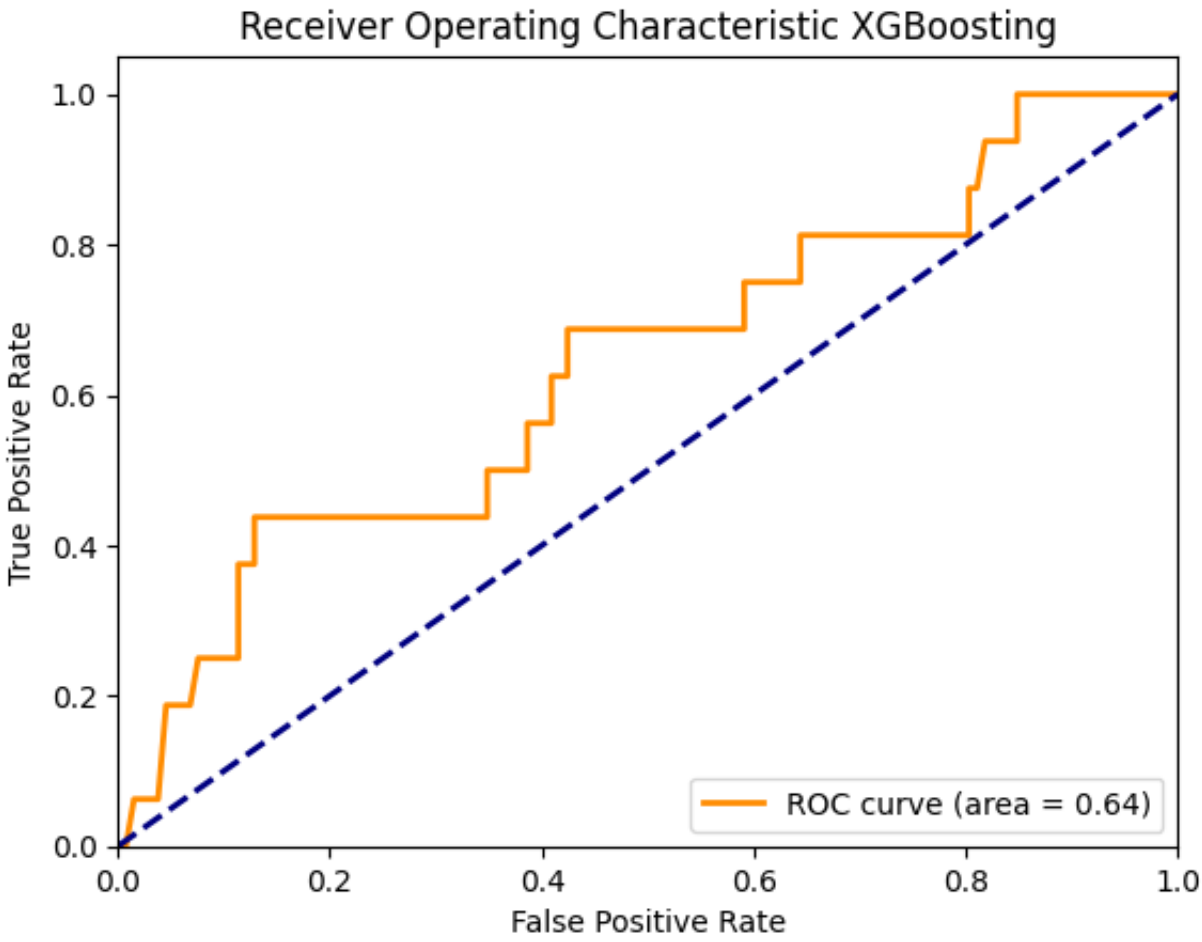
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
```

```
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Best parameters for XGBoost: {'colsample_bytree': 1, 'gamma': 2, 'learning_
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Training - Accuracy: 0.961, Sensitivity: 0.951, Specificity: 0.971, F1: 0.9

Test Metrics for manual threshold 0.5:
Accuracy: 0.818, Sensitivity: 0.375, Specificity: 0.871, F1: 0.308, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.43243243243243246, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.581081081081081, 'Sensitivity': 0.
Threshold: 0.20, Metrics: {'Accuracy': 0.6081081081081081, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.6486486486486487, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7094594594594594, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.8040540540540541, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8243243243243243, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
```

