```
!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
```

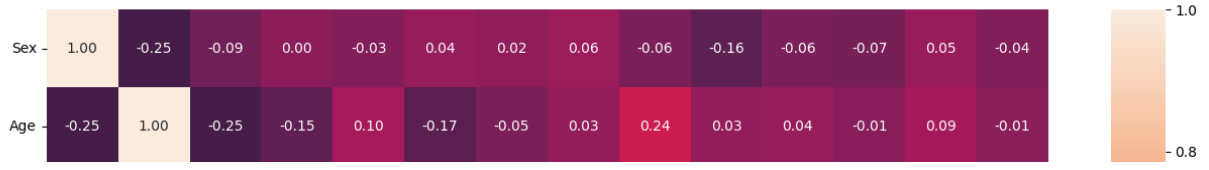
```
Requirement already satisfied: scikit-learn==1.2.2 in /usr/local/lib/python
Requirement already satisfied: numpy==1.25.2 in /usr/local/lib/python3.11/d
Requirement already satisfied: pandas==2.0.3 in /usr/local/lib/python3.11/d
Requirement already satisfied: scipy==1.11.2 in /usr/local/lib/python3.11/d
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
```

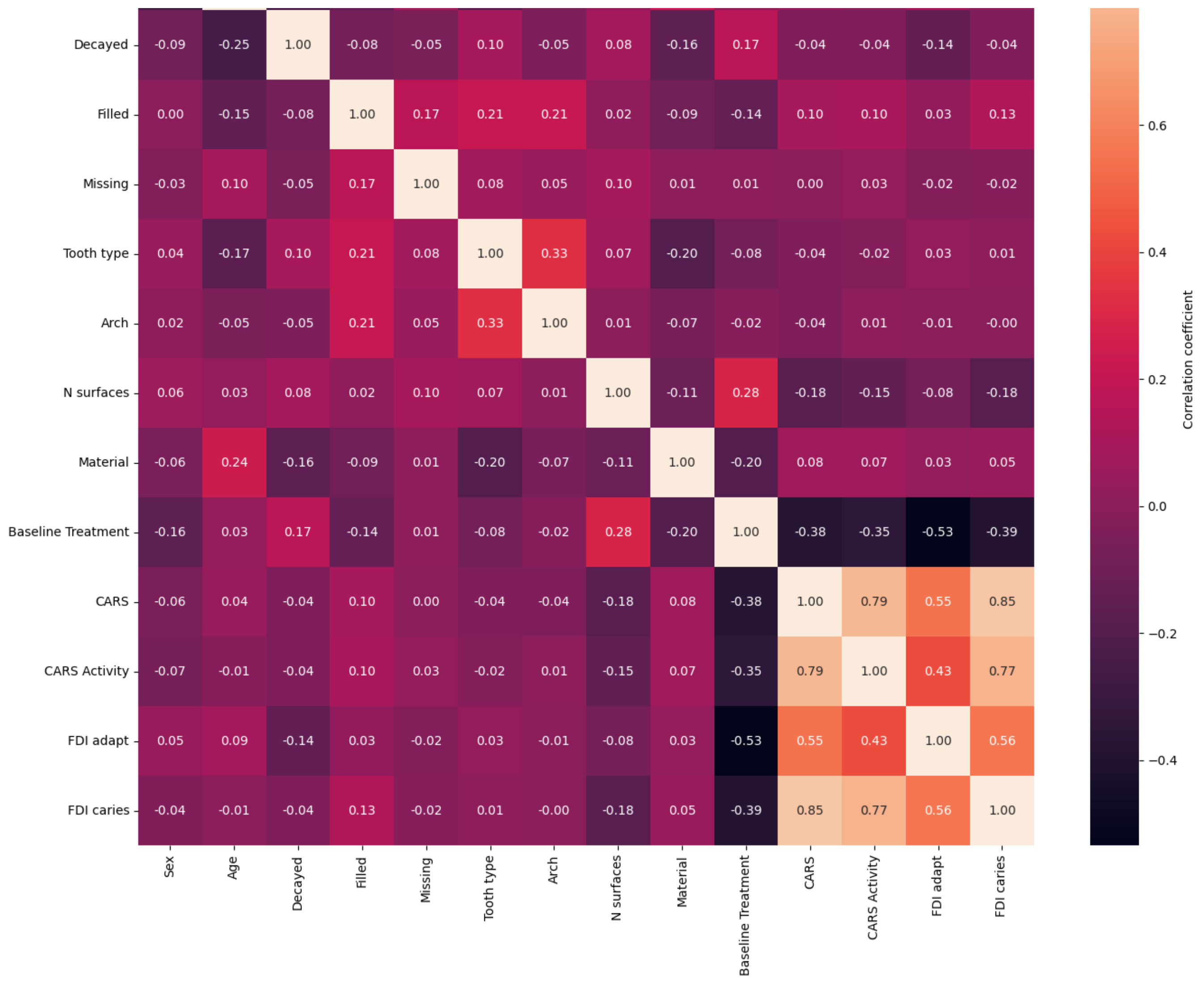
```
pip freeze > new_env_requirements.txt
```

```
!python --version
```

```
Python 3.10.12
```

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

# Load the data from an Excel file
data = pd.read_excel('2024_corrected_subst_CARDEC_3_ML_Vitor.xlsx')

# Split the dataset into training and testing sets based on a unique identifier
# This ensures that data related to the same 'IDpac' is not split across both tr
unique_n_part = data['IDpac'].unique()
train_n_part, test_n_part = train_test_split(unique_n_part, test_size=0.2, rando

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

# Separate features and target variable for training set
# 'drop' removes specified columns from the dataset, in this case removing targe
X_train = train_data.drop(['Failure', 'IDrest', 'IDpac'], axis=1)
y_train = train_data['Failure']  # Isolate the target variable for the training

# Separate features and target variable for testing set following the same proce
X_test = test_data.drop(['Failure', 'IDrest', 'IDpac'], axis=1)
y_test = test_data['Failure']  # Isolate the target variable for the testing set


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 variat
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.
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 lists for each type of variable in the dataset: numeric, binary, and c
numeric_vars = ['Age', 'Decayed', 'Filled', 'Missing']
binary_vars = ['Sex', 'Tooth type', 'Arch', 'Failure', 'CARS Activity']
categorical_vars = ['N surfaces', 'Material', 'Baseline Treatment', 'CARS', 'FD

def descriptive_statistics(train_data, test_data):
    # Print a heading for the descriptive statistics of numeric variables.
    print("Descriptive Statistics for Numeric Variables:")
    # Display descriptive statistics (like count, mean, std, min, max, etc.) fo
    print("\nTraining Set:")
    print(train_data[numeric_vars].describe())
    # Repeat the process for the test set.
    print("\nTest Set:")
    print(test_data[numeric_vars].describe())

    # Initialize an empty dictionary to store statistics for binary and ordinal
    stats = {}
    # Loop through each variable in the binary and ordinal lists to calculate t
    for var in binary_vars + categorical_vars:
        stats[var] = {
            "Training Set": {
                "Count": train_data[var].value_counts().to_dict(),  # Count occ
                "Percentage": (train_data[var].value_counts(normalize=True) * 1
            },
            "Test Set": {
                "Count": test_data[var].value_counts().to_dict(),  # Count occu
                "Percentage": (test_data[var].value_counts(normalize=True) * 10
            }
        }

    # Loop through the stats dictionary to print the statistics for each catego
    for var, data in stats.items():
        print(f"\n{var} Statistics:")  # Print the variable name.
        for dataset, values in data.items():
            print(f"\n{dataset}:")  # Print which dataset (training or test) th
            for metric, metric_values in values.items():
                print(f"{metric}: {metric_values}")  # Print the count and perc

# Call the function with the training and test datasets as arguments to display
descriptive_statistics(train_data, test_data)
```

```
N surfaces Statistics:

Training Set:
Count: {1: 207, 2: 139, 3: 72, 4: 60, 5: 29}
Percentage: {1: 40.828402366863905, 2: 27.416173570019726, 3: 14.2011834319
```

```
Test Set:
Count: {1: 56, 2: 27, 3: 20, 4: 14, 5: 13}
Percentage: {1: 43.07692307692308, 2: 20.76923076923077, 3: 15.384615384615

Material Statistics:

Training Set:
Count: {1: 304, 0: 189, 2: 14}
Percentage: {1: 59.96055226824457, 0: 37.278106508875744, 2: 2.761341222879

Test Set:
Count: {1: 74, 0: 51, 2: 5}
Percentage: {1: 56.92307692307692, 0: 39.23076923076923, 2: 3.8461538461538

Baseline Treatment Statistics:

Training Set:
Count: {0: 292, 1: 167, 2: 48}
Percentage: {0: 57.59368836291914, 1: 32.938856015779095, 2: 9.467455621301

Test Set:
Count: {0: 75, 1: 34, 2: 21}
Percentage: {0: 57.692307692307686, 1: 26.153846153846157, 2: 16.1538461538

CARS Statistics:

Training Set:
Count: {0: 399, 2: 56, 1: 46, 3: 6}
Percentage: {0: 78.69822485207101, 2: 11.045364891518737, 1: 9.072978303747

Test Set:
Count: {0: 99, 2: 20, 1: 11}
Percentage: {0: 76.15384615384615, 2: 15.384615384615385, 1: 8.461538461538

FDI adapt Statistics:

Training Set:
Count: {0: 333, 1: 155, 2: 19}
Percentage: {0: 65.68047337278107, 1: 30.57199211045365, 2: 3.7475345167652

Test Set:
Count: {0: 82, 1: 43, 2: 5}
Percentage: {0: 63.07692307692307, 1: 33.07692307692307, 2: 3.8461538461538

FDI caries Statistics:

Training Set:
Count: {0: 401, 1: 98, 2: 8}
Percentage: {0: 79.09270216962526, 1: 19.32938856015779, 2: 1.5779092702169

Test Set:
Count: {0: 97, 1: 30, 2: 3}
Percentage: {0: 74.61538461538461, 1: 23.076923076923077, 2: 2.307692307692
```

```
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Convert specified categorical variables in the training data to 'category' dt
X_train['Material'] = X_train['Material'].astype('category')
X_train['Baseline Treatment'] = X_train['Baseline Treatment'].astype('category'
X_train['CARS'] = X_train['CARS'].astype('category')
X_train['FDI adapt'] = X_train['FDI adapt'].astype('category')
X_train['FDI caries'] = X_train['FDI caries'].astype('category')

# Apply one-hot encoding to the specified categorical columns in the training c
# 'prefix' argument specifies the prefix to add to the columns resulting from t
one_hot_train = pd.get_dummies(X_train[['Material', 'Baseline Treatment', 'CARS
                               prefix=['Material', 'Baseline_Treatment', 'CARS'

# Concatenate the original training data (minus the now-encoded variables) with
X_train = pd.concat([X_train.drop(['Material', 'Baseline Treatment', 'CARS', 'F

# 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['Material'] = X_test['Material'].astype('category')
X_test['Baseline Treatment'] = X_test['Baseline Treatment'].astype('category')
X_test['CARS'] = X_test['CARS'].astype('category')
X_test['FDI adapt'] = X_test['FDI adapt'].astype('category')
X_test['FDI caries'] = X_test['FDI caries'].astype('category')

one_hot_test = pd.get_dummies(X_test[['Material', 'Baseline Treatment', 'CARS',
                              prefix=['Material', 'Baseline_Treatment', 'CARS',

# 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 w

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

# Define a dictionary to rename the one-hot encoded columns for clarity, making
column_renaming = {'Material_0': 'Composite',
    'Material_1': 'Glass Ionomer Cement',
    'Material_2': 'Amalgam',
    'Baseline_Treatment_0': 'No initial intervention',
    'Baseline_Treatment_1': 'Repaired baseline',
    'Baseline_Treatment_2': 'Replaced baseline',
    'CARS_0': 'CARS No caries',
    'CARS_1': 'CARS Initial',
    'CARS_2': 'CARS Moderate/advanced',
    'FDI_adapt_0': 'FDI No adaptation',
    'FDI_adapt_1': 'FDI Initial adaptation',
    'FDI_adapt_2': 'FDI Moderate/advanced adaptation',
    'FDI_caries_0': 'FDI No caries',
    'FDI_caries_1': 'FDI Initial caries',
    'FDI_caries_2': 'FDI Moderate/advanced caries'}

# 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', 'Decayed', 'Filled', 'Missing']] = scaler.fit_transform(
X_test.loc[:, ['Age', 'Decayed', 'Filled', 'Missing']] = scaler.transform(X_tes

# Define which columns are considered categorical, excluding numerical columns
categorical_features = list(range(len(X_train.columns)))
for col in ['Age', 'Decayed', 'Filled', 'Missing']:  # Assuming these are your
    categorical_features.remove(X_train.columns.get_loc(col))
```

```
<ipython-input-6-1e795d60a953>:64: 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 lists categorizing the types of variables in the dataset.
numeric_vars = ['Age', 'Decayed', 'Filled', 'Missing']  # Numeric variables
original_categorical_vars = ['Sex', 'Tooth type', 'Arch', 'N surfaces', 'Failur
# One-hot encoded variables, representing categories as separate binary columns
one_hot_encoded_vars = ['Composite', 'Glass Ionomer Cement', 'Amalgam', 'No ini
                        'Repaired baseline', 'Replaced baseline', 'CARS No cari
                        'CARS Moderate/advanced', 'FDI No adaptation', 'FDI Ini
                        'FDI Moderate/advanced adaptation', 'FDI No caries', 'F
                        'FDI Moderate/advanced caries']
```

```
def descriptive_statistics(X_train, y_train, X_test, y_test):
    # Merge the feature DataFrame (X) and target variable Series (y) for both t
    # This facilitates combined operations for descriptive statistics.
    train_data = pd.concat([X_train, y_train], axis=1)
    test_data = pd.concat([X_test, y_test], axis=1)

    # Print a heading and then descriptive statistics (count, mean, std, min, c
    print("Descriptive Statistics for Numeric Variables:")
    print("\nTraining Set:")
    print(train_data[numeric_vars].describe())
    print("\nTest Set:")
    print(test_data[numeric_vars].describe())

    # Initialize a dictionary to hold statistics for categorical variables.
    stats = {}
    # Calculate and store counts and percentages for original (non-encoded) cat
    for var in original_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
            }
        }

    # Handle one-hot encoded variables by identifying all columns that match th
    # Then calculate counts and percentages for these as well.
    for var in one_hot_encoded_vars:
        encoded_columns = [col for col in train_data if col.startswith(var)]
        for col in encoded_columns:
            stats[col] = {
                "Training Set": {
                    "Count": train_data[col].value_counts().to_dict(),
                    "Percentage": (train_data[col].value_counts(normalize=True)
                },
                "Test Set": {
                    "Count": test_data[col].value_counts().to_dict(),
                    "Percentage": (test_data[col].value_counts(normalize=True)
                }
            }

    # Print the calculated statistics for each categorical variable, both origi
    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, passing the training and test datasets (features and targe
descriptive_statistics(X_train, y_train, X_test, y_test)
```

```
Descriptive Statistics for Numeric Variables:

Training Set:
                Age       Decayed        Filled       Missing
count  5.070000e+02  5.070000e+02  5.070000e+02  5.070000e+02
mean   6.376666e-16  6.306592e-17  1.121172e-16  5.605860e-17
std    1.000988e+00  1.000988e+00  1.000988e+00  1.000988e+00
min   -2.015079e+00 -7.869911e-01 -1.942576e+00 -6.040245e-01
25%   -7.306641e-01 -7.869911e-01 -7.540572e-01 -6.040245e-01
50%   -8.845659e-02 -3.045189e-01  3.828892e-02 -6.040245e-01
75%    7.143028e-01  6.604254e-01  8.306351e-01  4.593103e-01
max    2.105753e+00  5.967620e+00  2.415327e+00  3.649315e+00

Test Set:
              Age     Decayed      Filled     Missing
count  130.000000  130.000000  130.000000  130.000000
mean    -0.181083    0.096304    0.093144   -0.015101
std      0.994113    0.821815    1.029736    0.973496
min     -2.122114   -0.786991   -1.942576   -0.604025
25%     -0.944733   -0.304519   -0.754057   -0.604025
50%     -0.356043   -0.063283    0.038289   -0.604025
75%      0.607268    0.660425    0.830635    0.459310
max      1.463545    2.590314    2.415327    2.585980

Sex Statistics:

Training Set:
Count: {0: 262, 1: 245}
Percentage: {0: 51.67652859960552, 1: 48.32347140039448}

Test Set:
Count: {0: 82, 1: 48}
Percentage: {0: 63.07692307692307, 1: 36.92307692307693}

Tooth type Statistics:

Training Set:
Count: {0: 432, 1: 75}
Percentage: {0: 85.20710059171599, 1: 14.792899408284024}

Test Set:
Count: {0: 117, 1: 13}
Percentage: {0: 90.0, 1: 10.0}

Arch Statistics:
```

```
Training Set:
Count: {0: 266, 1: 241}
Percentage: {0: 52.46548323471401, 1: 47.53451676528599}

Test Set:
Count: {0: 69, 1: 61}
Percentage: {0: 53.07692307692308, 1: 46.92307692307692}

N surfaces Statistics:

Training Set:
Count: {1: 207, 2: 139, 3: 72, 4: 60, 5: 29}
Percentage: {1: 40.828402366863905, 2: 27.416173570019726, 3: 14.2011834319
```

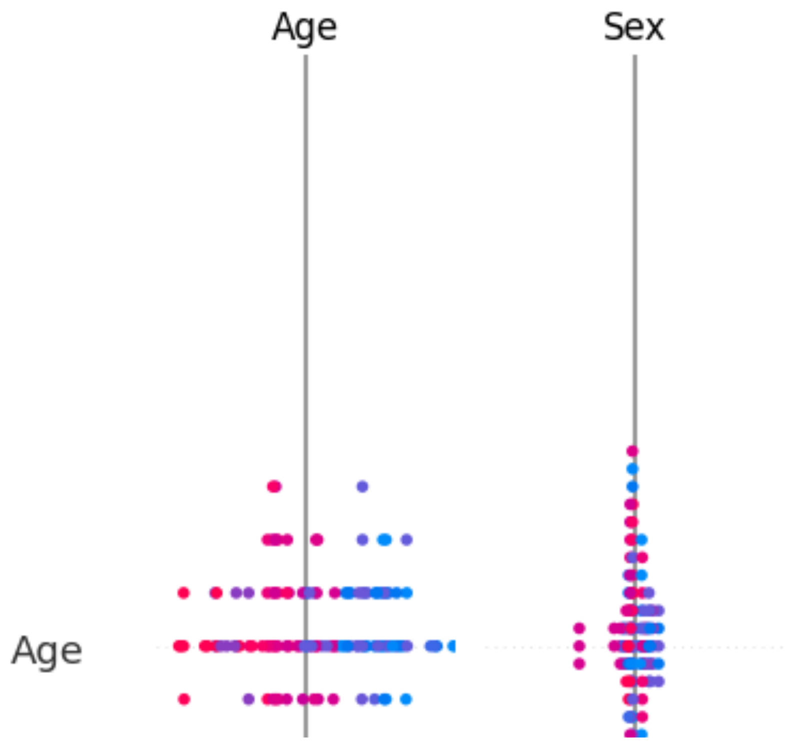
```
# Define custom metrics
def sensitivity(y_true, y_pred):
    tn, fp, fn, tp = confusion_matrix(y_true, y_pred).ravel()
    return tp / (tp + fn)