```
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for XGBoost
```







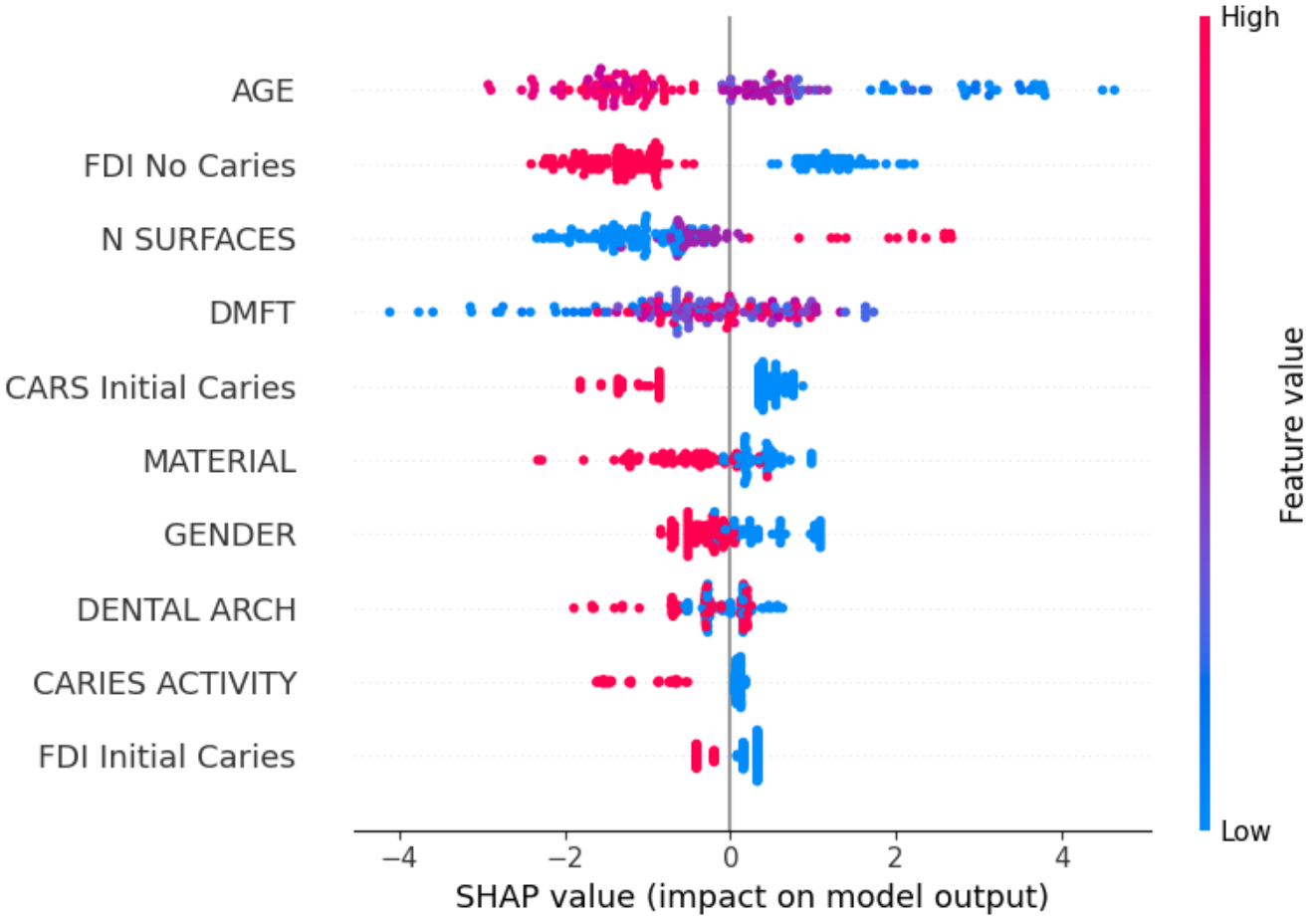

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.01515152 0.03787879 0.04545455 0.06818182
 0.07575758 0.11363636 0.11363636 0.12878788 0.12878788 0.15151515
 0.16666667 0.17424242 0.18939394 0.26515152 0.28787879 0.29545455
 0.31818182 0.34848485 0.34848485 0.38636364 0.38636364 0.40909091
 0.40909091 0.42424242 0.42424242 0.46212121 0.47727273 0.50757576
 0.52272727 0.59090909 0.59090909 0.62878788 0.64393939 0.64393939
 0.65151515 0.66666667 0.70454545 0.71969697 0.8030303  0.8030303
 0.81060606 0.81818182 0.84848485 0.84848485 0.89393939 0.90909091
 1.        ]