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

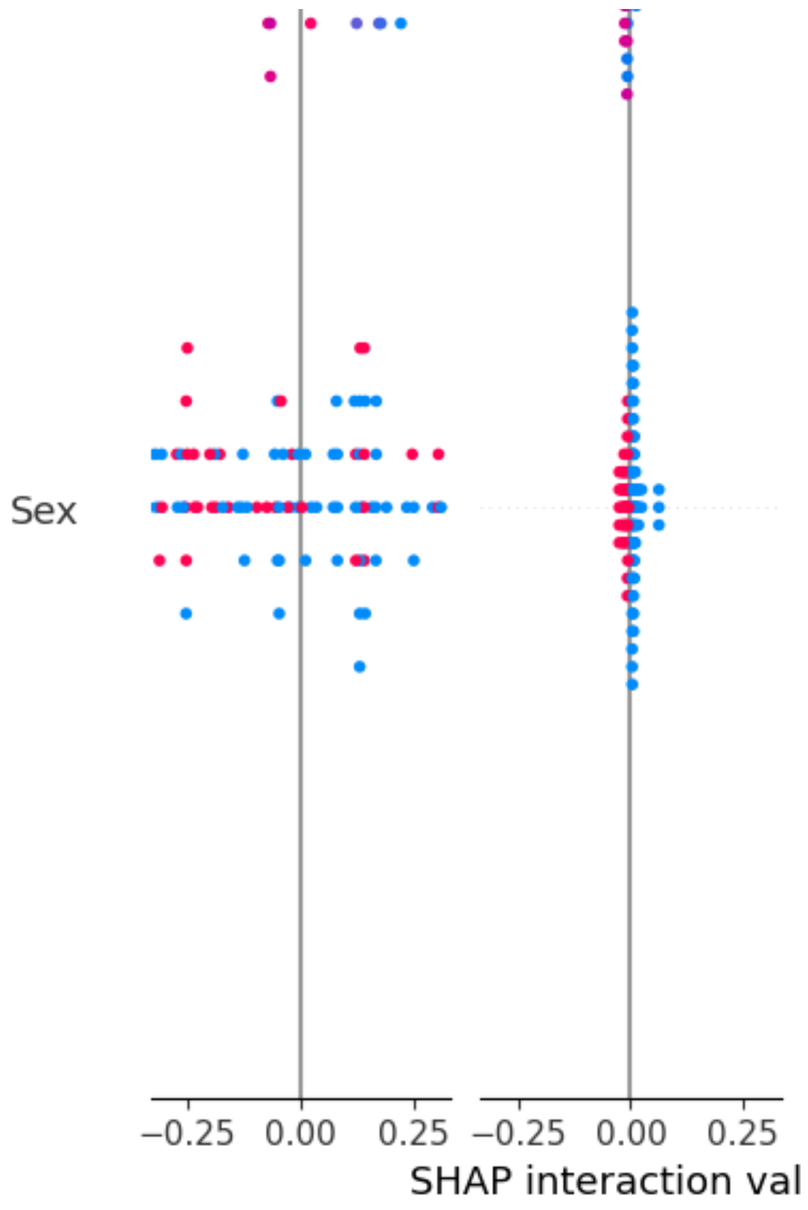
```
import pandas as pd
import numpy as np
import shap
import sys
import tensorflow as tf
import matplotlib.pyplot as plt
import random
import seaborn as sns
from sklearn.model_selection import cross_val_score
from sklearn.calibration import CalibratedClassifierCV
from sklearn.tree import DecisionTreeClassifier, 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


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

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

    clf = GridSearchCV(model, grid, cv=inner_cv, scoring='roc_auc')
    nested_scores = cross_validate(clf, X=X_train, y=y_train, cv=outer_cv, scor

    clf.fit(X_train, y_train)
    best_model = clf.best_estimator_
    best_params = clf.best_params_
    print(f"Best parameters for {name}: {best_params}")

    calibrated_clf = CalibratedClassifierCV(estimator=best_model, method='sigmc
    calibrated_clf.fit(X_train, y_train)

    y_probs = calibrated_clf.predict_proba(X_test)[:, 1]

    # Calculate ROC curve and AUC
    fpr, tpr, thresholds = roc_curve(y_test, y_probs)
    roc_auc = auc(fpr, tpr)

    print("\n--- ROC Data for Copying ---")
    print("FPR =", fpr.tolist())
    print("TPR =", tpr.tolist())
    print("AUC =", roc_auc)
    print("--- End of ROC Data ---\n")

    # --- 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_

    # --- 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: {manu

    # --- 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}")

    calculate_and_plot_shap(best_model, X_train, X_test, name)

    # Prepare dictionary of test metrics 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, calibratec

def calculate_and_plot_shap(model, X_train, X_test, model_name):
    if isinstance(model, DecisionTreeClassifier):
        explainer = shap.TreeExplainer(model)
    else:
        explainer = shap.KernelExplainer(model.predict_proba, X_train.sample(10
    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()

def evaluate_decision_tree(X_train, y_train, X_test, y_test, cv, scoring, manua
    model = DecisionTreeClassifier(random_state=seed)
    grid = {
        'max_depth': [6],
        'criterion': ['gini'],
        'min_samples_split': [4],
        'min_samples_leaf': [8],
        'ccp_alpha': [0.001]
    }
    return evaluate_model(model, "Decision Tree", grid, X_train, y_train, X_tes

def main(X_train, y_train, 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 = 0.35
    threshold_list = np.arange(0.1, 1.05, 0.05)

    aggregated_metrics = []

    # Loop over seeds
    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_t
```

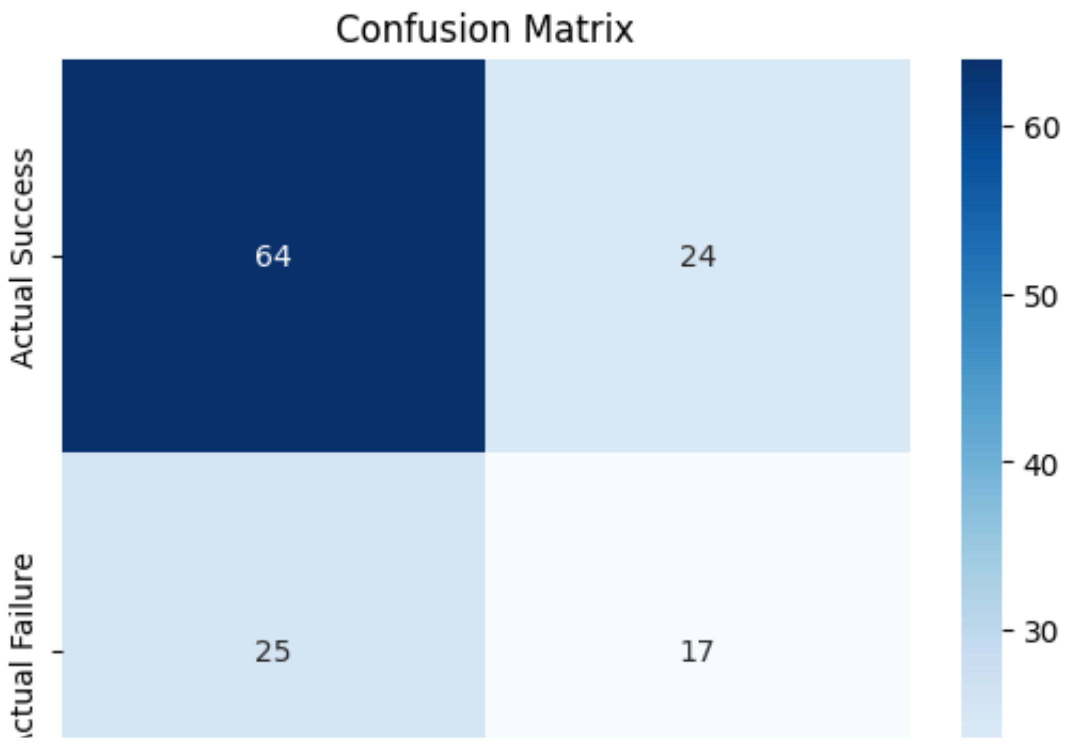
```
            X_train, y_train, X_test, y_test, cv, scoring, manual_threshold, th
        )

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

        plot_confusion_matrix(y_test, y_pred_manual)
        plot_roc_curve(y_test, y_probs)

        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)

    # Compute mean, standard error, and 95% confidence interval for each metric
    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)
        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
    for metric, summary in metrics_summary.items():
        print(f"{metric.capitalize()}: Mean = {summary['Mean']:.3f}, SE = {summ
               f"95% CI = [{summary['95% CI'][0]:.3f}, {summary['95% CI'][1]:.3f

# RUN THE MAIN FUNCTION (Ensure X_train, y_train, X_test, y_test are defined)
if __name__ == '__main__':
    main(X_train, y_train, X_test, y_test)
```

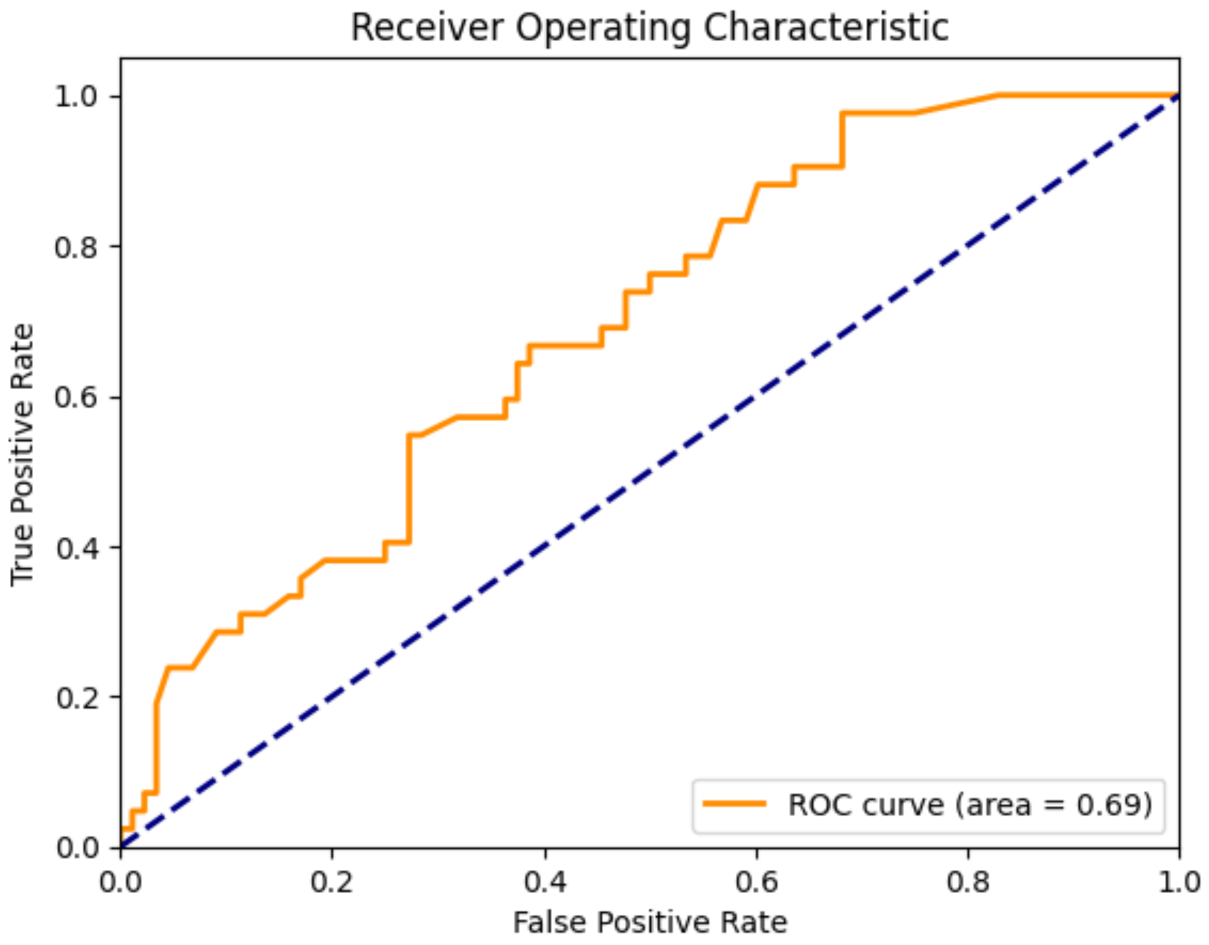
```
Running evaluation with seed 40
```

```
Evaluating Decision Tree with seed 40...
Best parameters for Decision Tree: {'ccp_alpha': 0.001, 'criterion': 'gini'

--- ROC Data for Copying ---
FPR = [0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.022727272727
TPR = [0.0, 0.023809523809523808, 0.023809523809523808, 0.04761904761904761
AUC = 0.6910173160173161
--- End of ROC Data ---

Training - Accuracy: 0.759, Sensitivity: 0.397, Specificity: 0.913, F1: 0.4

Test Metrics for manual threshold 0.35:
Accuracy: 0.623, Sensitivity: 0.405, Specificity: 0.727, F1: 0.410, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5076923076923077, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.5923076923076923, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6230769230769231, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7, 'Sensitivity': 0.14285714285714
Threshold: 0.45, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Decision Tree
```



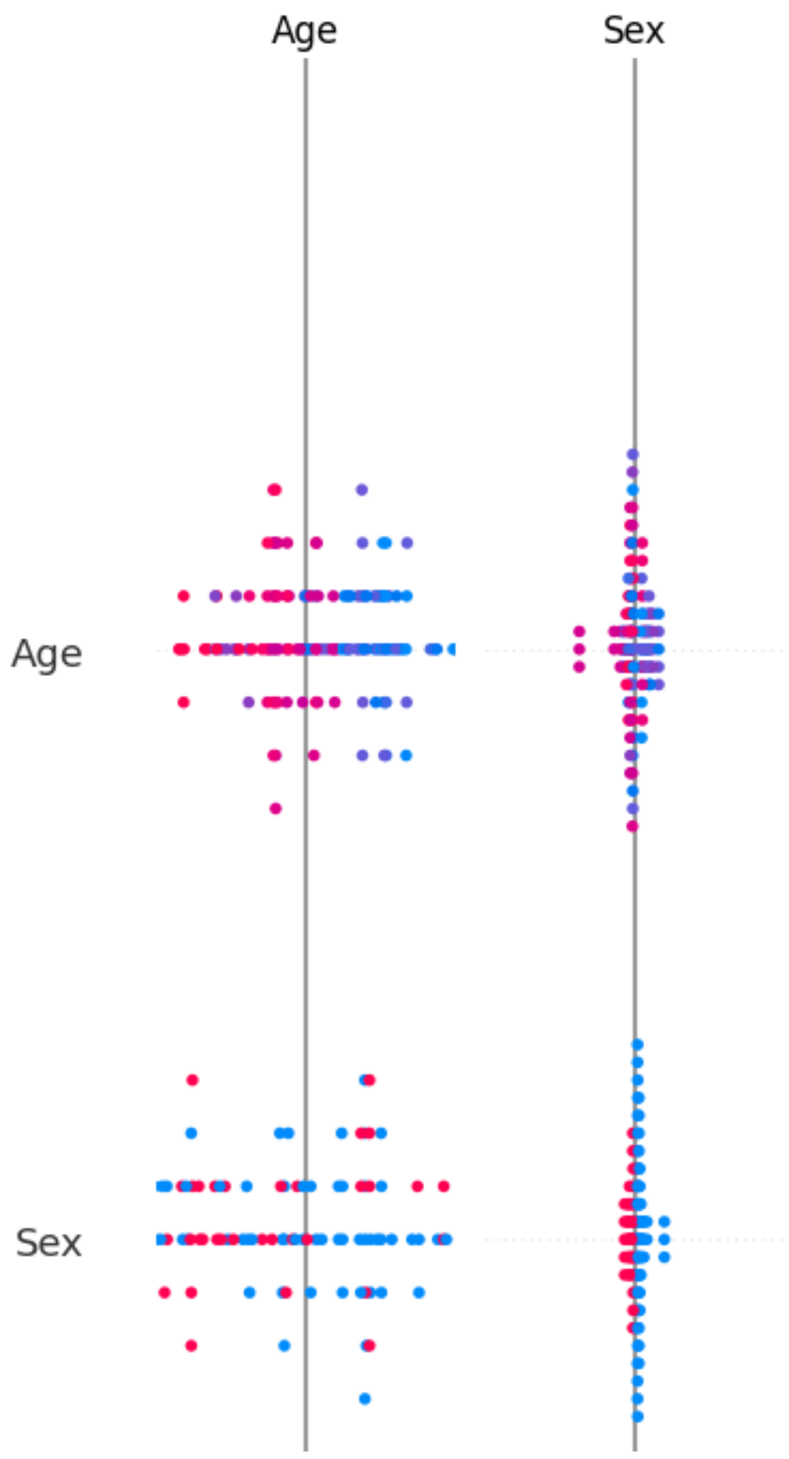

Sex
−0.25
0.00
0.25
−0.25
0.00
0.25
SHAP interaction val

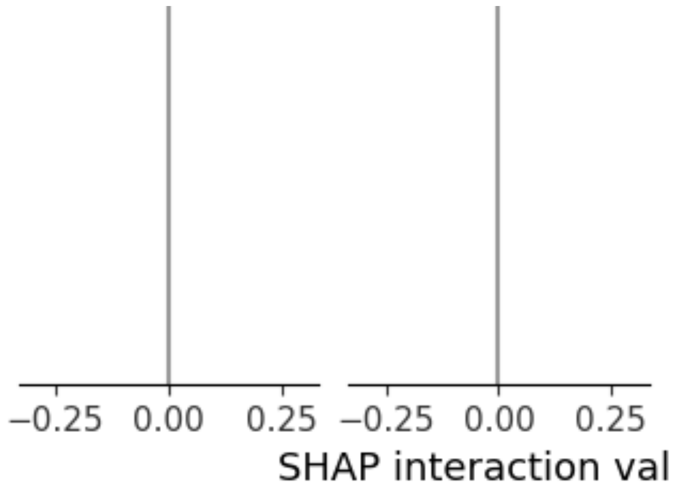

Confusion Matrix
Actual Success
Actual Failure
64
24
25
17
60
50
40
30

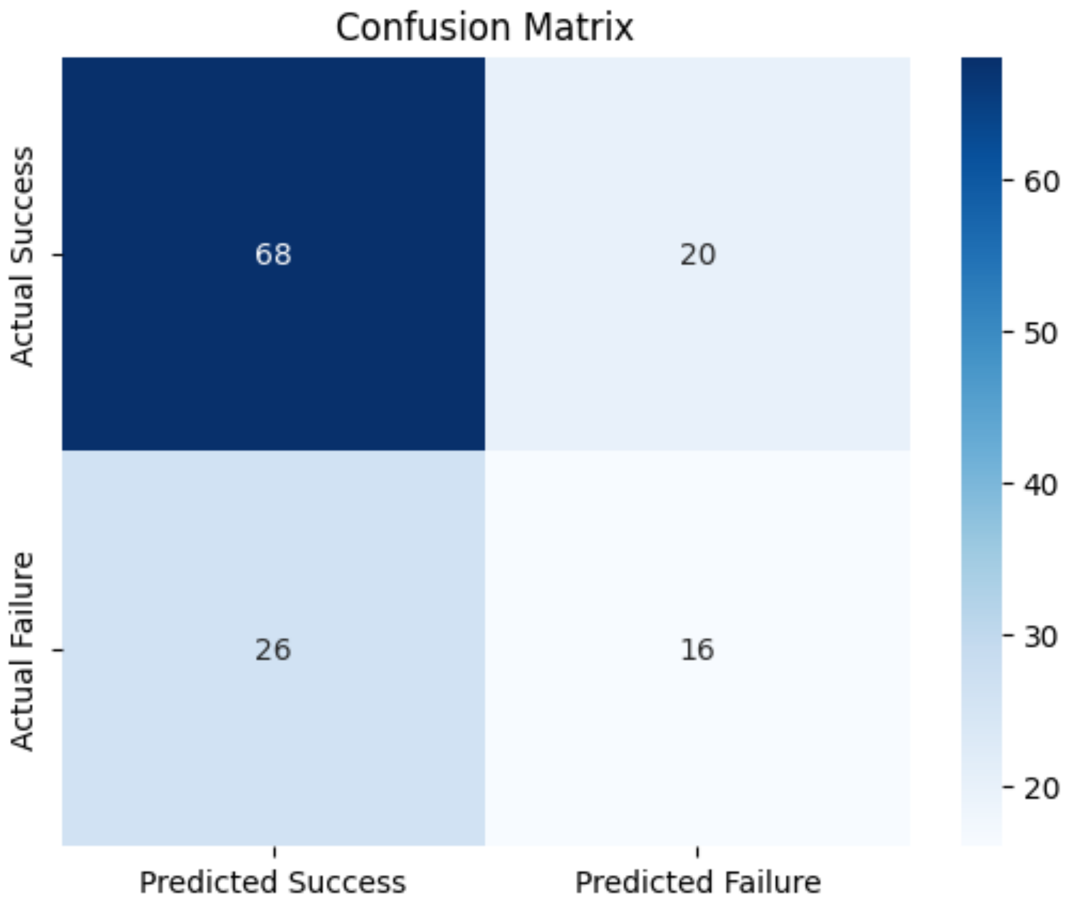


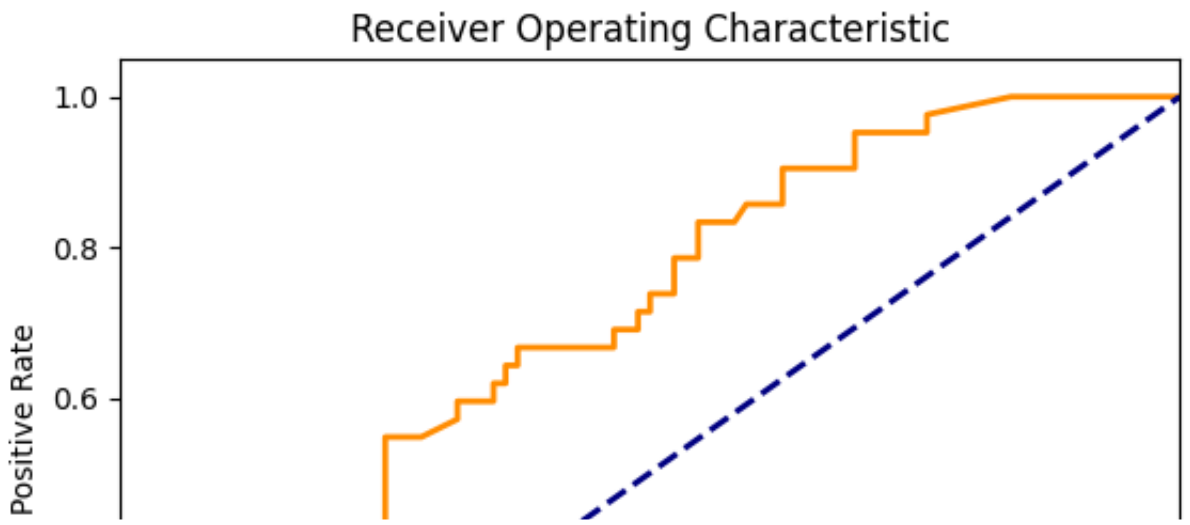
```
Running evaluation with seed 41

Evaluating Decision Tree with seed 41...
Best parameters for Decision Tree: {'ccp_alpha': 0.001, 'criterion': 'gini'

--- ROC Data for Copying ---
FPR = [0.0, 0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.0340909
TPR = [0.0, 0.023809523809523808, 0.047619047619047616, 0.04761904761904761
AUC = 0.6930465367965367
--- End of ROC Data ---

Training - Accuracy: 0.759, Sensitivity: 0.397, Specificity: 0.913, F1: 0.4

Test Metrics for manual threshold 0.35:
Accuracy: 0.646, Sensitivity: 0.381, Specificity: 0.773, F1: 0.410, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5153846153846153, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.5923076923076923, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6461538461538462, 'Sensitivity': 0
```