TPR: [0.     0.     0.0625 0.0625 0.1875 0.1875 0.25   0.25   0.375  0.375
 0.4375 0.4375 0.4375 0.4375 0.4375 0.4375 0.4375 0.4375 0.4375 0.4375
 0.5    0.5    0.5625 0.5625 0.625  0.625  0.6875 0.6875 0.6875 0.6875
 0.6875 0.6875 0.75   0.75   0.75   0.8125 0.8125 0.8125 0.8125 0.8125
 0.8125 0.875  0.875  0.9375 0.9375 1.     1.     1.     1.     ]
ROC AUC: 0.638

Running evaluation with seed 47
Inside evaluate_xgboost function

Evaluating XGBoost with seed 47...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

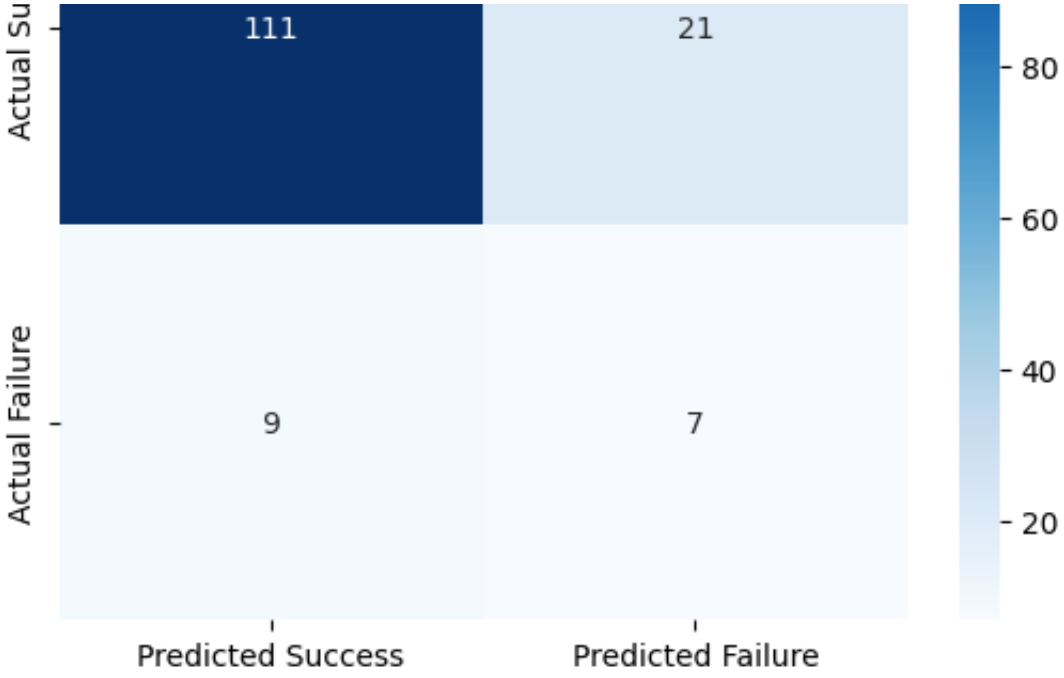
```
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
```

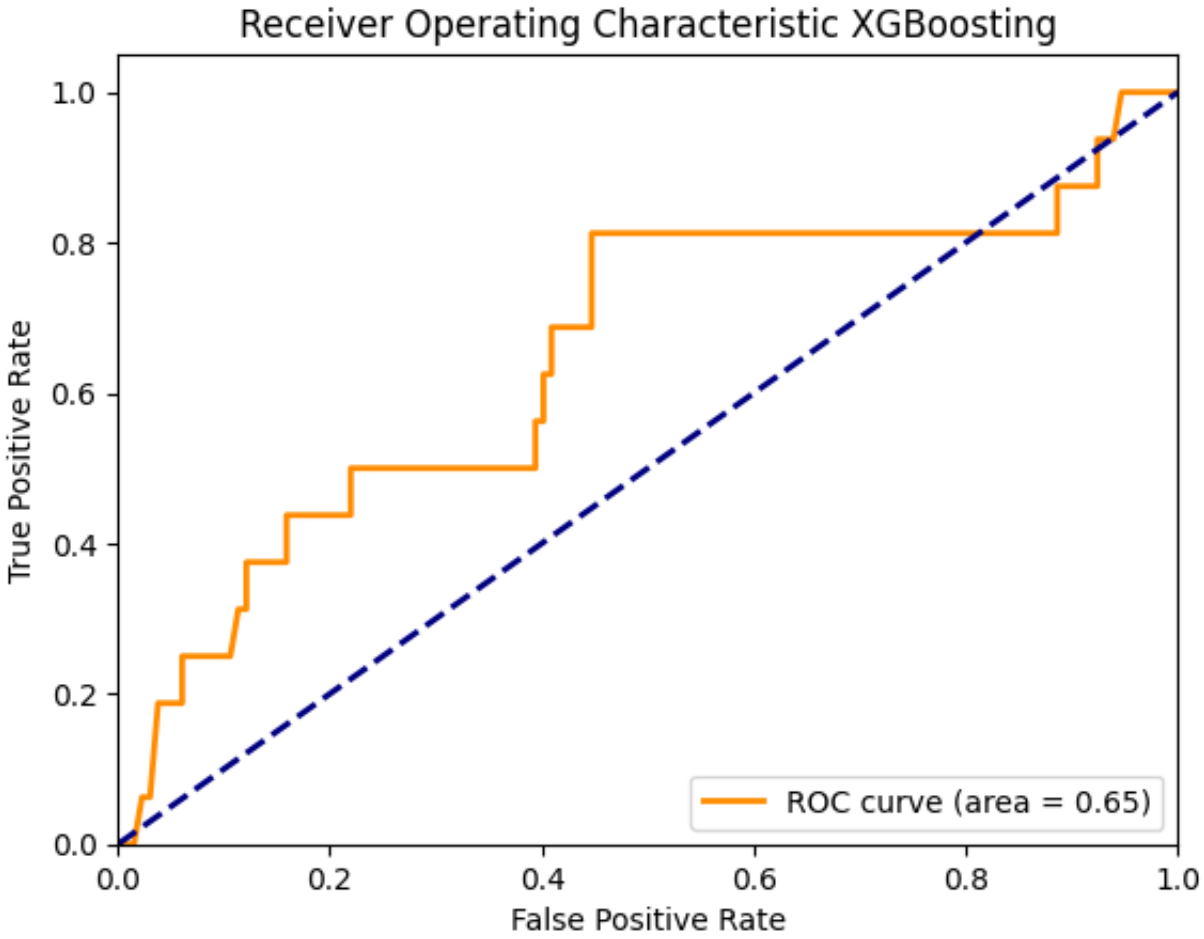
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

```
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Best parameters for XGBoost: {'colsample_bytree': 1, 'gamma': 2, 'learning_
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Training - Accuracy: 0.961, Sensitivity: 0.958, Specificity: 0.964, F1: 0.9

Test Metrics for manual threshold 0.5:
```

```
Accuracy: 0.797, Sensitivity: 0.438, Specificity: 0.841, F1: 0.318, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.38513513513513514, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.5202702702702703, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.5743243243243243, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7027027027027027, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.75, 'Sensitivity': 0.5, 'Specifici
Threshold: 0.40, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7837837837837838, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7972972972972973, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for XGBoost
```







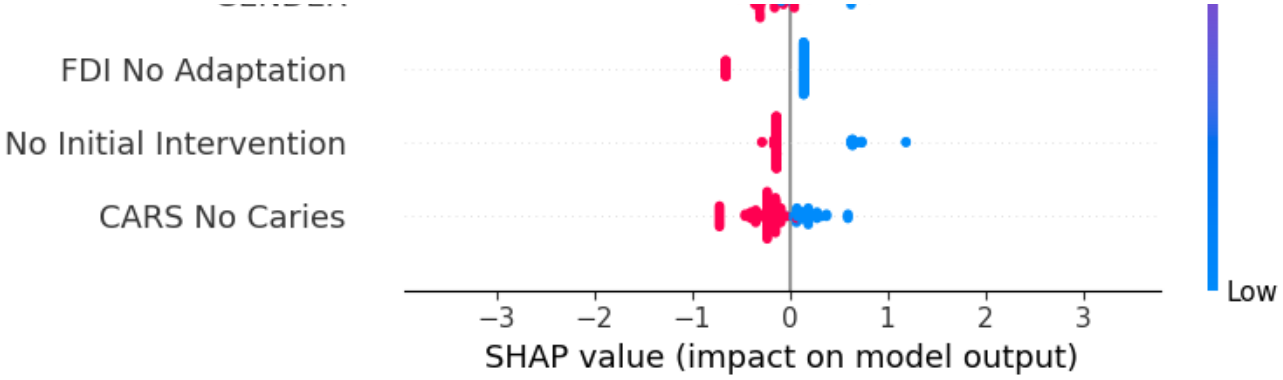

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.01515152 0.02272727 0.03030303 0.03787879
 0.06060606 0.06060606 0.10606061 0.11363636 0.12121212 0.12121212
 0.14393939 0.15909091 0.15909091 0.21969697 0.21969697 0.25757576
 0.27272727 0.3030303  0.31818182 0.34848485 0.39393939 0.39393939
 0.40151515 0.40151515 0.40909091 0.40909091 0.41666667 0.43181818
 0.4469697  0.4469697  0.45454545 0.46969697 0.5530303  0.56818182
 0.88636364 0.88636364 0.92424242 0.92424242 0.93939394 0.9469697
```