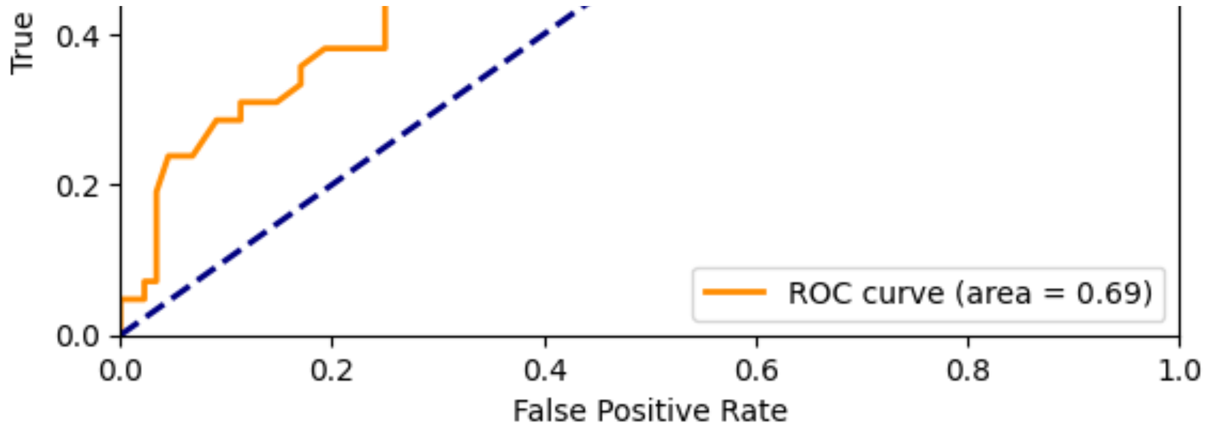
```
Threshold: 0.40, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.45, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Decision Tree
```



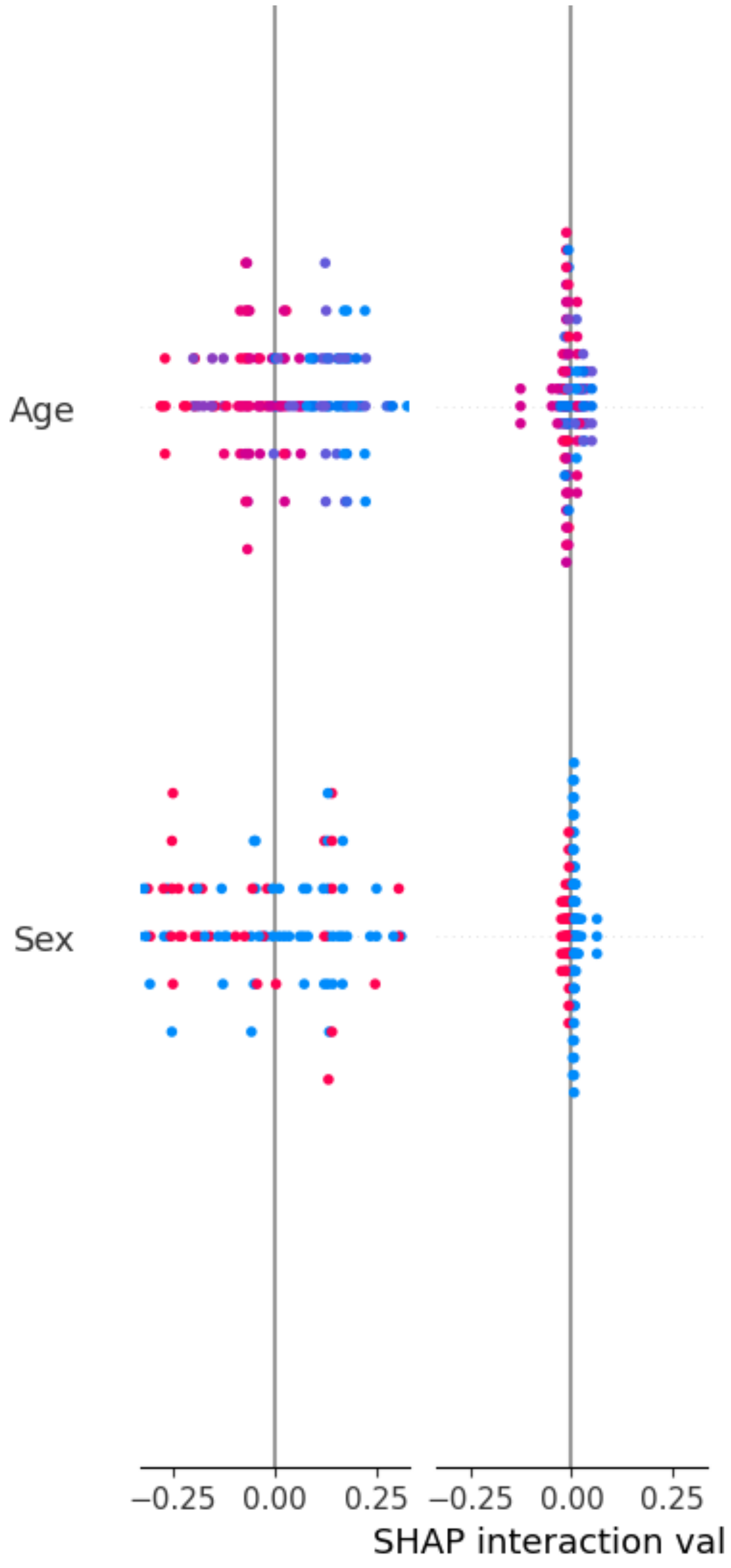

−0.25
0.00
0.25
−0.25
0.00
0.25
SHAP interaction val


Confusion Matrix
Actual Success
Actual Failure
Predicted Success
Predicted Failure
68
20
26
16
60
50
40
30
20

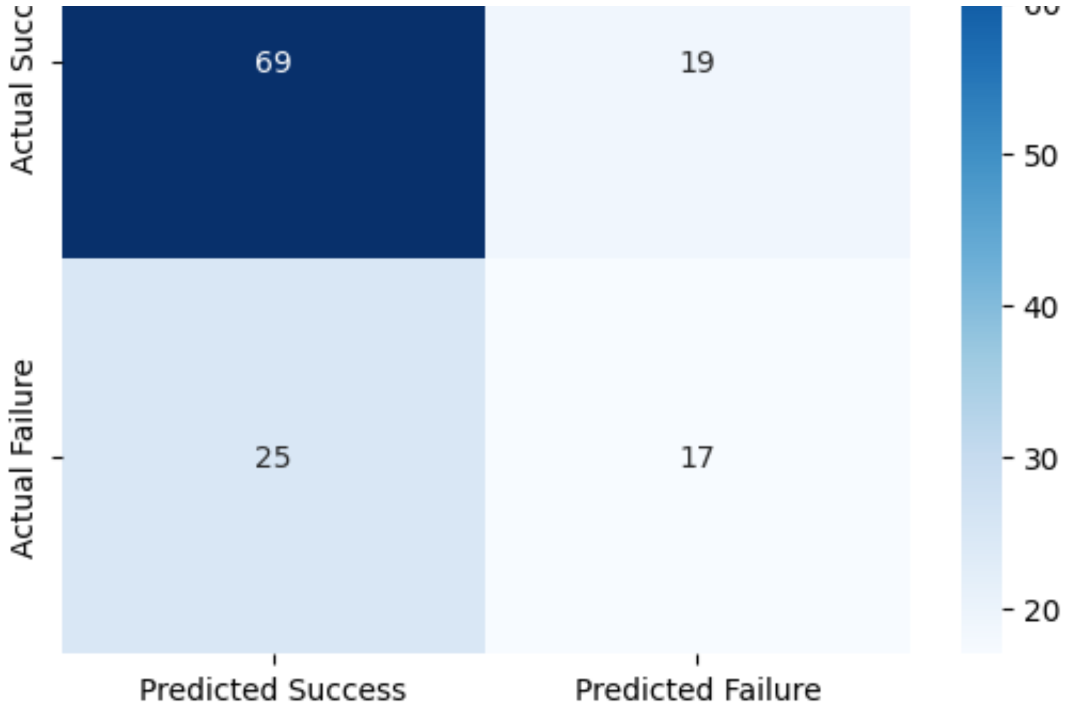

Receiver Operating Characteristic
Positive Rate
1.0
0.8
0.6

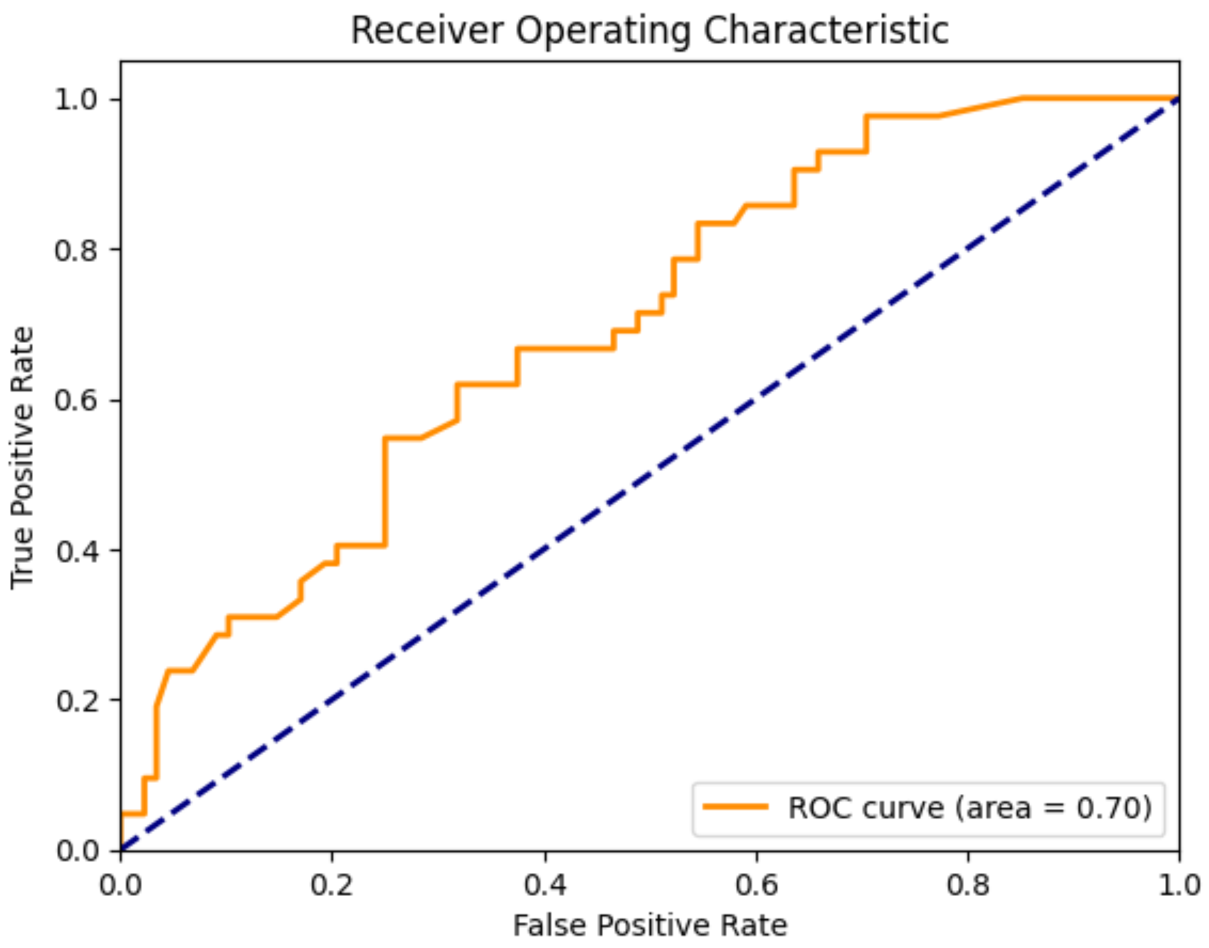


```
Running evaluation with seed 42

Evaluating Decision Tree with seed 42...
Best parameters for Decision Tree: {'ccp_alpha': 0.001, 'criterion': 'gini'

--- ROC Data for Copying ---
FPR = [0.0, 0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.0340909
TPR = [0.0, 0.023809523809523808, 0.047619047619047616, 0.04761904761904761
AUC = 0.6960227272727273
--- End of ROC Data ---

Training - Accuracy: 0.759, Sensitivity: 0.397, Specificity: 0.913, F1: 0.4

Test Metrics for manual threshold 0.35:
Accuracy: 0.662, Sensitivity: 0.405, Specificity: 0.784, F1: 0.436, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.47692307692307695, 'Sensitivity':
Threshold: 0.30, Metrics: {'Accuracy': 0.5846153846153846, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6615384615384615, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.45, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Decision Tree
```