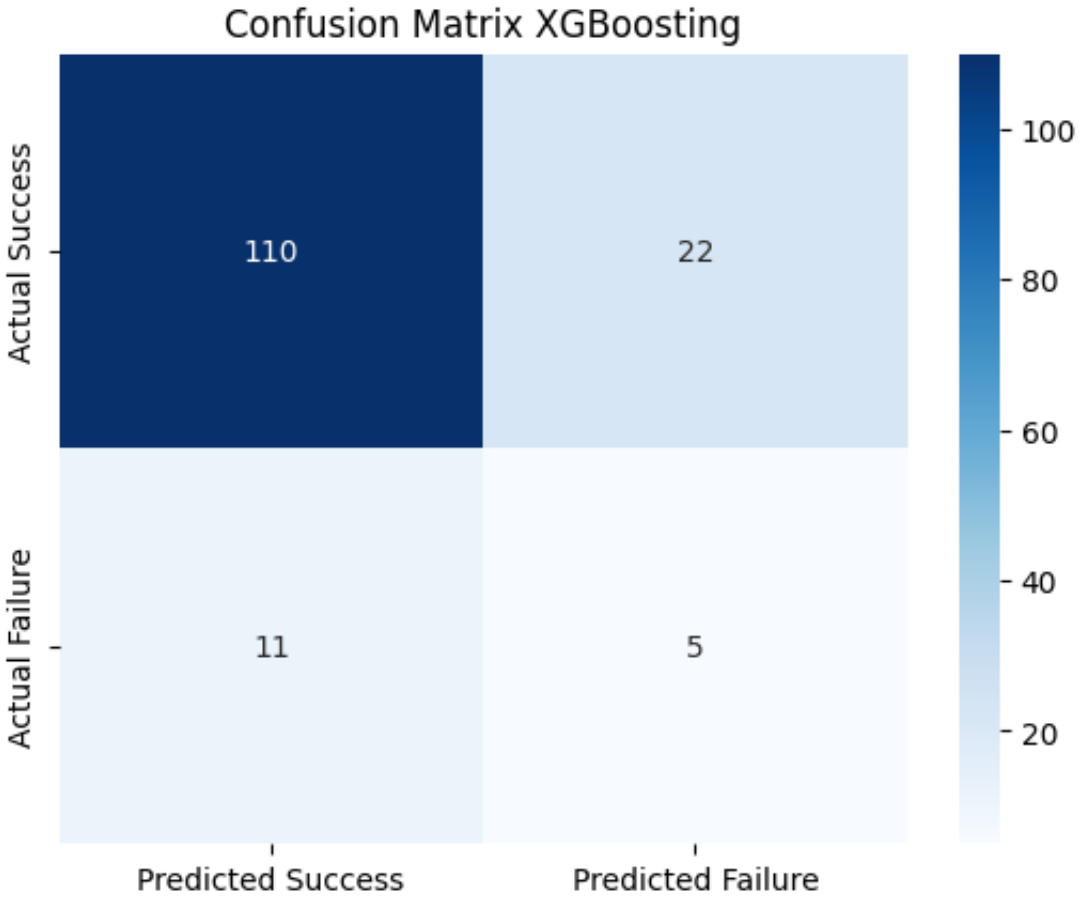
```
 0.95454545 0.96969697 1.        ]
TPR: [0.     0.     0.     0.0625 0.0625 0.1875 0.1875 0.25   0.25   0.3125
 0.3125 0.375  0.375  0.375  0.4375 0.4375 0.5    0.5    0.5    0.5
 0.5    0.5    0.5    0.5625 0.5625 0.625  0.625  0.6875 0.6875 0.6875
 0.6875 0.8125 0.8125 0.8125 0.8125 0.8125 0.8125 0.875  0.875  0.9375
 0.9375 1.     1.     1.     1.    ]
ROC AUC: 0.649

Running evaluation with seed 48
Inside evaluate_xgboost function

Evaluating XGBoost with seed 48...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

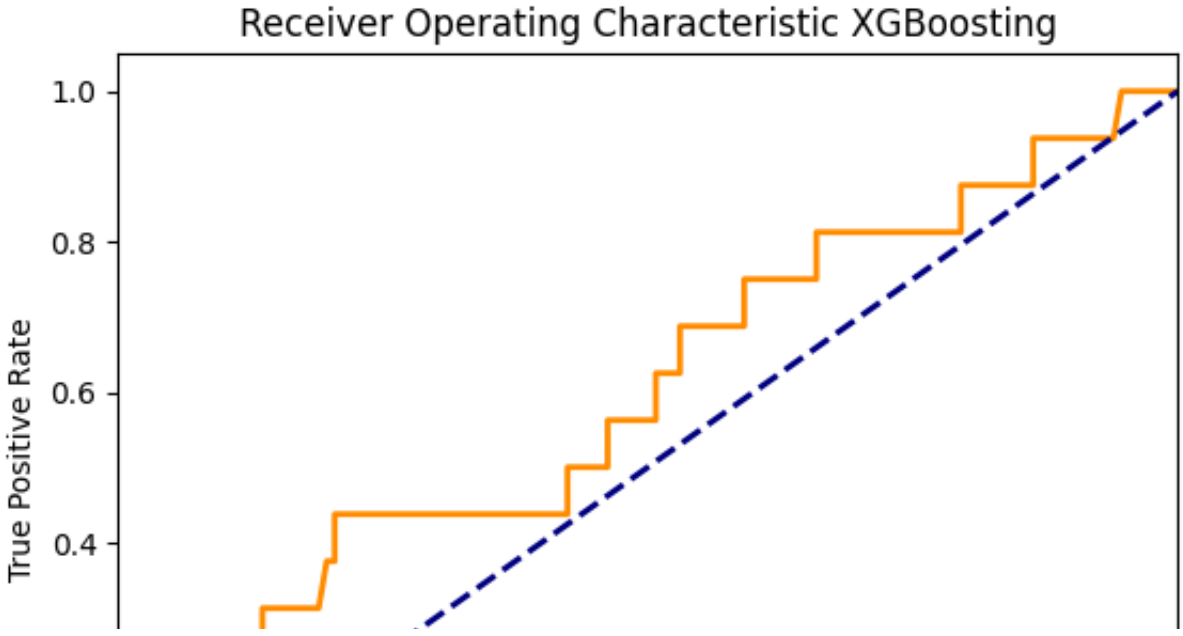
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
```

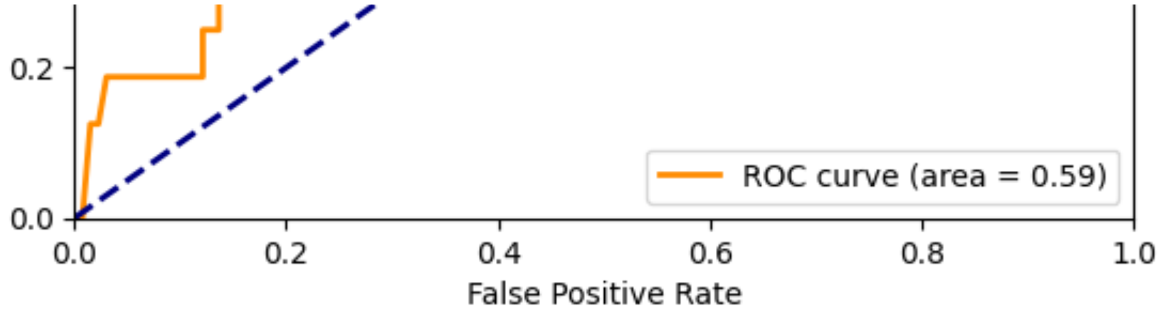
```
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Best parameters for XGBoost: {'colsample_bytree': 1, 'gamma': 2, 'learning_
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
```

```
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Training - Accuracy: 0.953, Sensitivity: 0.948, Specificity: 0.958, F1: 0.9

Test Metrics for manual threshold 0.5:
Accuracy: 0.777, Sensitivity: 0.312, Specificity: 0.833, F1: 0.233, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.47297297297297297, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.5337837837837838, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.6216216216216216, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.6486486486486487, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7027027027027027, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.722972972972973, 'Sensitivity': 0.
Threshold: 0.40, Metrics: {'Accuracy': 0.75, 'Sensitivity': 0.4375, 'Specif
Threshold: 0.45, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.777027027027027, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.8040540540540541, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for XGBoost
```


FDI No Adaptation
No Initial Intervention
CARS No Caries
−3
−2
−1
0
1
2
3
SHAP value (impact on model output)
Low


Confusion Matrix XGBoosting
Actual Success
Actual Failure
110
22
11
5
Predicted Success
Predicted Failure
100
80
60
40
20


Receiver Operating Characteristic XGBoosting
True Positive Rate
1.0
0.8
0.6
0.4



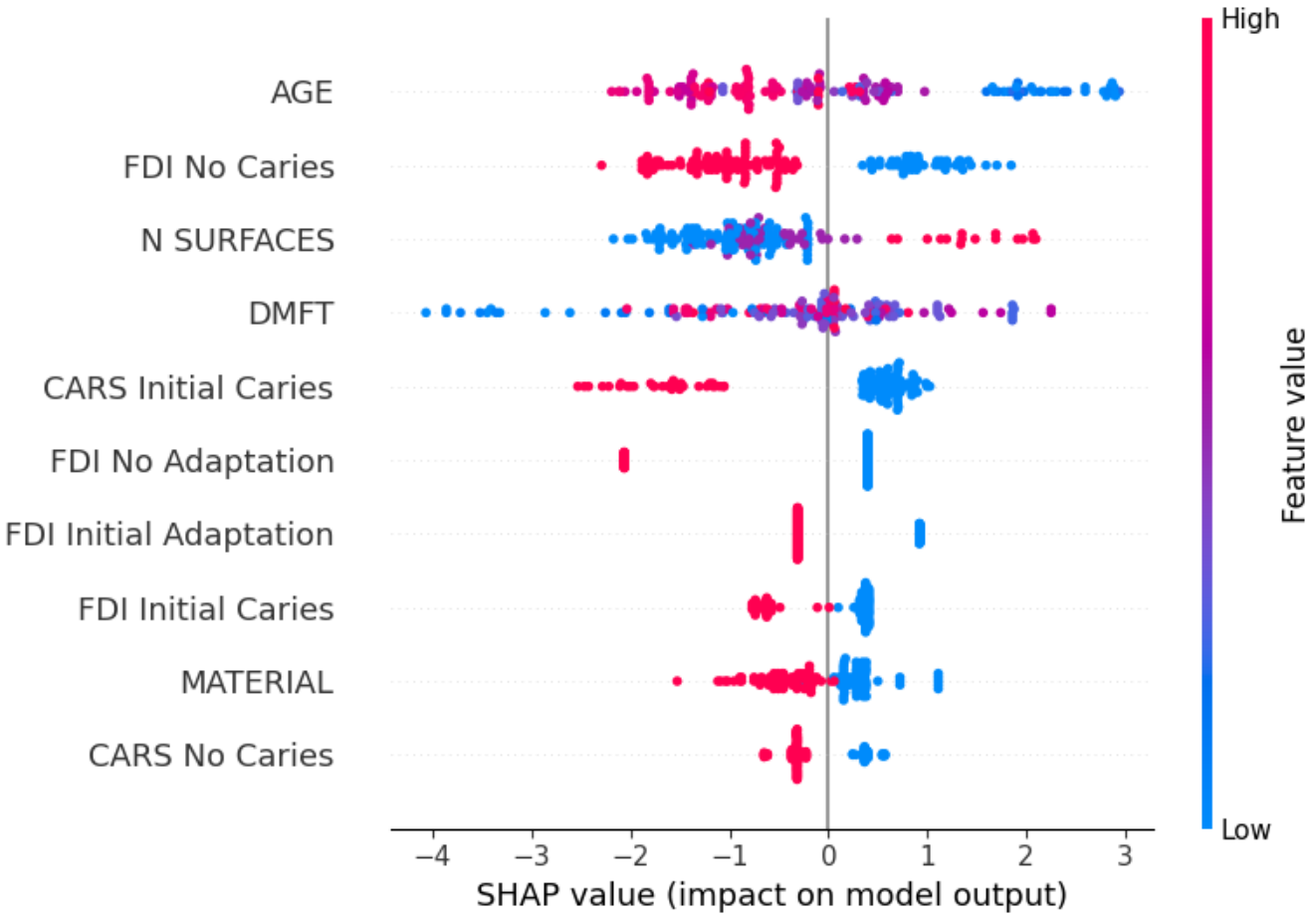

```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.01515152 0.02272727 0.03030303 0.07575758
 0.09090909 0.12121212 0.12121212 0.13636364 0.13636364 0.16666667
 0.18181818 0.18939394 0.1969697  0.20454545 0.20454545 0.26515152
 0.28787879 0.35606061 0.37121212 0.37878788 0.40151515 0.41666667
 0.42424242 0.42424242 0.43939394 0.46212121 0.46212121 0.50757576
 0.50757576 0.53030303 0.53030303 0.59090909 0.59090909 0.65909091
 0.65909091 0.67424242 0.68939394 0.74242424 0.75757576 0.79545455
 0.79545455 0.86363636 0.86363636 0.88636364 0.90151515 0.93939394
 0.9469697  1.        ]
TPR: [0.     0.     0.125  0.125  0.1875 0.1875 0.1875 0.1875 0.25   0.25
 0.3125 0.3125 0.3125 0.3125 0.375  0.375  0.4375 0.4375 0.4375 0.4375
 0.4375 0.4375 0.4375 0.4375 0.4375 0.5    0.5    0.5    0.5625 0.5625
 0.625  0.625  0.6875 0.6875 0.75   0.75   0.8125 0.8125 0.8125 0.8125
 0.8125 0.8125 0.875  0.875  0.9375 0.9375 0.9375 0.9375 1.     1.    ]
ROC AUC: 0.595

Running evaluation with seed 49
Inside evaluate_xgboost function

Evaluating XGBoost with seed 49...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
```