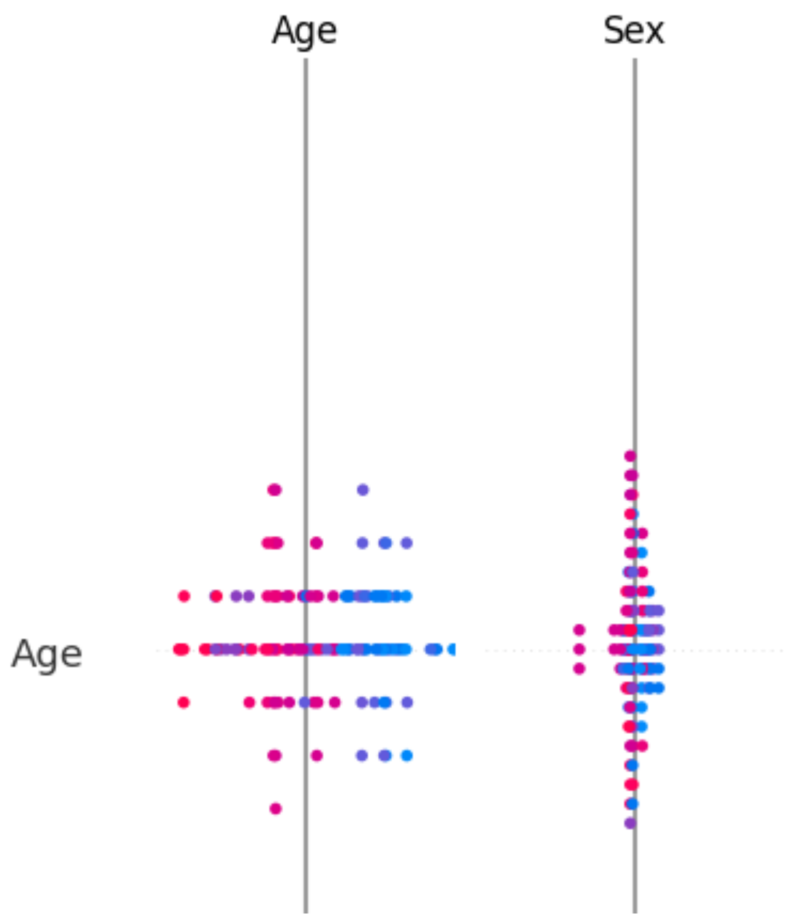

Age
Sex
−0.25
0.00
0.25
−0.25
0.00
0.25
SHAP interaction val

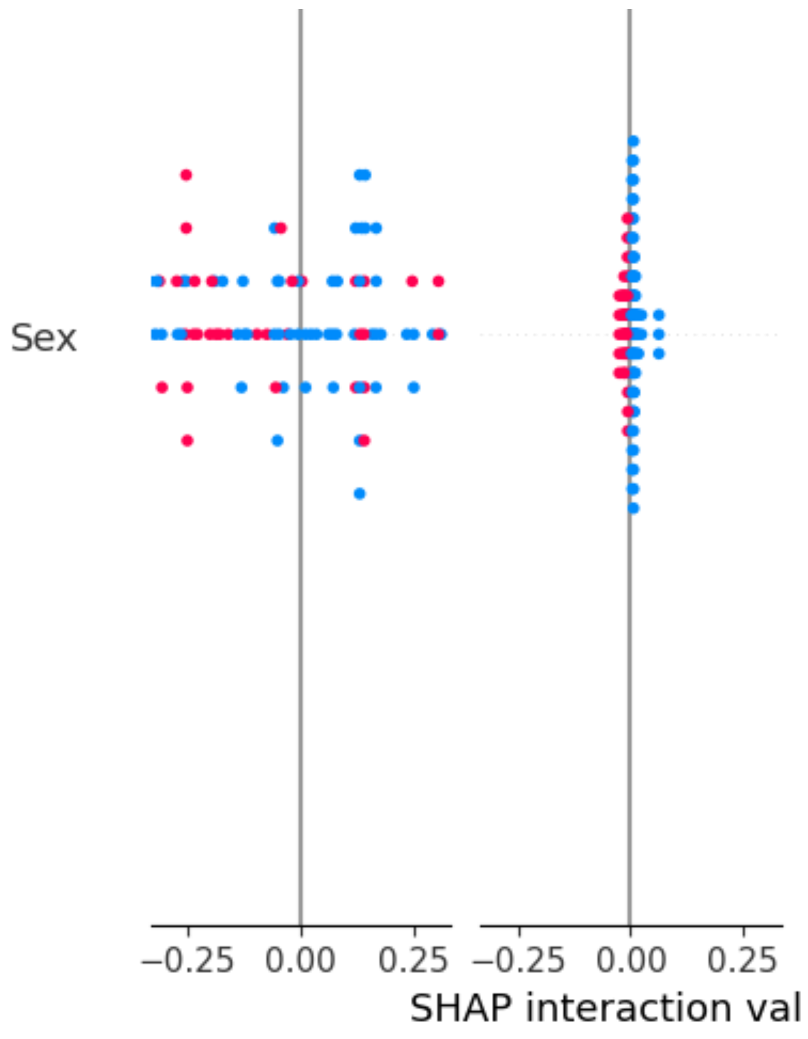
Confusion Matrix
60

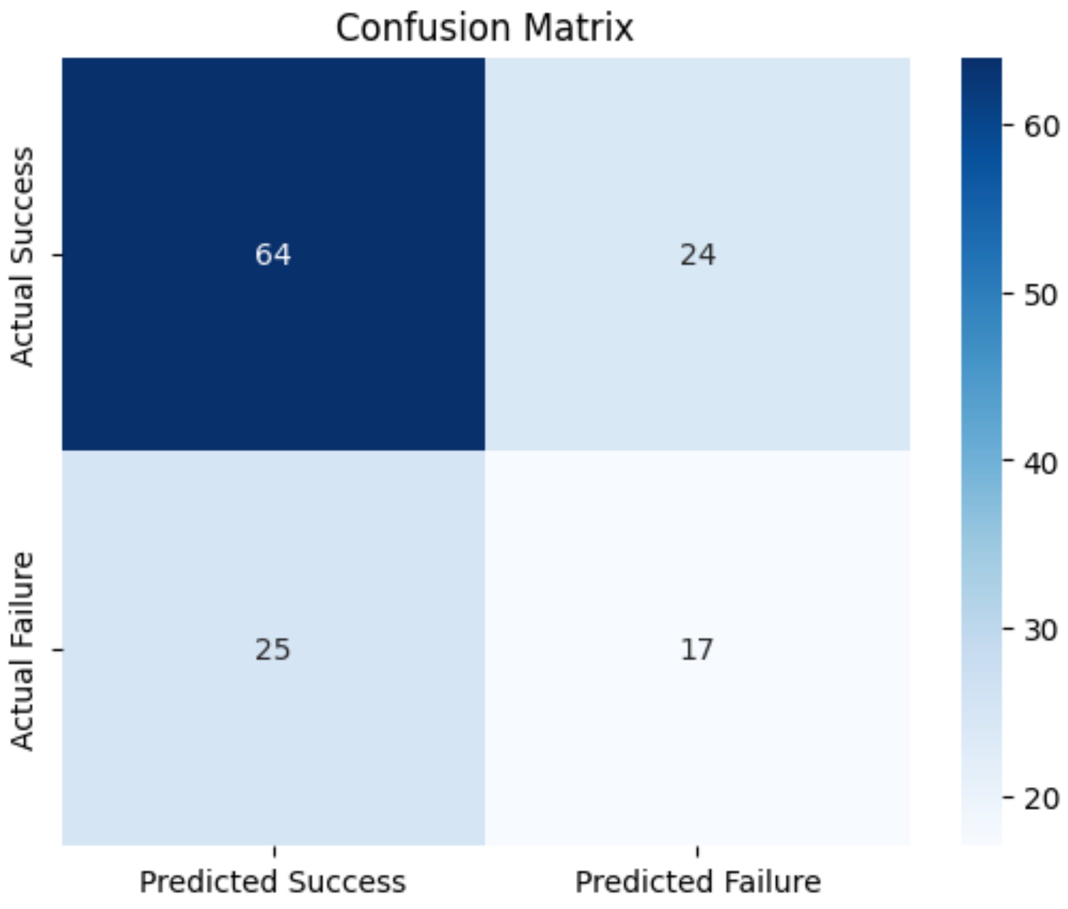


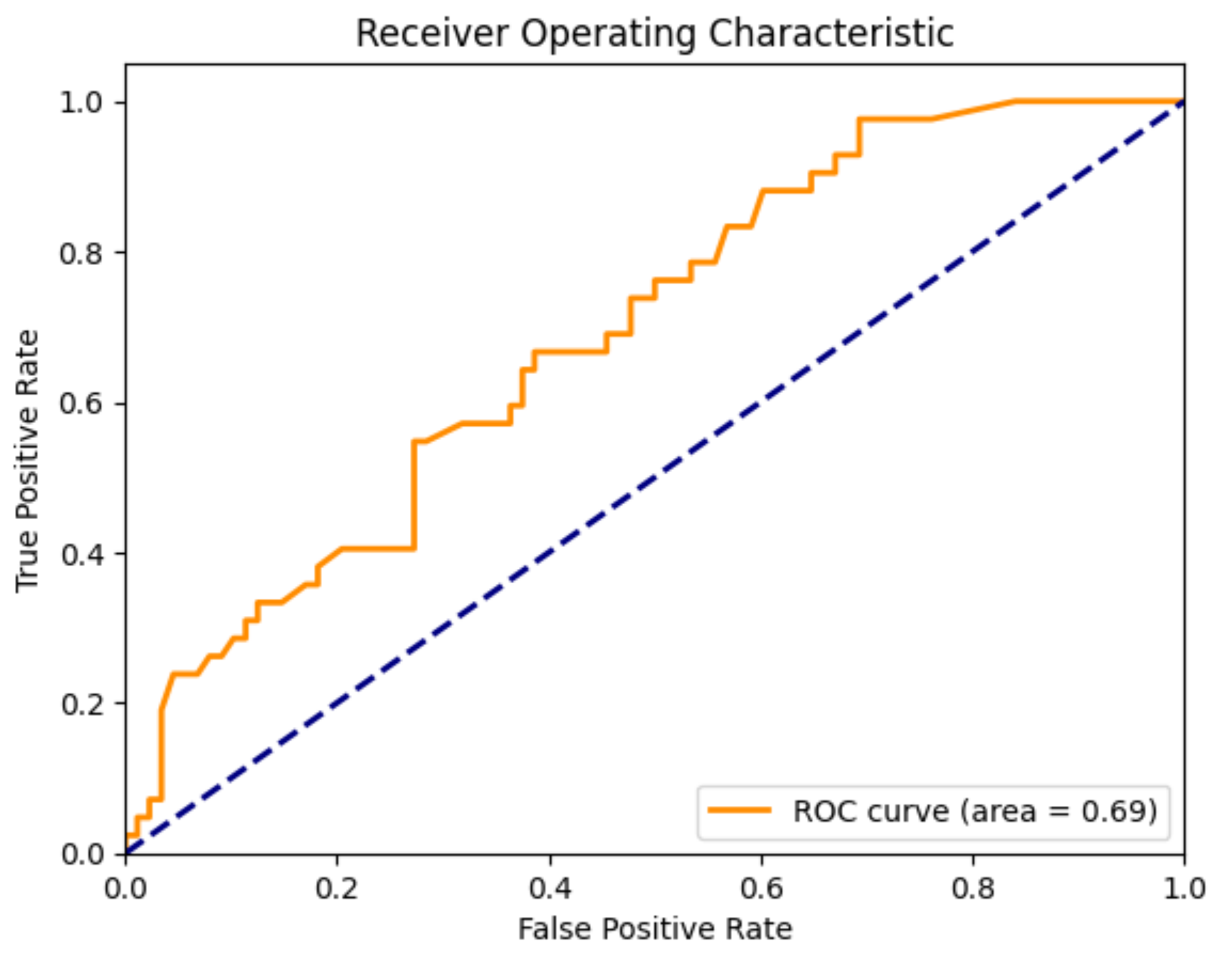


```
Running evaluation with seed 43

Evaluating Decision Tree with seed 43...
Best parameters for Decision Tree: {'ccp_alpha': 0.001, 'criterion': 'gini'

--- ROC Data for Copying ---
FPR = [0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.022727272727
FPR = [0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.022727272727
TPR = [0.0, 0.023809523809523808, 0.023809523809523808, 0.047619047619047616
AUC = 0.6920995670995671
--- End of ROC Data ---

Training - Accuracy: 0.759, Sensitivity: 0.397, Specificity: 0.913, F1: 0.4

Test Metrics for manual threshold 0.35:
Accuracy: 0.623, Sensitivity: 0.405, Specificity: 0.727, F1: 0.410, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5076923076923077, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.5923076923076923, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6230769230769231, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7, 'Sensitivity': 0.14285714285714
Threshold: 0.45, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Decision Tree
```



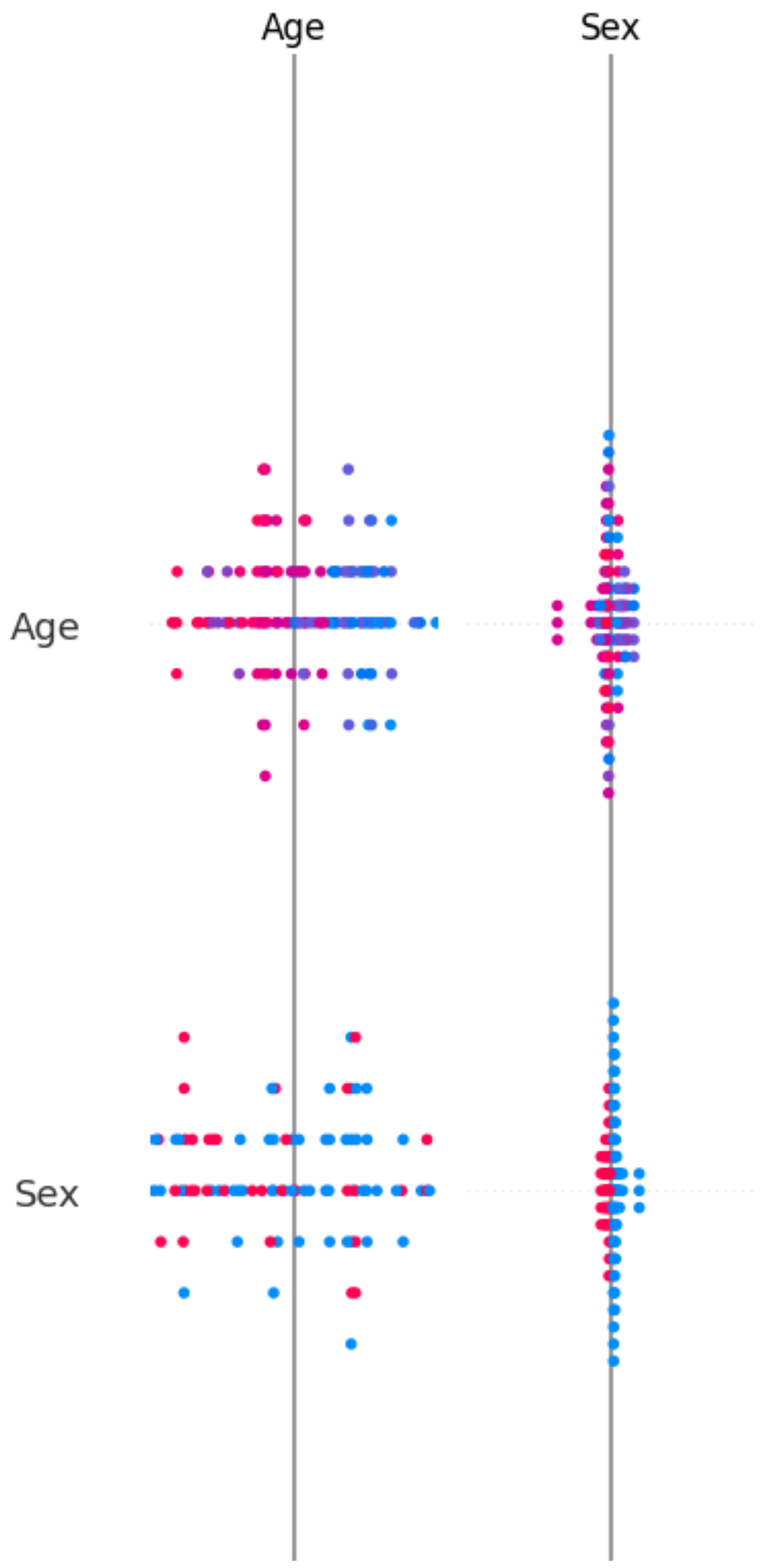

Sex
−0.25
0.00
0.25
−0.25
0.00
0.25
SHAP interaction val

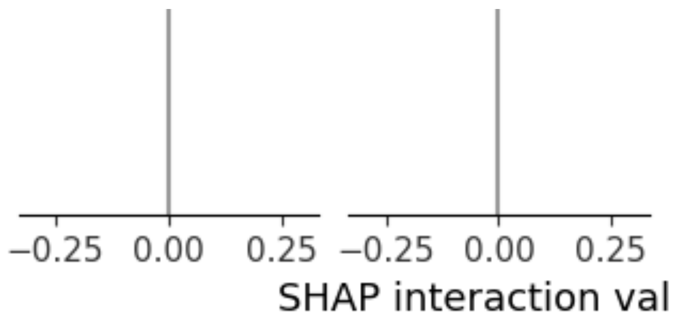

Confusion Matrix
Actual Success
Actual Failure
Predicted Success
Predicted Failure
64
24
25
17
60
50
40
30
20

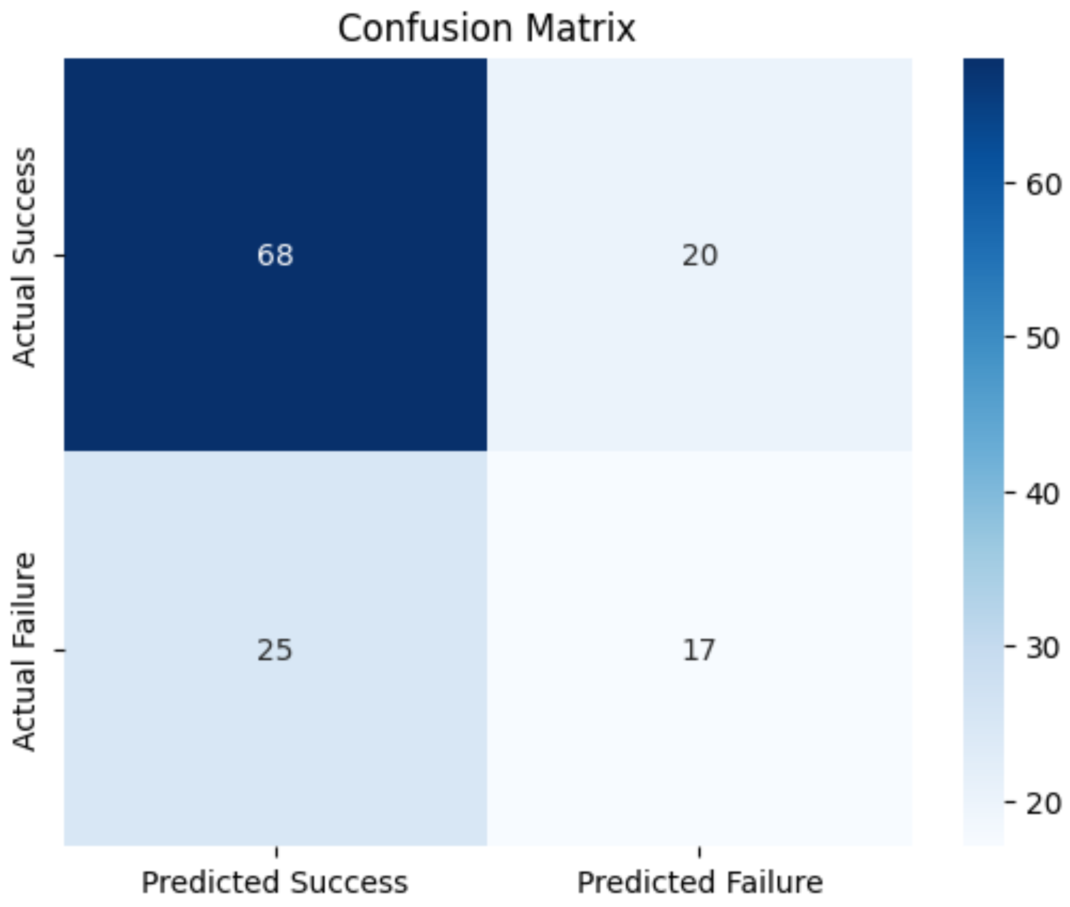


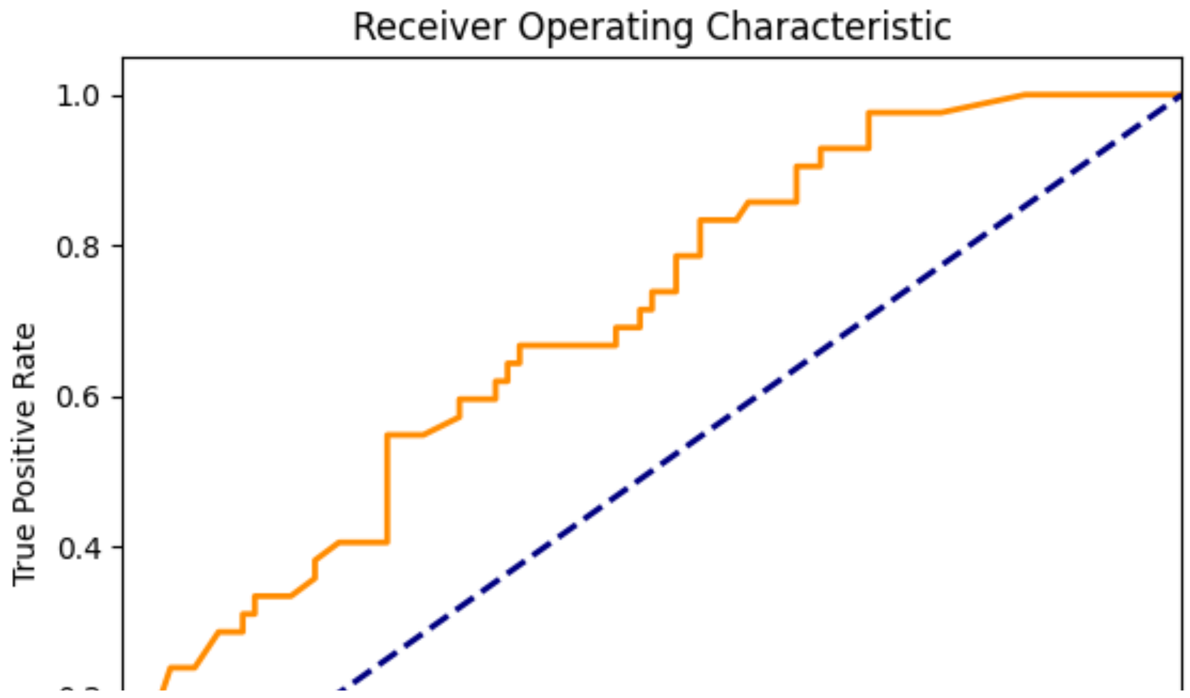
```
Running evaluation with seed 44

Evaluating Decision Tree with seed 44...
Best parameters for Decision Tree: {'ccp_alpha': 0.001, 'criterion': 'gini'

--- ROC Data for Copying ---
FPR = [0.0, 0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.0340909
TPR = [0.0, 0.023809523809523808, 0.047619047619047616, 0.04761904761904761
AUC = 0.6962932900432901
--- End of ROC Data ---

Training - Accuracy: 0.759, Sensitivity: 0.397, Specificity: 0.913, F1: 0.4

Test Metrics for manual threshold 0.35:
Accuracy: 0.654, Sensitivity: 0.405, Specificity: 0.773, F1: 0.430, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5153846153846153, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.5923076923076923, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6538461538461539, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.45, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
```

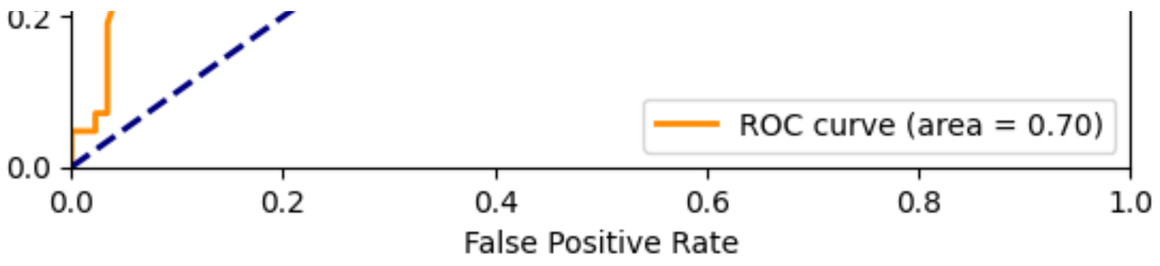
```
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Decision Tree
```



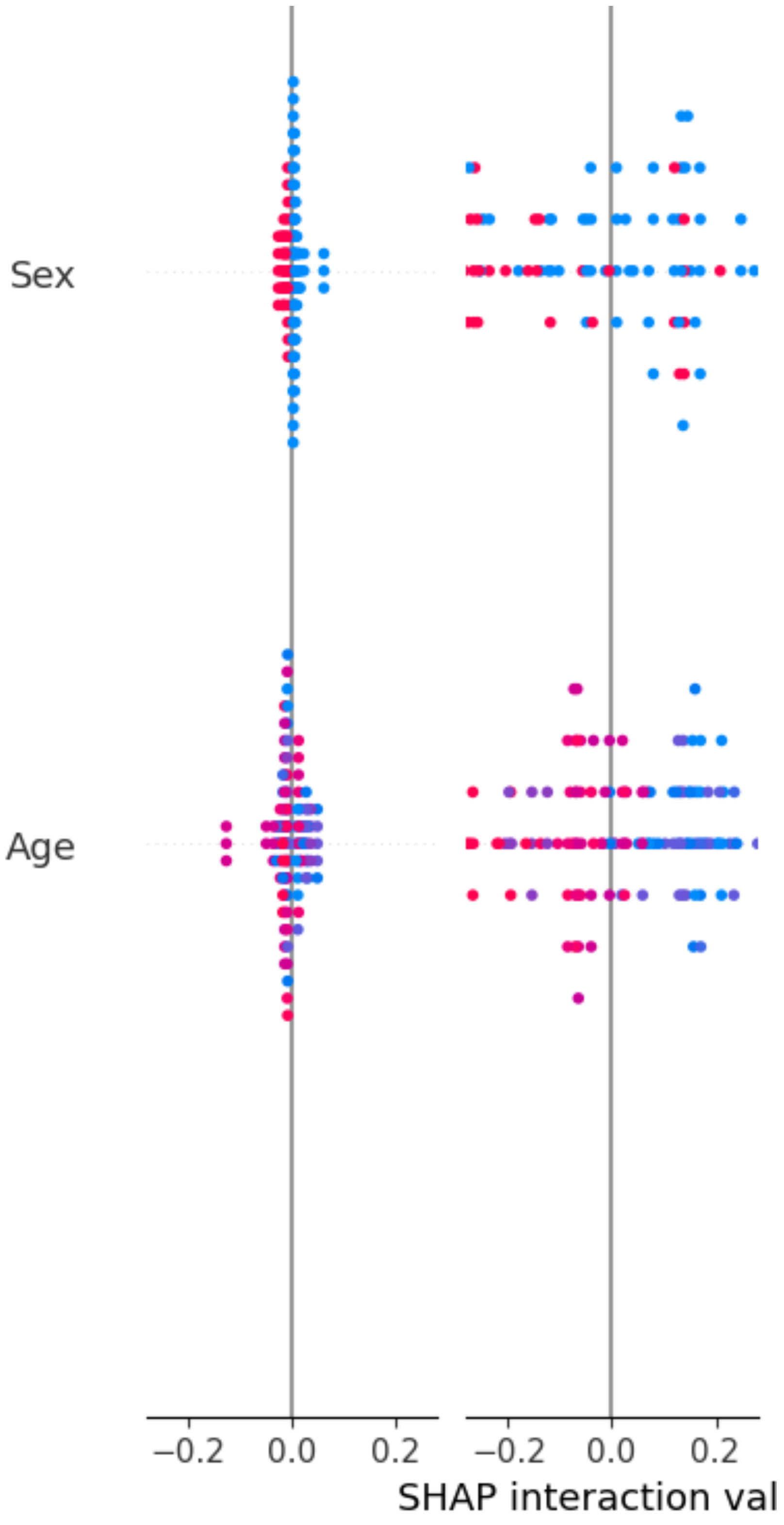

−0.25
0.00
0.25
−0.25
0.00
0.25
SHAP interaction val

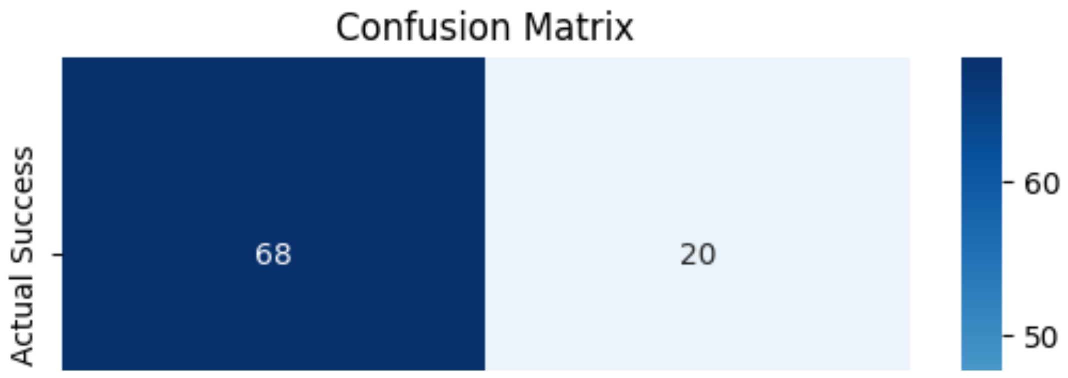

Confusion Matrix
Actual Success
Actual Failure
Predicted Success
Predicted Failure
68
20
25
17
60
50
40
30
20