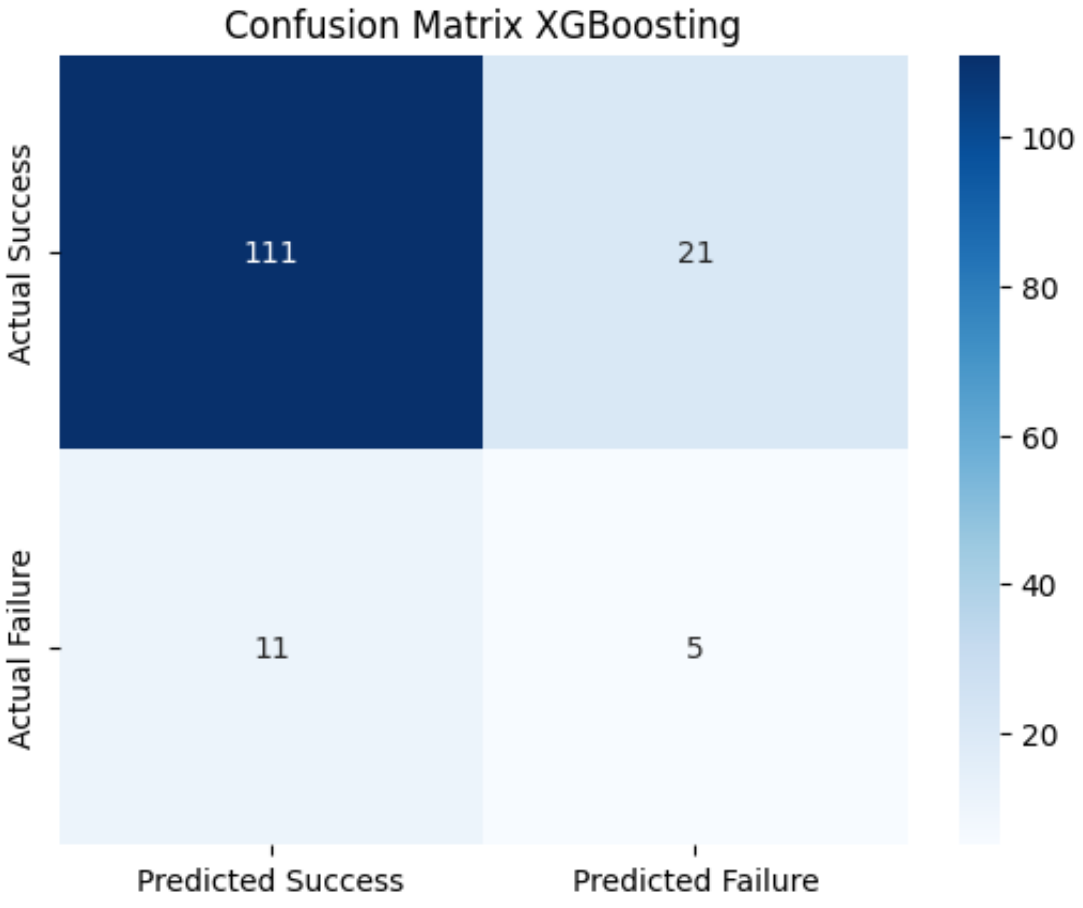
```
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.
```