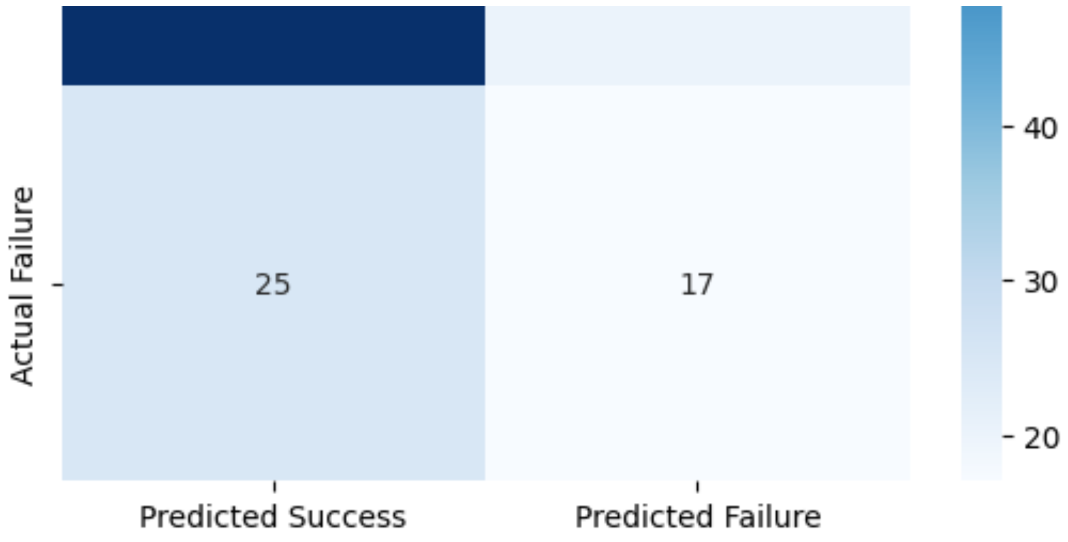

Receiver Operating Characteristic
True Positive Rate
1.0
0.8
0.6
0.4

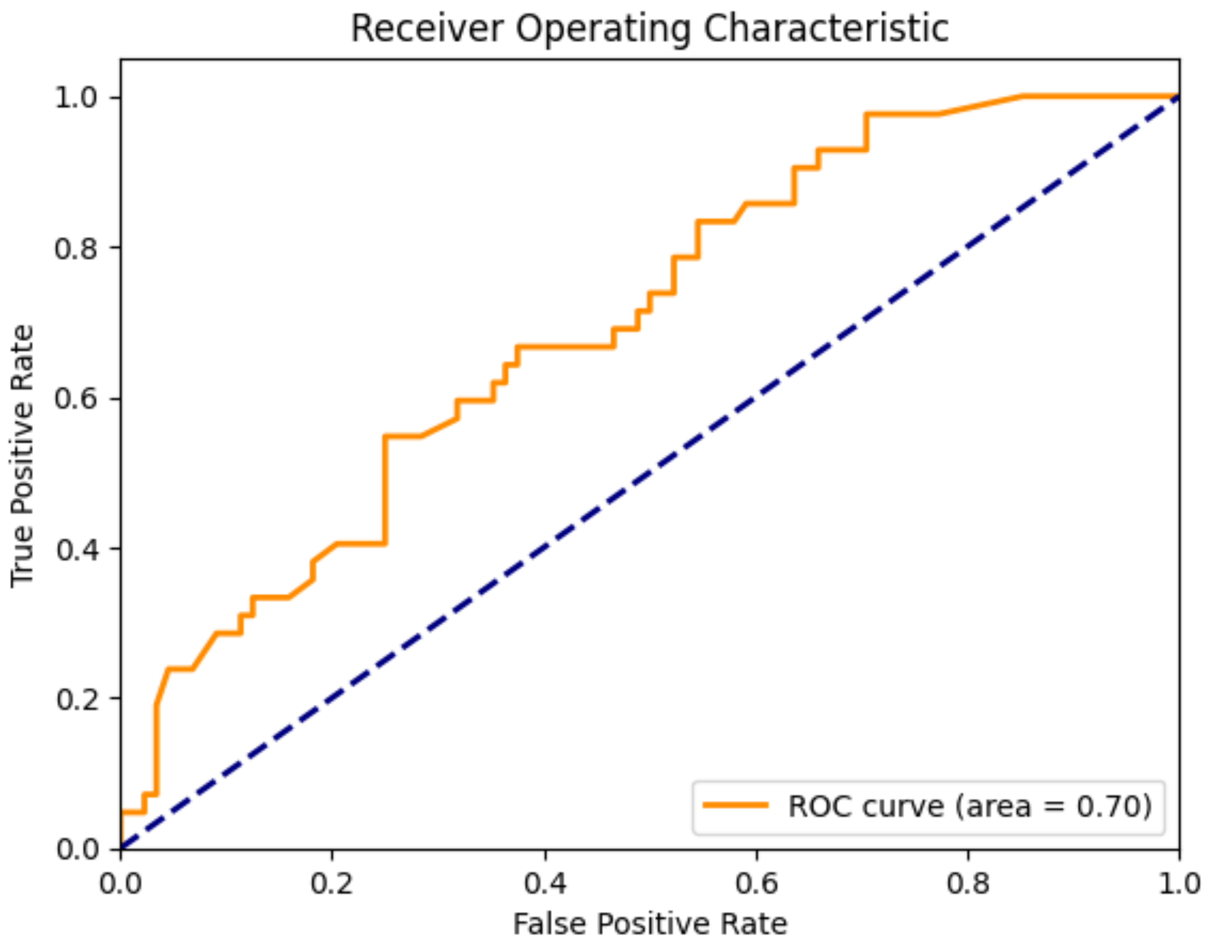


```
Running evaluation with seed 45

Evaluating Decision Tree with seed 45...
Best parameters for Decision Tree: {'ccp_alpha': 0.001, 'criterion': 'gini'

--- ROC Data for Copying ---
FPR = [0.0, 0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.0340909
TPR = [0.0, 0.023809523809523808, 0.047619047619047616, 0.04761904761904761
AUC = 0.6962932900432901
--- End of ROC Data ---

Training - Accuracy: 0.759, Sensitivity: 0.397, Specificity: 0.913, F1: 0.4

Test Metrics for manual threshold 0.35:
Accuracy: 0.654, Sensitivity: 0.405, Specificity: 0.773, F1: 0.430, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5153846153846153, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.5923076923076923, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6538461538461539, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.45, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Decision Tree
```


Sex
Age
−0.2
0.0
0.2
−0.2
0.0
0.2
SHAP interaction val

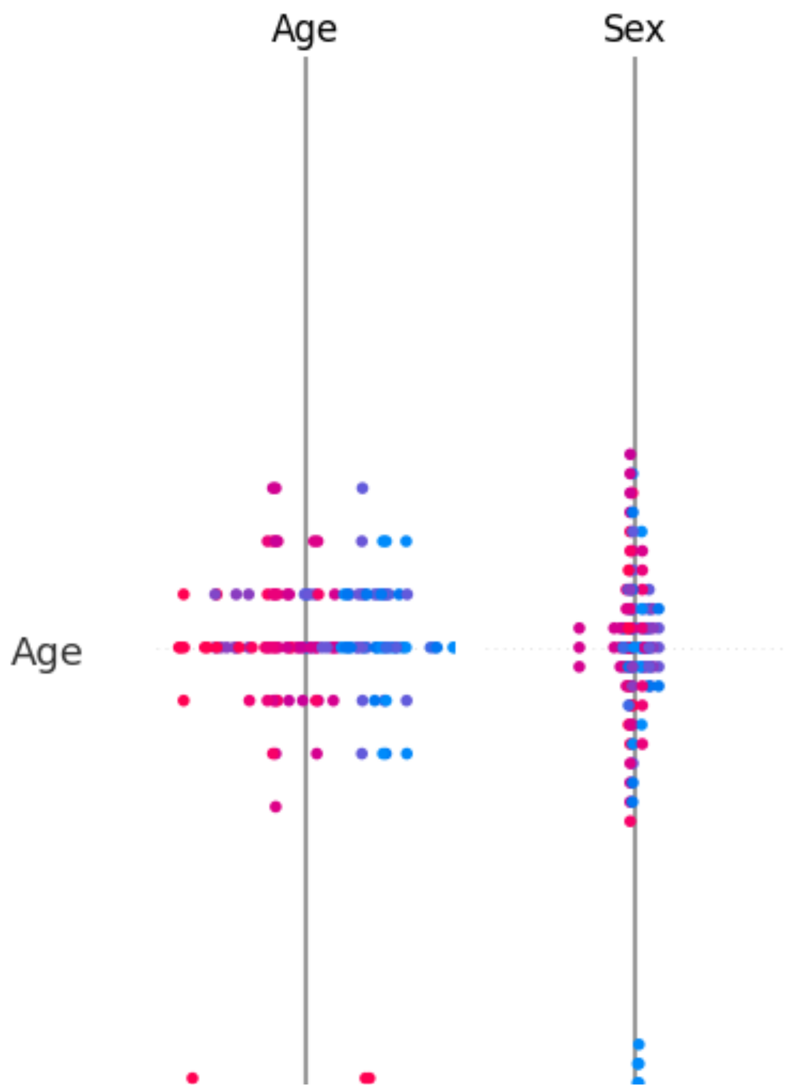

Confusion Matrix
Actual Success
68
20
60
50

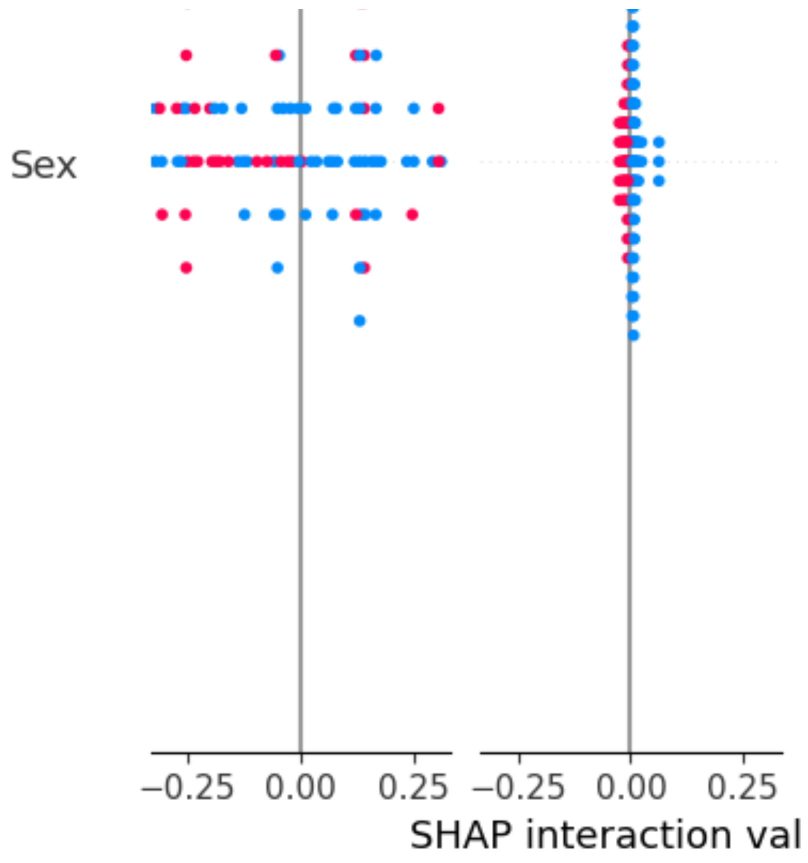


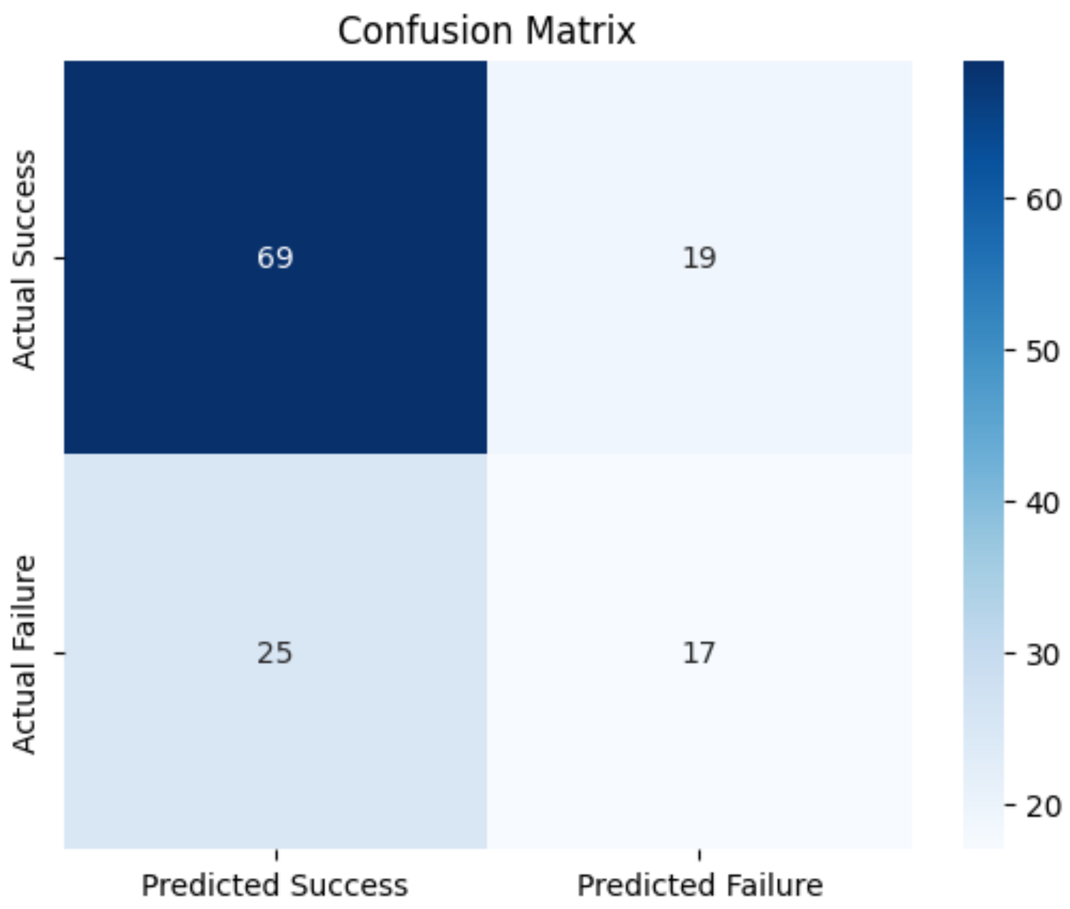


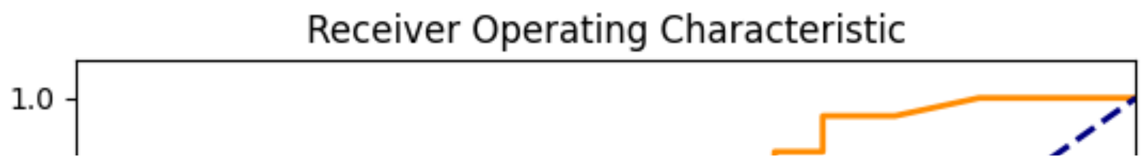
```
Running evaluation with seed 46

Evaluating Decision Tree with seed 46...
Best parameters for Decision Tree: {'ccp_alpha': 0.001, 'criterion': 'gini'

--- ROC Data for Copying ---
FPR = [0.0, 0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.0340909
TPR = [0.0, 0.023809523809523808, 0.047619047619047616, 0.04761904761904761
AUC = 0.6981872294372294
--- End of ROC Data ---
```

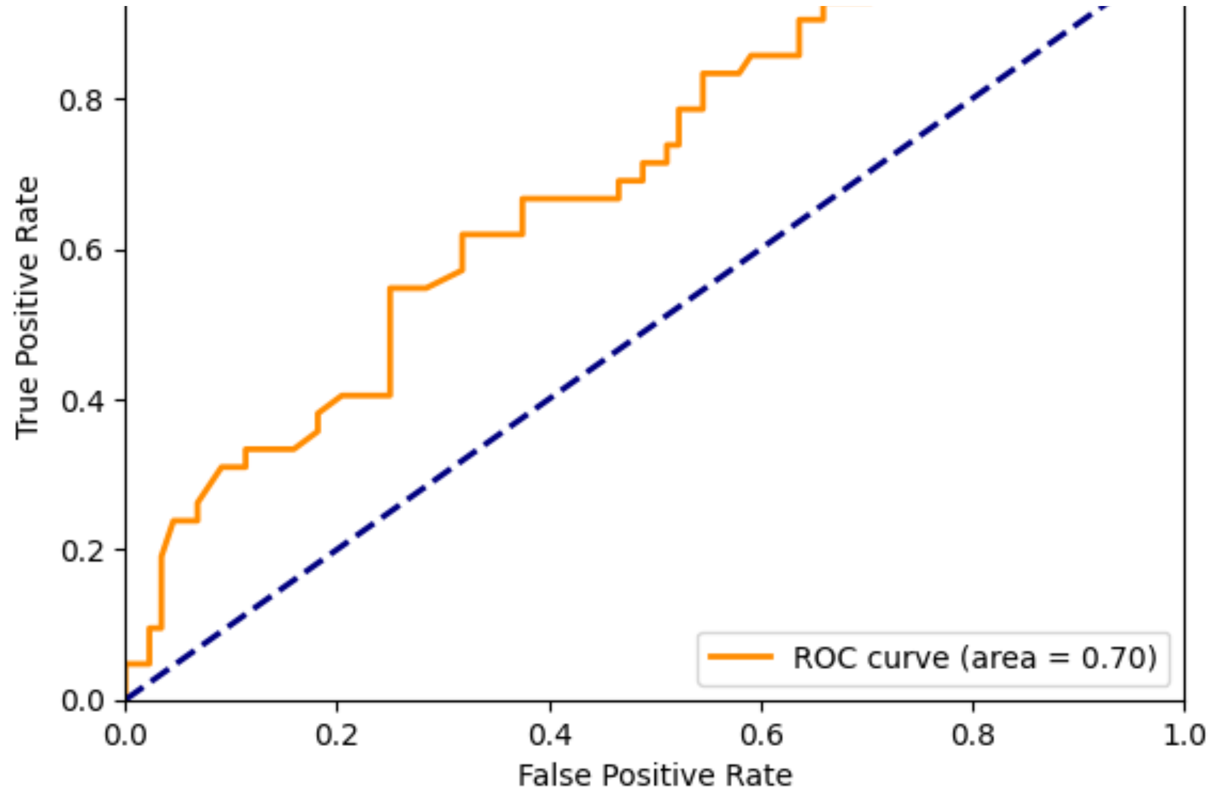
```
Training - Accuracy: 0.759, Sensitivity: 0.397, Specificity: 0.913, F1: 0.4

Test Metrics for manual threshold 0.35:
Accuracy: 0.662, Sensitivity: 0.405, Specificity: 0.784, F1: 0.436, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.47692307692307695, 'Sensitivity':
Threshold: 0.30, Metrics: {'Accuracy': 0.5846153846153846, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6615384615384615, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.45, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Decision Tree
```




Sex
−0.25
0.00
0.25
−0.25
0.00
0.25
SHAP interaction val

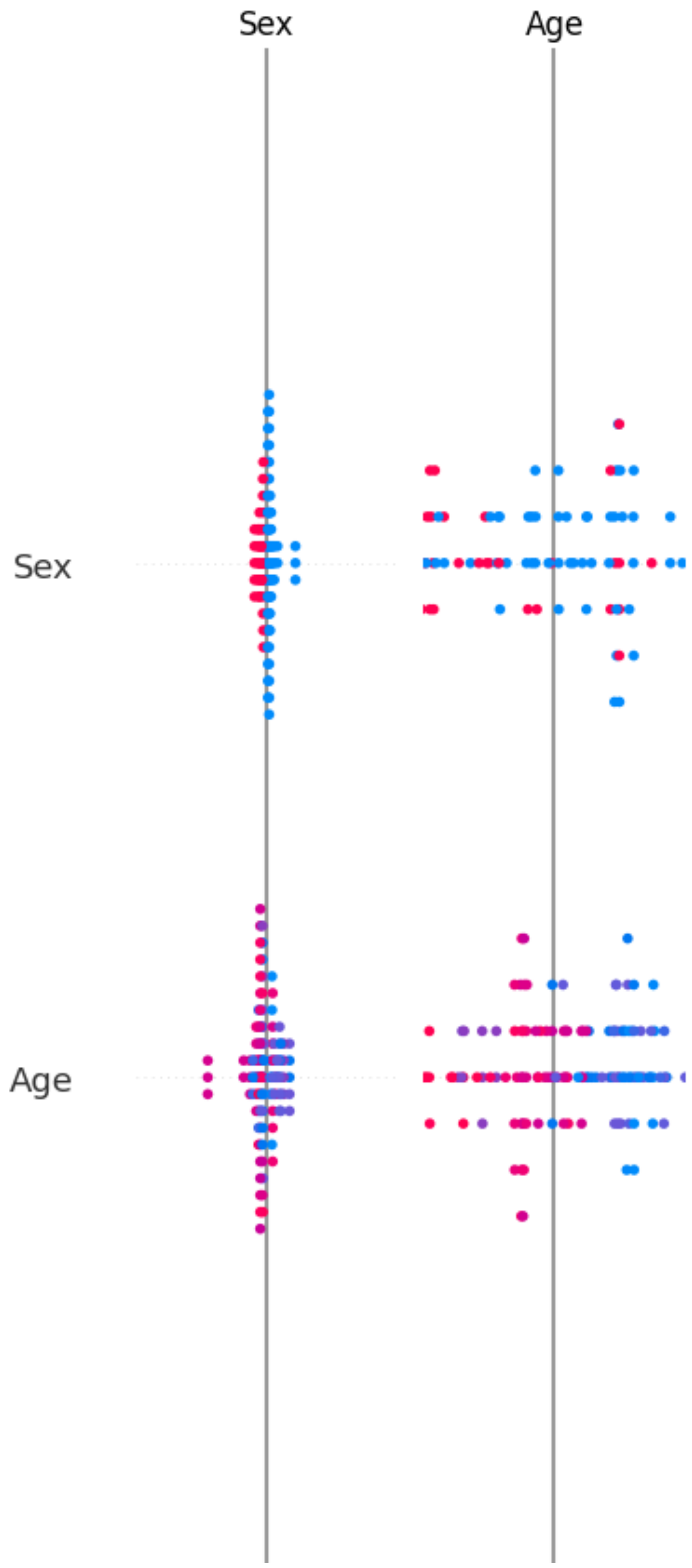

Confusion Matrix
Actual Success
Actual Failure
Predicted Success
Predicted Failure
69
19
25
17
60
50
40
30
20

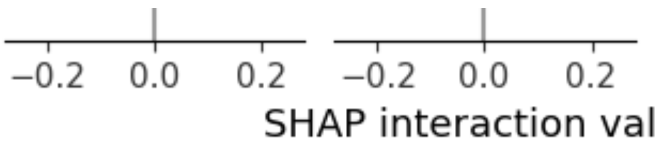

Receiver Operating Characteristic
1.0

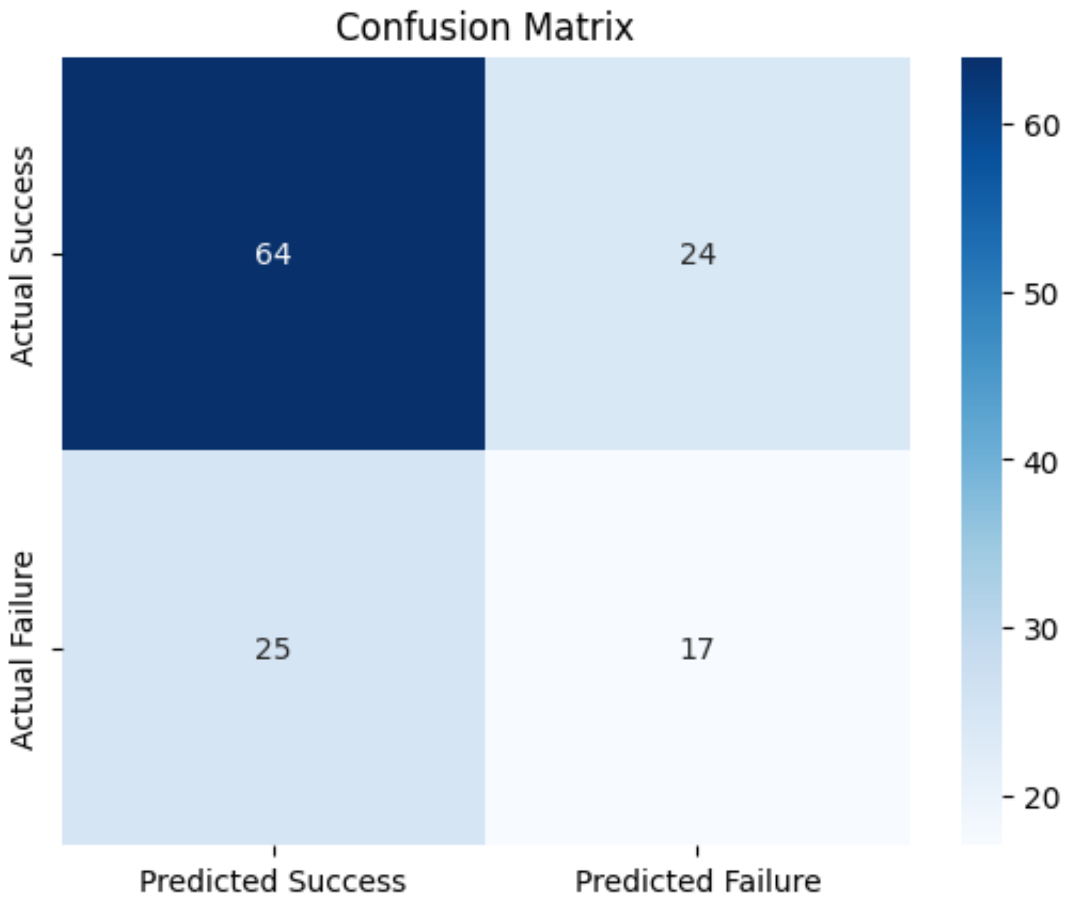


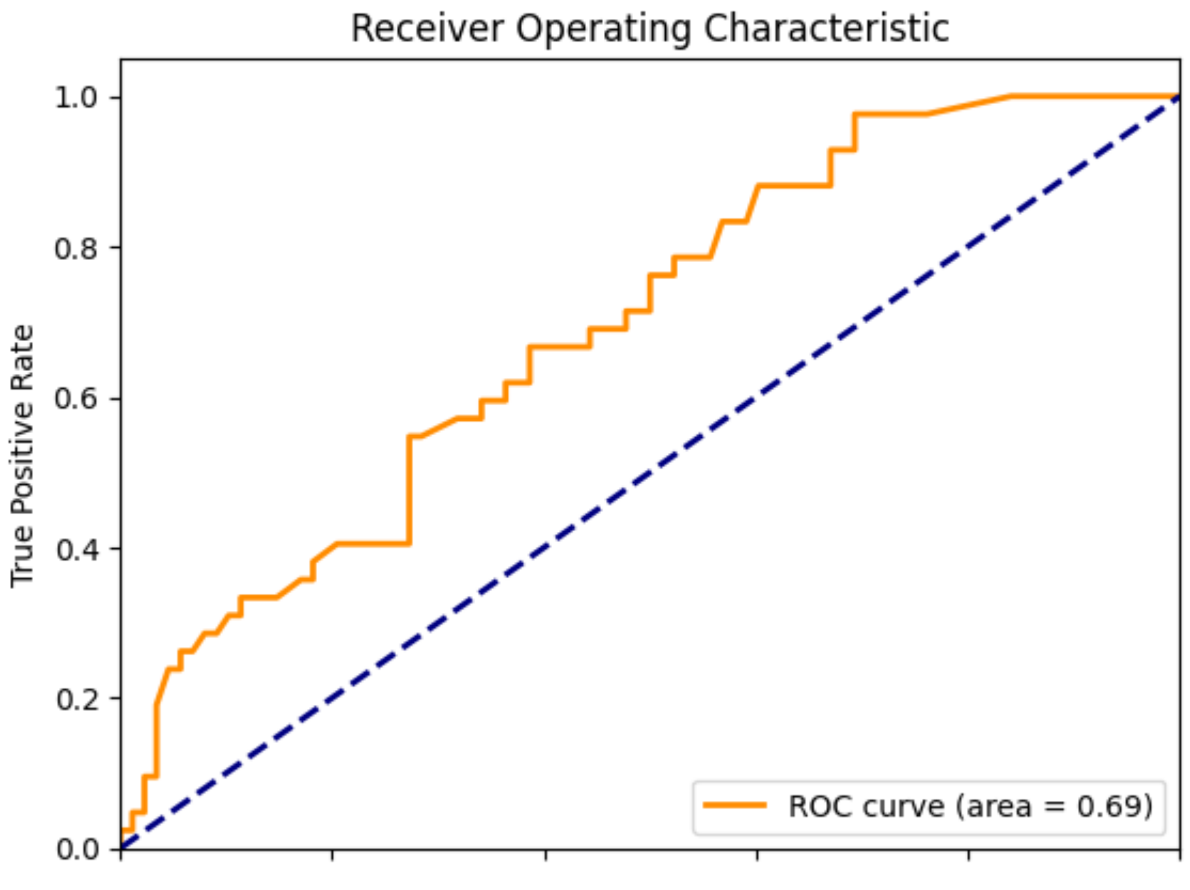
```
Running evaluation with seed 47

Evaluating Decision Tree with seed 47...
Best parameters for Decision Tree: {'ccp_alpha': 0.001, 'criterion': 'gini'

--- ROC Data for Copying ---
FPR = [0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.022727272727
TPR = [0.0, 0.023809523809523808, 0.023809523809523808, 0.04761904761904761
AUC = 0.6939935064935066
--- End of ROC Data ---

Training - Accuracy: 0.759, Sensitivity: 0.397, Specificity: 0.913, F1: 0.4

Test Metrics for manual threshold 0.35:
Accuracy: 0.623, Sensitivity: 0.405, Specificity: 0.727, F1: 0.410, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5230769230769231, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.5923076923076923, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6230769230769231, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7, 'Sensitivity': 0.14285714285714
Threshold: 0.45, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
```

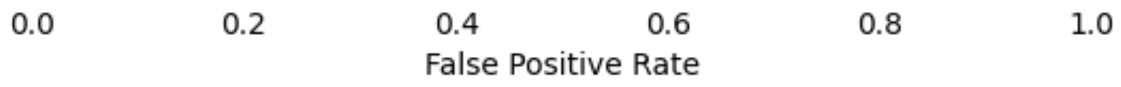
```
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Decision Tree
```



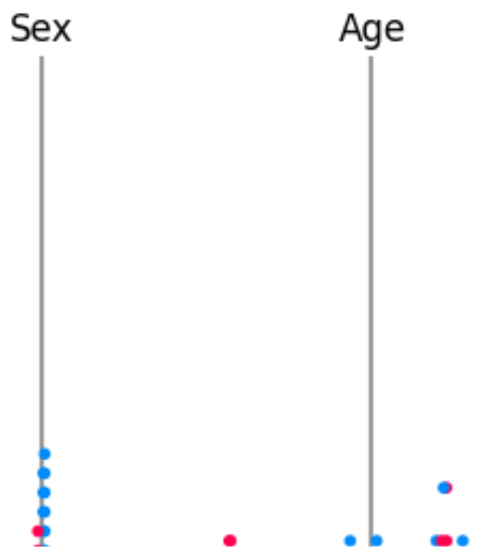

−0.2
0.0
0.2
−0.2
0.0
0.2
SHAP interaction val

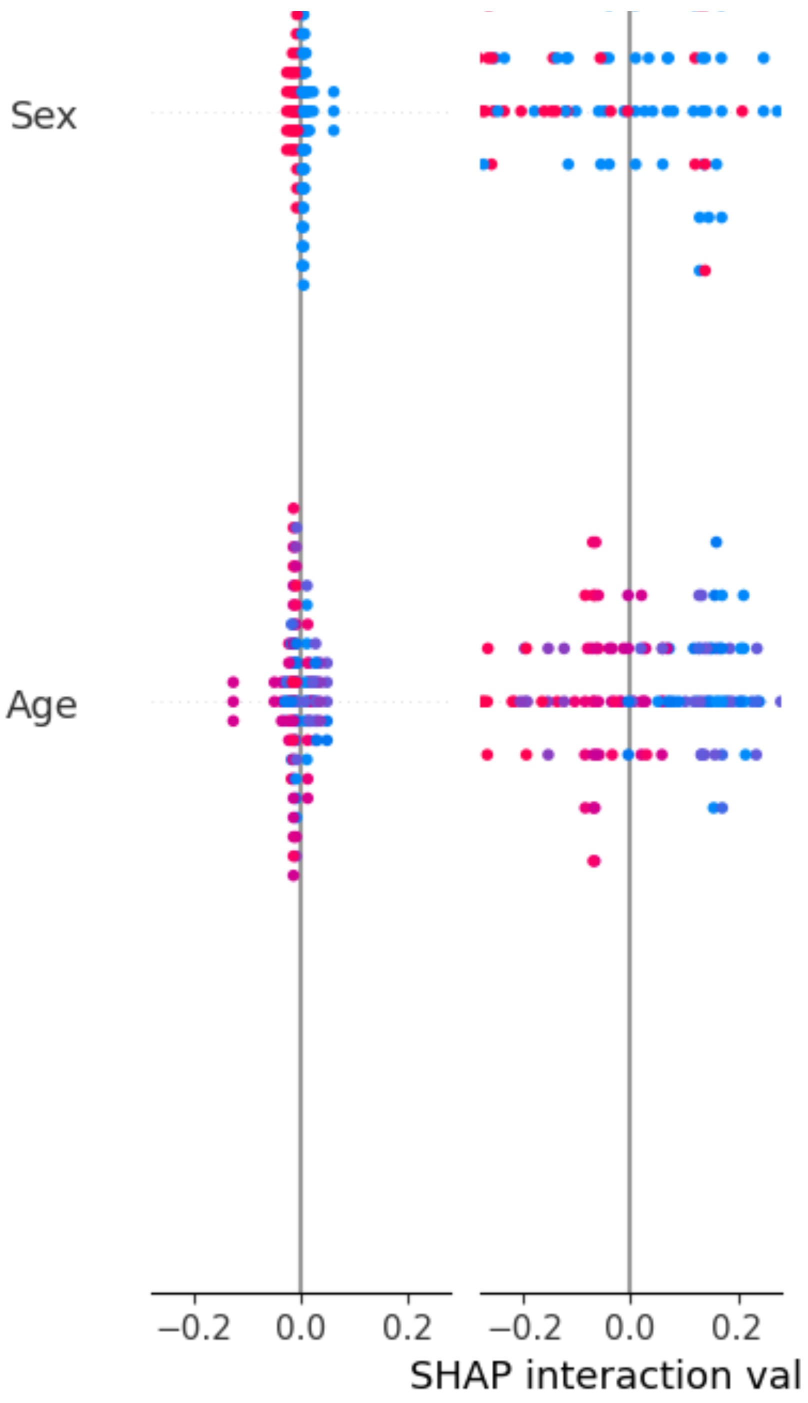

Confusion Matrix
Actual Success
Actual Failure
Predicted Success
Predicted Failure
64
24
25
17
60
50
40
30
20

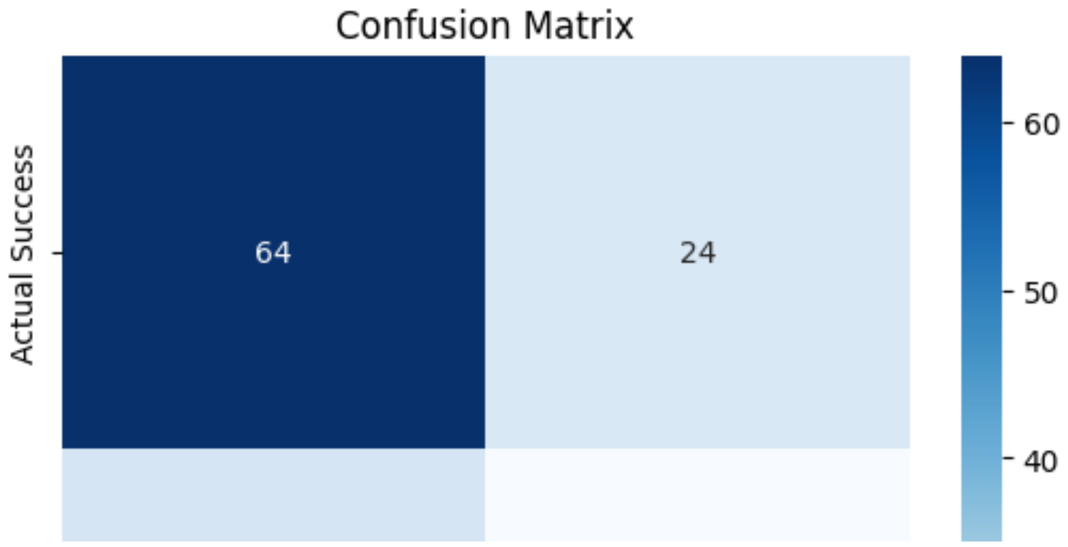

Receiver Operating Characteristic
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
ROC curve (area = 0.69)

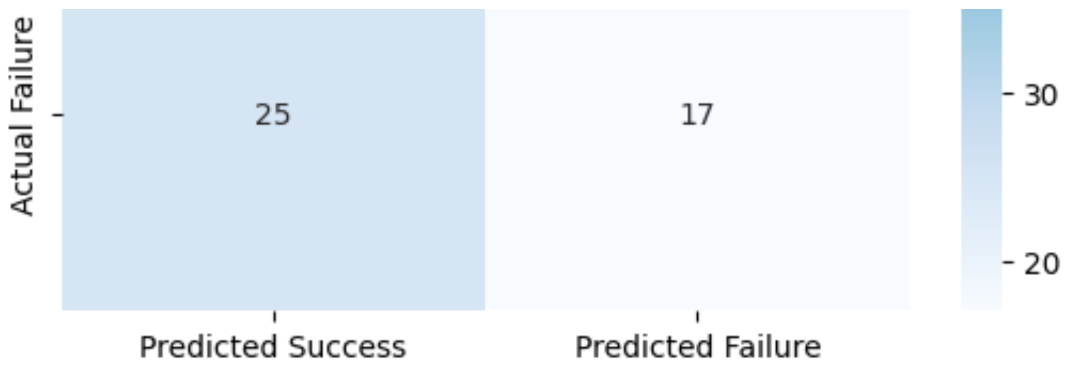


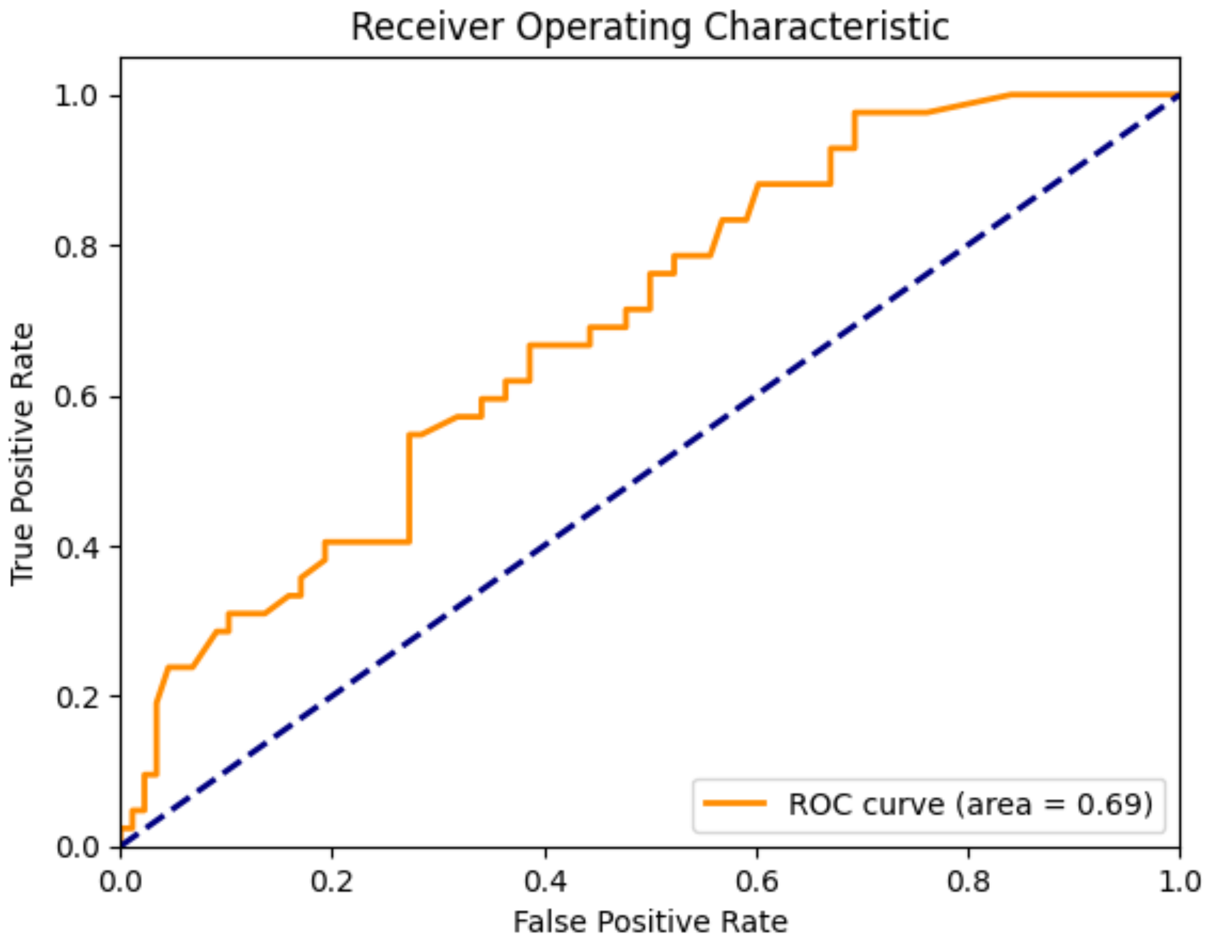
```
Running evaluation with seed 48

Evaluating Decision Tree with seed 48...
Best parameters for Decision Tree: {'ccp_alpha': 0.001, 'criterion': 'gini'

--- ROC Data for Copying ---
FPR = [0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.022727272727
TPR = [0.0, 0.023809523809523808, 0.023809523809523808, 0.04761904761904761
AUC = 0.6920995670995671
--- End of ROC Data ---

Training - Accuracy: 0.759, Sensitivity: 0.397, Specificity: 0.913, F1: 0.4

Test Metrics for manual threshold 0.35:
Accuracy: 0.623, Sensitivity: 0.405, Specificity: 0.727, F1: 0.410, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5230769230769231, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.5923076923076923, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6230769230769231, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7, 'Sensitivity': 0.14285714285714
Threshold: 0.45, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Decision Tree
```