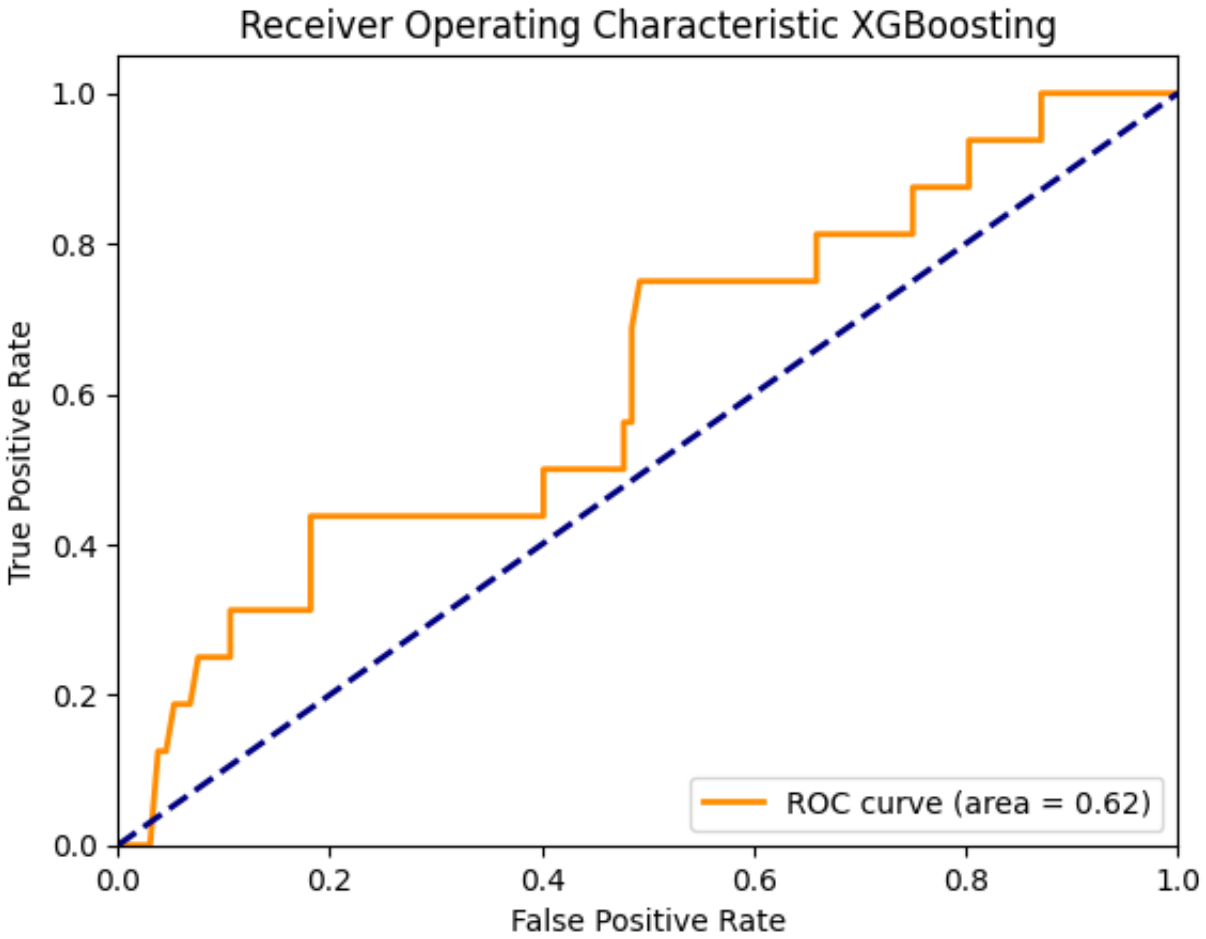
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
```

```
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Best parameters for XGBoost: {'colsample_bytree': 1, 'gamma': 2, 'learning_
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Training - Accuracy: 0.963, Sensitivity: 0.961, Specificity: 0.964, F1: 0.9

Test Metrics for manual threshold 0.5:
Accuracy: 0.784, Sensitivity: 0.312, Specificity: 0.841, F1: 0.238, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.41216216216216217, 'Sensitivity':
Threshold: 0.15, Metrics: {'Accuracy': 0.527027027027027, 'Sensitivity': 0.
Threshold: 0.20, Metrics: {'Accuracy': 0.5743243243243243, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.6486486486486487, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6621621621621622, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7837837837837838, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8513513513513513, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8513513513513513, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
```

```
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for XGBoost
```








```
ROC Curve Metrics:
FPR: [0.         0.00757576 0.03030303 0.03787879 0.04545455 0.0530303
 0.06818182 0.07575758 0.09090909 0.10606061 0.10606061 0.12878788
 0.14393939 0.18181818 0.18181818 0.24242424 0.26515152 0.28030303
 0.31060606 0.34090909 0.35606061 0.37878788 0.40151515 0.40151515
 0.47727273 0.47727273 0.48484848 0.48484848 0.49242424 0.63636364
 0.65151515 0.65909091 0.65909091 0.75       0.75       0.8030303
 0.8030303  0.81818182 0.87121212 0.87121212 0.91666667 0.93181818
 1.        ]
TPR: [0.     0.     0.     0.125  0.125  0.1875 0.1875 0.25   0.25   0.25
 0.3125 0.3125 0.3125 0.3125 0.4375 0.4375 0.4375 0.4375 0.4375 0.4375
 0.4375 0.4375 0.4375 0.5    0.5    0.5625 0.5625 0.6875 0.75   0.75
 0.75   0.75   0.8125 0.8125 0.875  0.875  0.9375 0.9375 0.9375 1.
 1.     1.     1.    ]
ROC AUC: 0.620

Aggregated Test Set Metrics Across Seeds:
   accuracy  sensitivity  specificity        f1   roc_auc
0  0.797297       0.3750     0.848485  0.285714  0.664062
1  0.804054       0.3125     0.863636  0.256410  0.691051
2  0.750000       0.3125     0.803030  0.212766  0.672112
3  0.837838       0.3125     0.901515  0.294118  0.672585
4  0.804054       0.3750     0.856061  0.292683  0.593513
5  0.810811       0.3750     0.863636  0.300000  0.626184
6  0.817568       0.3750     0.871212  0.307692  0.638021
7  0.797297       0.4375     0.840909  0.318182  0.649384
8  0.777027       0.3125     0.833333  0.232558  0.594934
```

```
8  0.777627       0.3125     0.833333  0.232956  0.594954
9  0.783784       0.3125     0.840909  0.238095  0.620028

Summary of Test Set Metrics (Mean, Standard Error, 95% Confidence Interval)
Accuracy: Mean = 0.798, SE = 0.008, 95% CI = [0.781, 0.815]
Sensitivity: Mean = 0.350, SE = 0.014, 95% CI = [0.319, 0.381]
Specificity: Mean = 0.852, SE = 0.008, 95% CI = [0.834, 0.871]
F1: Mean = 0.274, SE = 0.011, 95% CI = [0.248, 0.300]
Roc_auc: Mean = 0.642, SE = 0.011, 95% CI = [0.618, 0.666]
```