Sex
Age
−0.2
0.0
0.2
−0.2
0.0
0.2
SHAP interaction val

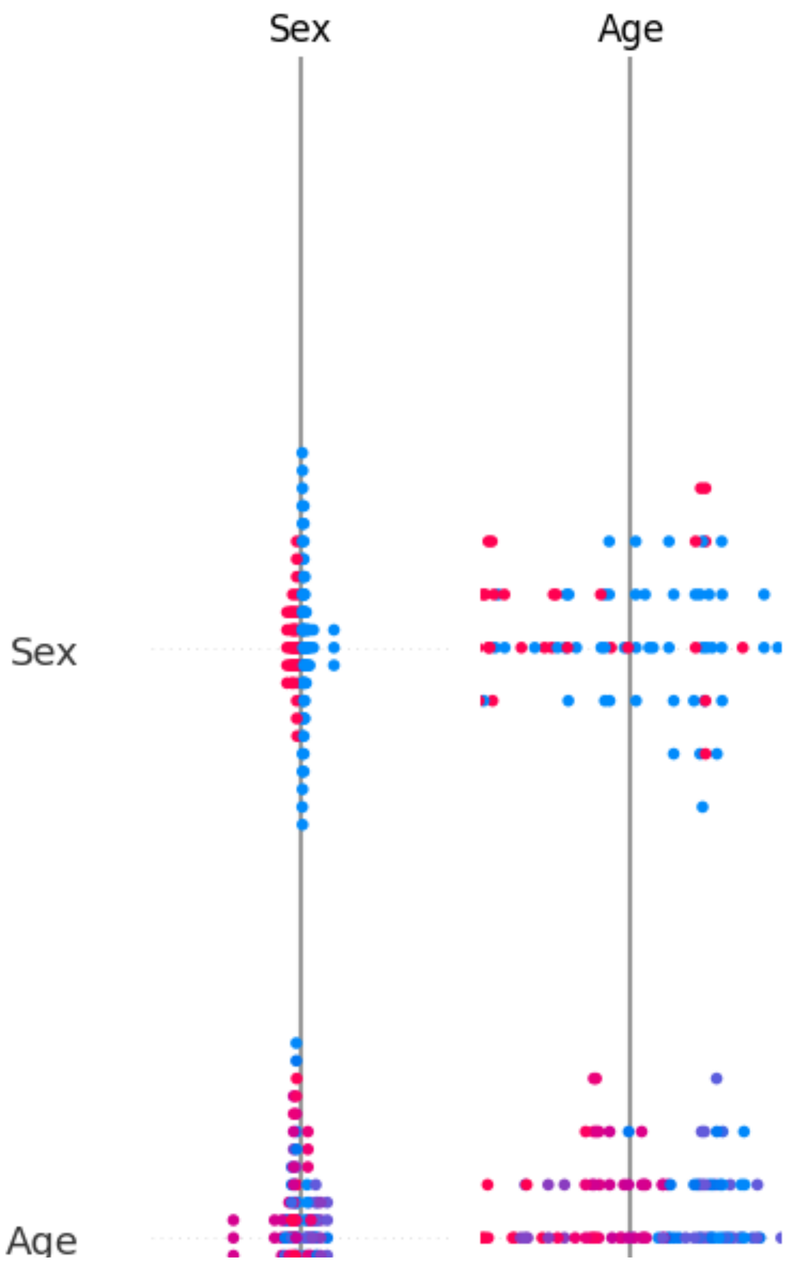

Confusion Matrix
Actual Success
64
24
60
50
40

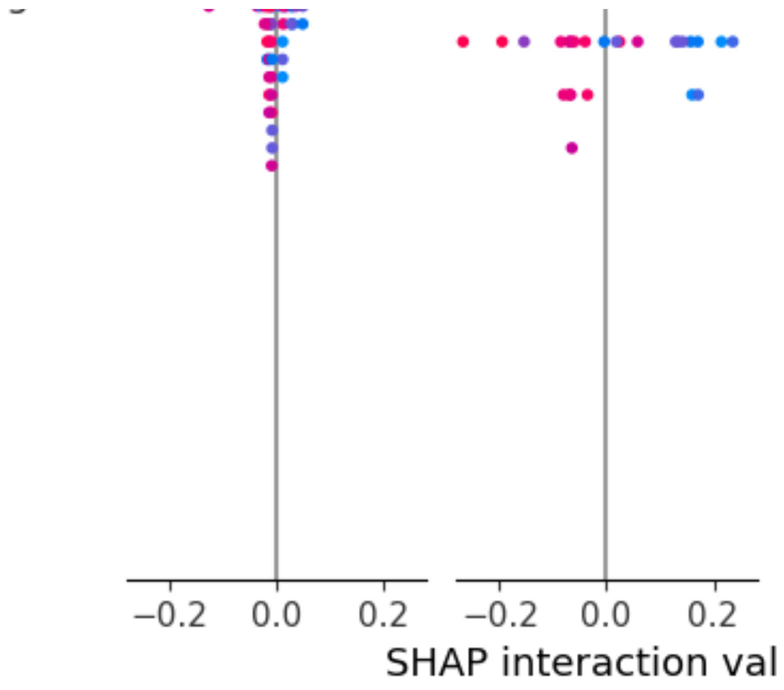


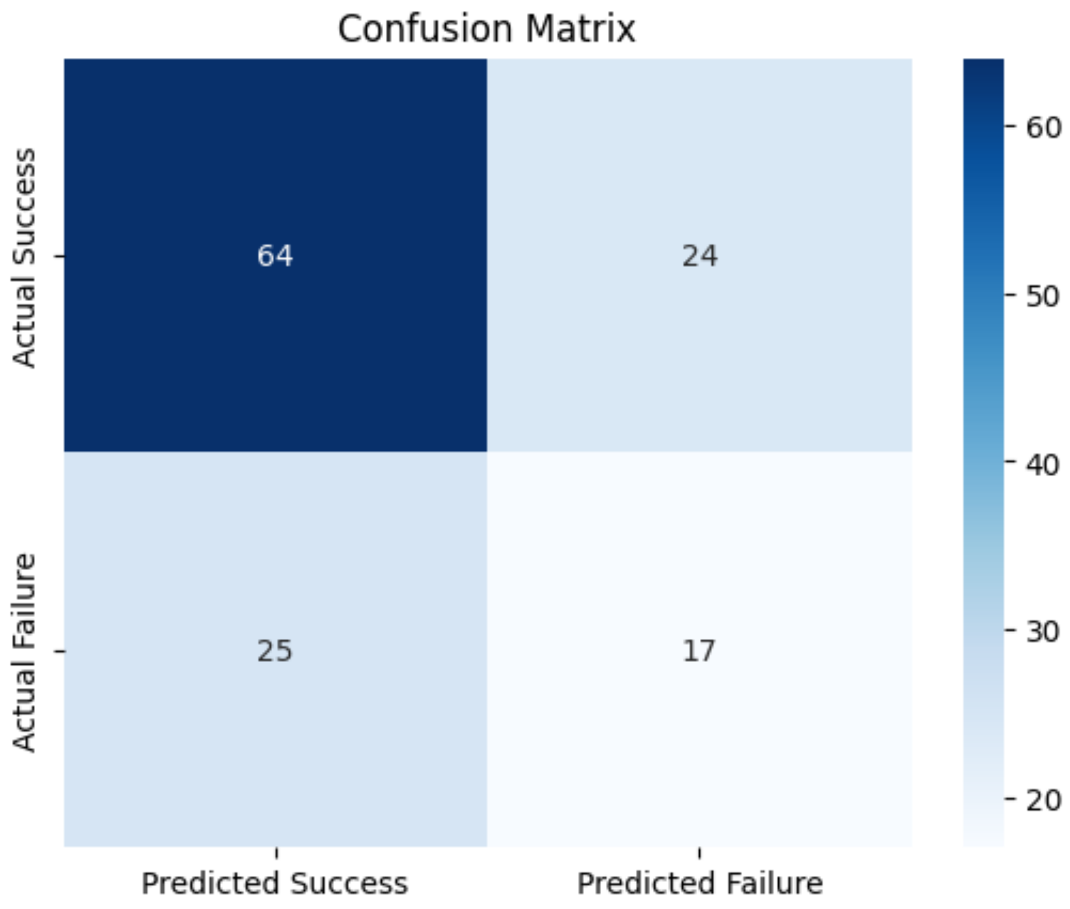


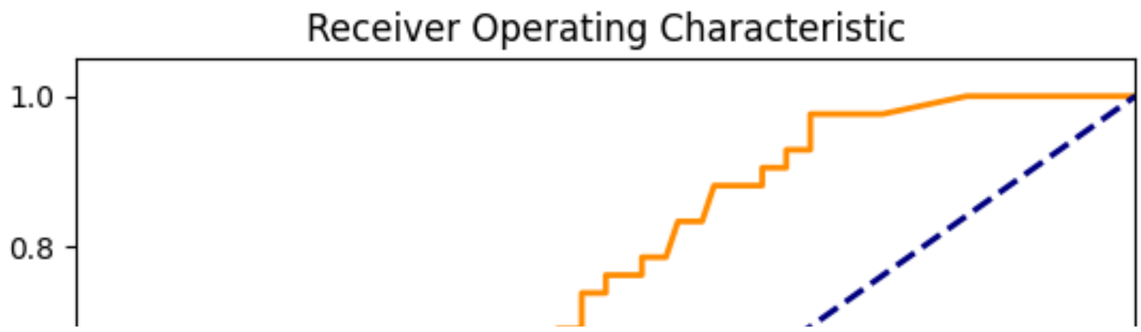
```
Running evaluation with seed 49

Evaluating Decision Tree with seed 49...
Best parameters for Decision Tree: {'ccp_alpha': 0.001, 'criterion': 'gini'

--- ROC Data for Copying ---
FPR = [0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.022727272727
TPR = [0.0, 0.023809523809523808, 0.023809523809523808, 0.047619047619047610
AUC = 0.6902056277056277
--- End of ROC Data ---

Training - Accuracy: 0.759, Sensitivity: 0.397, Specificity: 0.913, F1: 0.4

Test Metrics for manual threshold 0.35:
Accuracy: 0.623, Sensitivity: 0.405, Specificity: 0.727, F1: 0.410, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
```

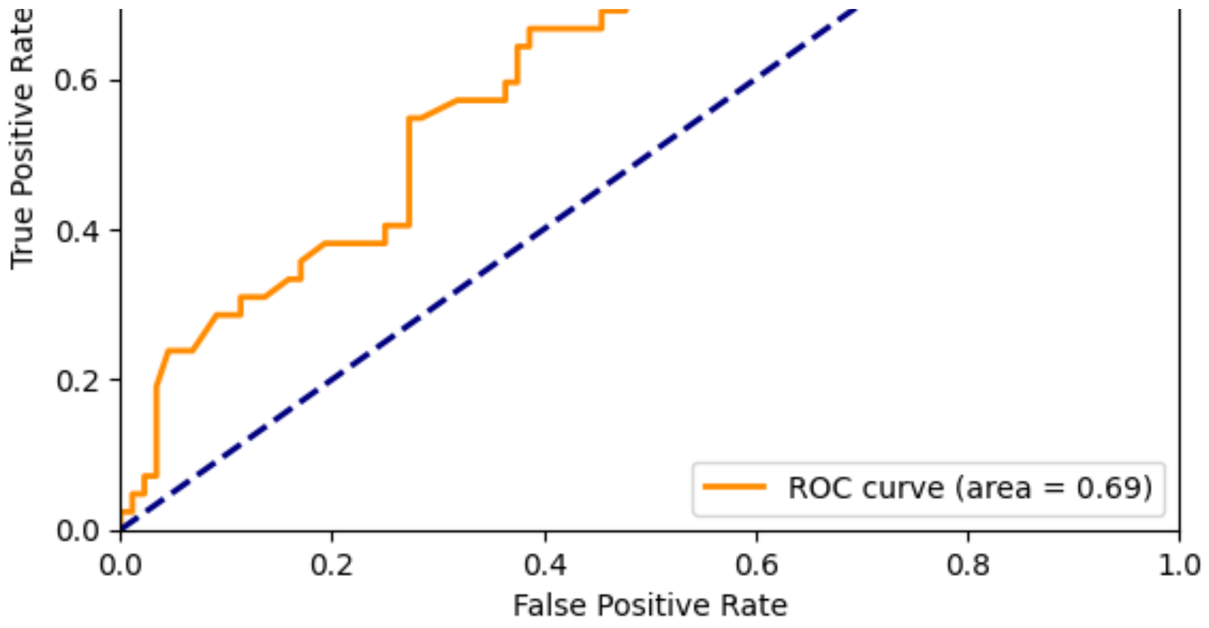
```
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5076923076923077, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.5923076923076923, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6230769230769231, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7, 'Sensitivity': 0.14285714285714
Threshold: 0.45, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Decision Tree
```




−0.2
0.0
0.2
−0.2
0.0
0.2
SHAP interaction val


Confusion Matrix
Actual Success
Actual Failure
Predicted Success
Predicted Failure
64
24
25
17
60
50
40
30
20


Receiver Operating Characteristic
1.0
0.8



```
Aggregated Test Set Metrics Across Seeds:
   accuracy  sensitivity  specificity        f1   roc_auc
0  0.623077     0.404762     0.727273  0.409639  0.691017
1  0.646154     0.380952     0.772727  0.410256  0.693047
2  0.661538     0.404762     0.784091  0.435897  0.696023
3  0.623077     0.404762     0.727273  0.409639  0.692100
4  0.653846     0.404762     0.772727  0.430380  0.696293
5  0.653846     0.404762     0.772727  0.430380  0.696293
6  0.661538     0.404762     0.784091  0.435897  0.698187
7  0.623077     0.404762     0.727273  0.409639  0.693994
8  0.623077     0.404762     0.727273  0.409639  0.692100
9  0.623077     0.404762     0.727273  0.409639  0.690206

Summary of Test Set Metrics (Mean, Standard Error, 95% Confidence Interval)
Accuracy: Mean = 0.639, SE = 0.006, 95% CI = [0.627, 0.652]
Sensitivity: Mean = 0.402, SE = 0.002, 95% CI = [0.397, 0.408]
Specificity: Mean = 0.752, SE = 0.008, 95% CI = [0.733, 0.771]
F1: Mean = 0.419, SE = 0.004, 95% CI = [0.410, 0.428]
Roc_auc: Mean = 0.694, SE = 0.001, 95% CI = [0.692, 0.696]
```

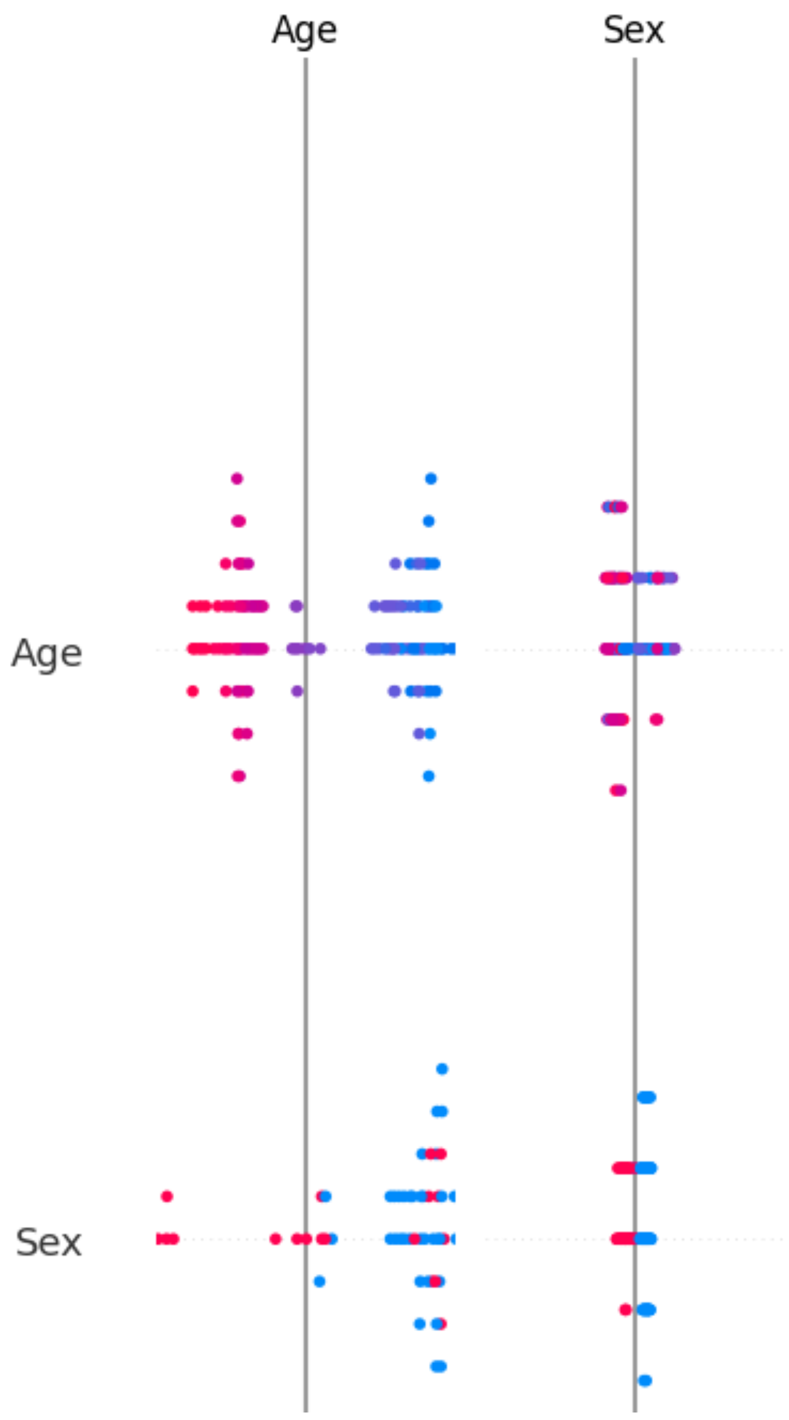
```
def evaluate_model(model, name, grid, X_train, y_train, X_test, y_test, cv, scc
    print(f"Evaluating {name} with seed {seed}...")

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

    clf = GridSearchCV(model, grid, cv=inner_cv, scoring='roc_auc')
    nested_scores = cross_validate(clf, X=X_train, y=y_train, cv=outer_cv, scor

    clf.fit(X_train, y_train)
    best_model = clf.best_estimator_
    best_params = clf.best_params_
```

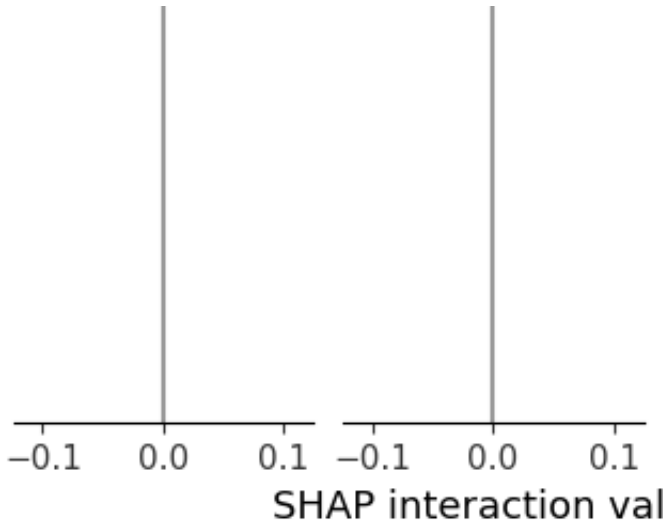
```
calibrated_clf = CalibratedClassifierCV(estimator=best_model, method='sigmo
calibrated_clf.fit(X_train, y_train)

y_probs = calibrated_clf.predict_proba(X_test)[:, 1]

# Calculate FPR, TPR, and AUC
fpr, tpr, thresholds = roc_curve(y_test, y_probs)
roc_auc = auc(fpr, tpr)

print("\n--- Dados ROC para copiar ---")
print("FPR =", fpr.tolist())
print("TPR =", tpr.tolist())
print("AUC =", roc_auc)
print("--- Fim dos Dados ROC ---\n")

# Calculate metrics for the training set
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}, Sensitivity: {train_sens}, Specif
      f"F1: {train_f1}, ROC AUC: {train_roc_auc}")

# 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"Metrics for manual threshold {manual_threshold}:")
print(f"Accuracy: {manual_acc}, Sensitivity: {manual_sens}, Specificity: {m
      f"F1: {manual_f1}, ROC AUC: {manual_roc_auc}")

# 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)
```

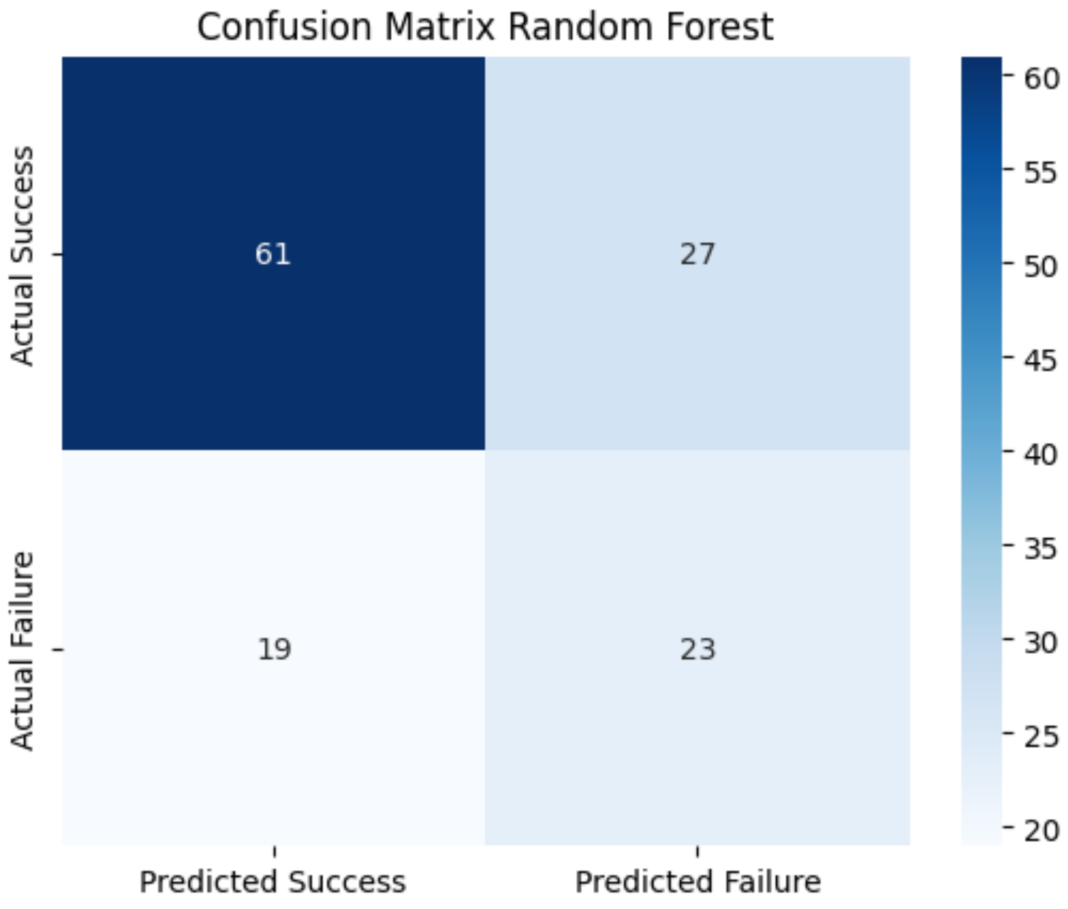
```
        threshold_metrics[threshold] = {
            'Accuracy': threshold_acc,
            'Sensitivity': threshold_sens,
            'Specificity': threshold_spec,
            'F1': threshold_f1,
            'ROC AUC': manual_roc_auc  # Same ROC AUC regardless of threshold
        }

    for threshold, metrics in threshold_metrics.items():
        print(f"Threshold: {threshold:.2f}, Metrics: {metrics}")

    calculate_and_plot_shap(best_model, X_train, X_test, name)

    # Prepare dictionary of test metrics for later 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, calibratec

def calculate_and_plot_shap(model, X_train, X_test, model_name):
    if isinstance(model, (RandomForestClassifier)):
        explainer = shap.TreeExplainer(model)
    else:
        explainer = shap.KernelExplainer(model.predict_proba, X_train.sample(10
    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 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()

def evaluate_random_forest(X_train, y_train, X_test, y_test, cv, scoring, manua
    model = RandomForestClassifier(n_jobs=-1, random_state=seed)
    grid = {
        'n_estimators': [500],
        'max_depth': [5],
        'min_samples_split': [2],
        'min_samples_leaf': [6],
        'max_features': ['sqrt'],
    }
    return evaluate_model(model, "Random Forest", grid, X_train, y_train, X_tes

def main(X_train, y_train, 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 = 0.35
    threshold_list = np.arange(0.1, 1.05, 0.05)

    aggregated_metrics = []

    for seed in range(40, 50):
        print(f"Running evaluation with seed {seed}")
        (best_model, manual_threshold, best_params, nested_scores, calibrated_c
         threshold_metrics, test_metrics) = evaluate_random_forest(X_train, y_t
                                                                    cv, scoring,

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

        plot_confusion_matrix(y_test, y_pred_manual)
        plot_roc_curve(y_test, y_probs)

        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
    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)
        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
    for metric, summary in metrics_summary.items():
        print(f"{metric.capitalize()}: Mean = {summary['Mean']:.3f}, SE = {summ
              f"95% CI = [{summary['95% CI'][0]:.3f}, {summary['95% CI'][1]:.3f

if __name__ == '__main__':
    main(X_train, y_train, X_test, y_test)
```

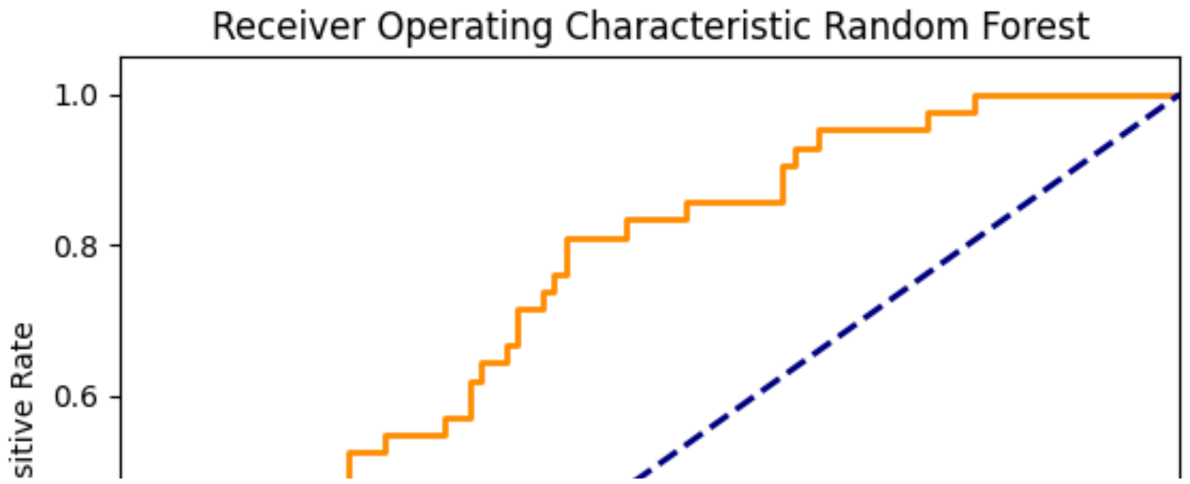
```
Running evaluation with seed 40
Evaluating Random Forest with seed 40...

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.03409090909090909, 0.03409090909090909, 0.034090909
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.726461038961039
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7218934911242604, Sensitivity: 0.06622516556291391,
Metrics for manual threshold 0.35:
Accuracy: 0.6461538461538462, Sensitivity: 0.5476190476190477, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.34615384615384615, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5615384615384615, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6538461538461539, 'Sensitivity': 0
```

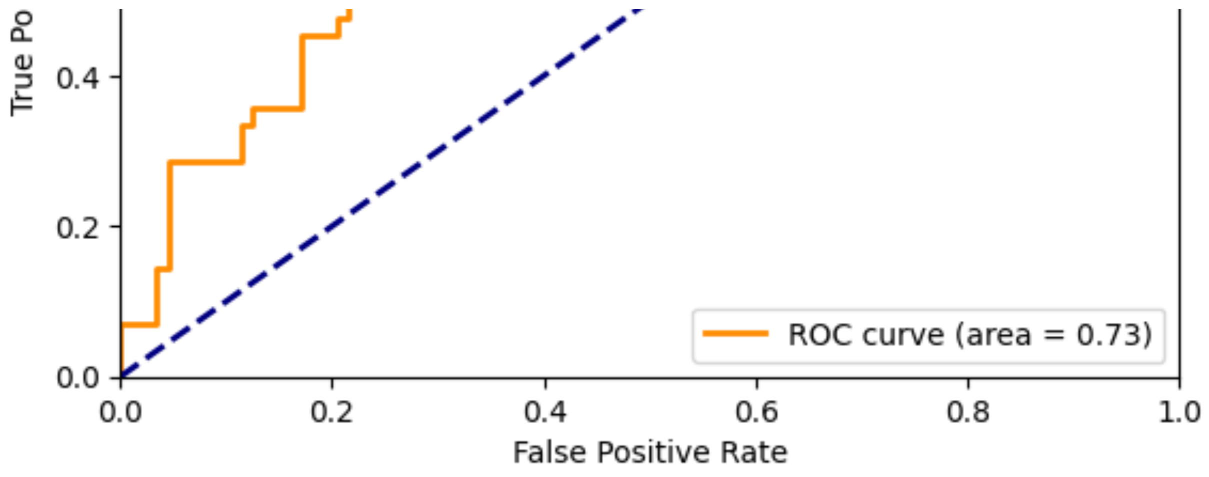
```
Threshold: 0.35, Metrics: {'Accuracy': 0.6461538461538462, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6923076923076923, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7076923076923077, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7, 'Sensitivity': 0.07142857142857
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Random Forest
```



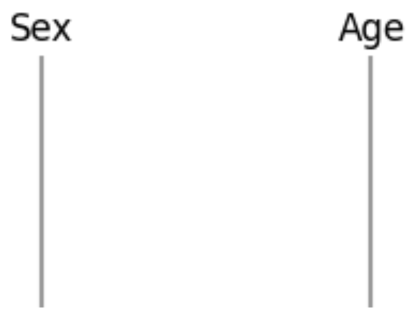

−0.1
0.0
0.1
−0.1
0.0
0.1
SHAP interaction val

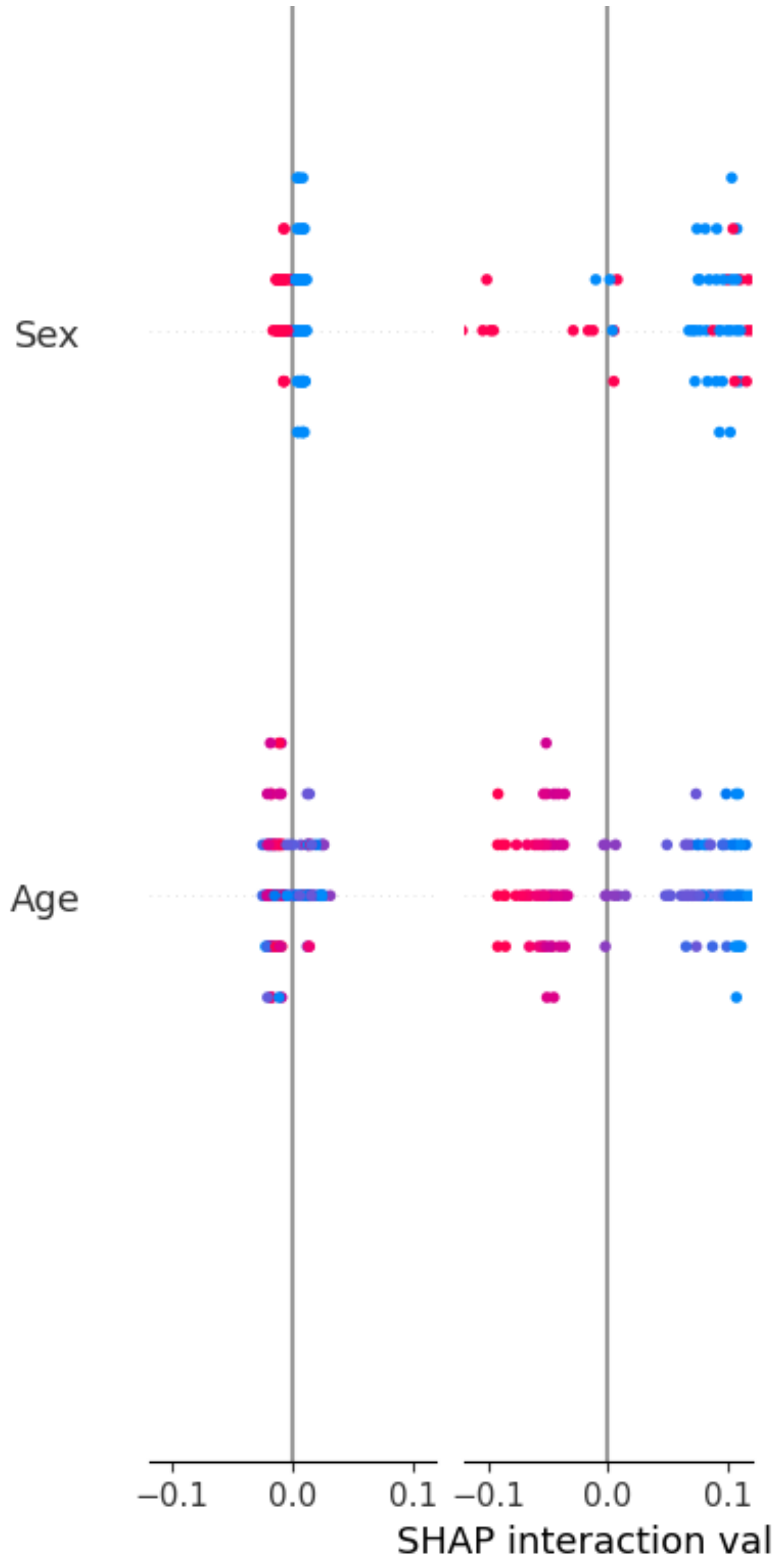

Confusion Matrix Random Forest
Actual Success
Actual Failure
Predicted Success
Predicted Failure
61
27
19
23
60
55
50
45
40
35
30
25
20

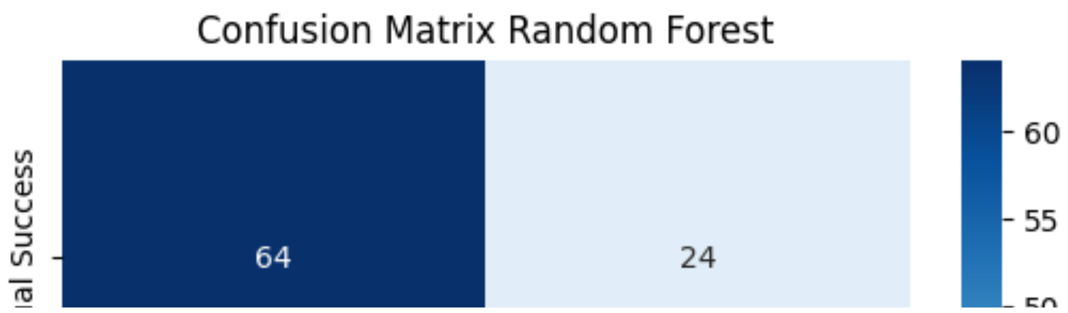

Receiver Operating Characteristic Random Forest
1.0
0.8
0.6
sitive Rate

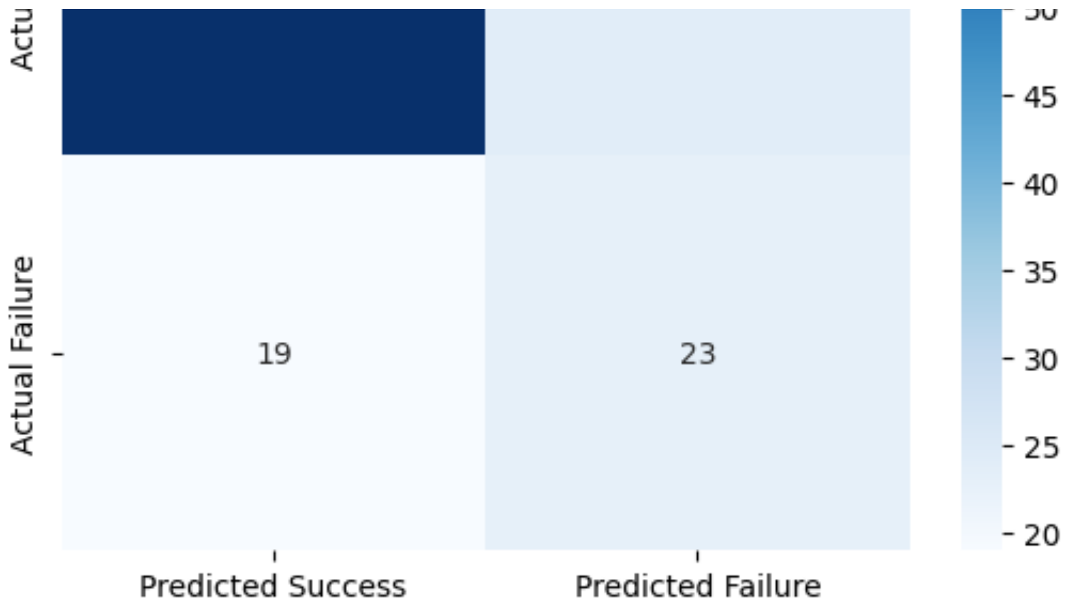


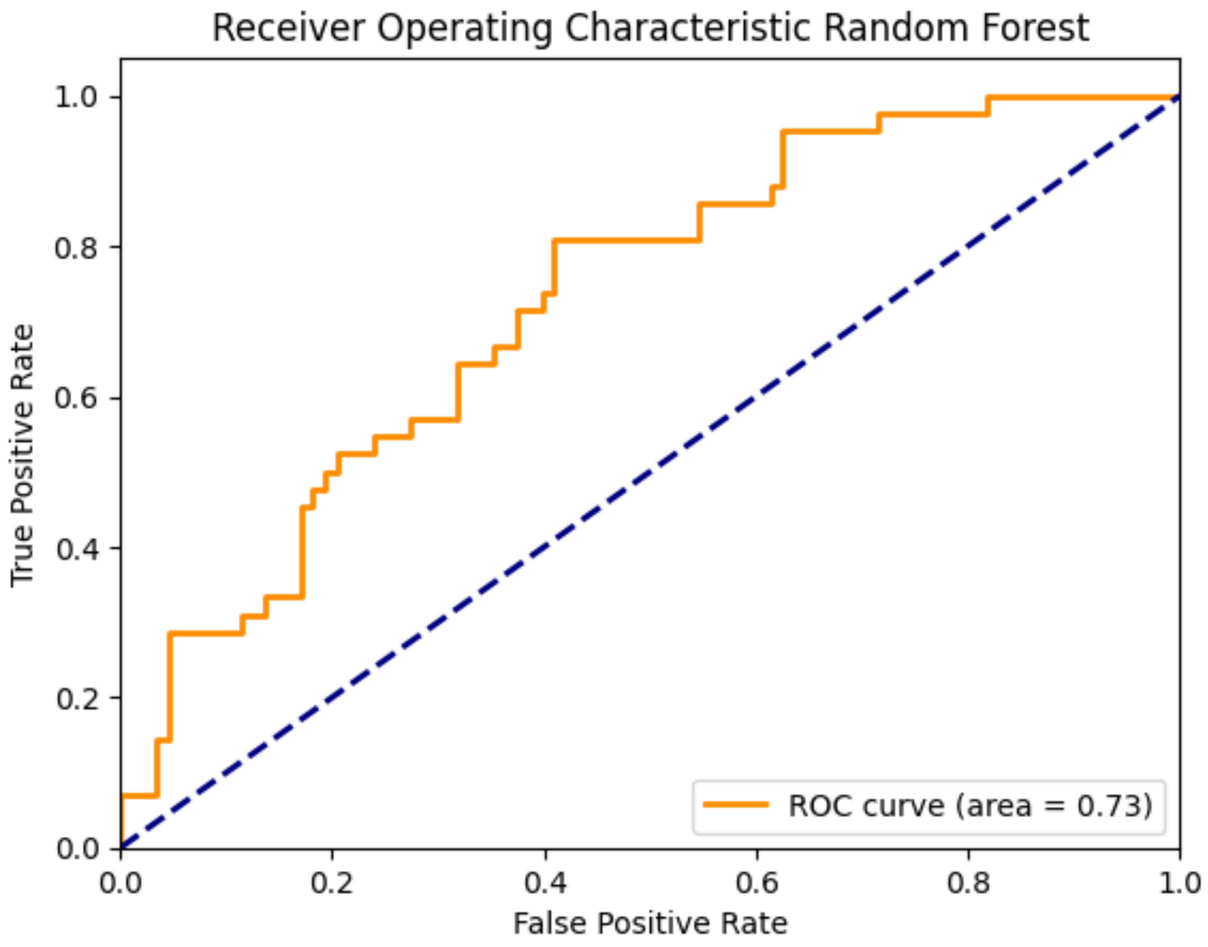
```
Running evaluation with seed 41
Evaluating Random Forest with seed 41...

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.03409090909090909, 0.03409090909090909, 0.034090909
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7294372294372294
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7258382642998028, Sensitivity: 0.07947019867549669,
Metrics for manual threshold 0.35:
Accuracy: 0.6692307692307692, Sensitivity: 0.5476190476190477, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.34615384615384615, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5461538461538461, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6461538461538462, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6692307692307692, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.45, Metrics: {'Accuracy': 0.7153846153846154, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7, 'Sensitivity': 0.07142857142857
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Random Forest
```




Sex
Age
−0.1
0.0
0.1
−0.1
0.0
0.1
SHAP interaction val

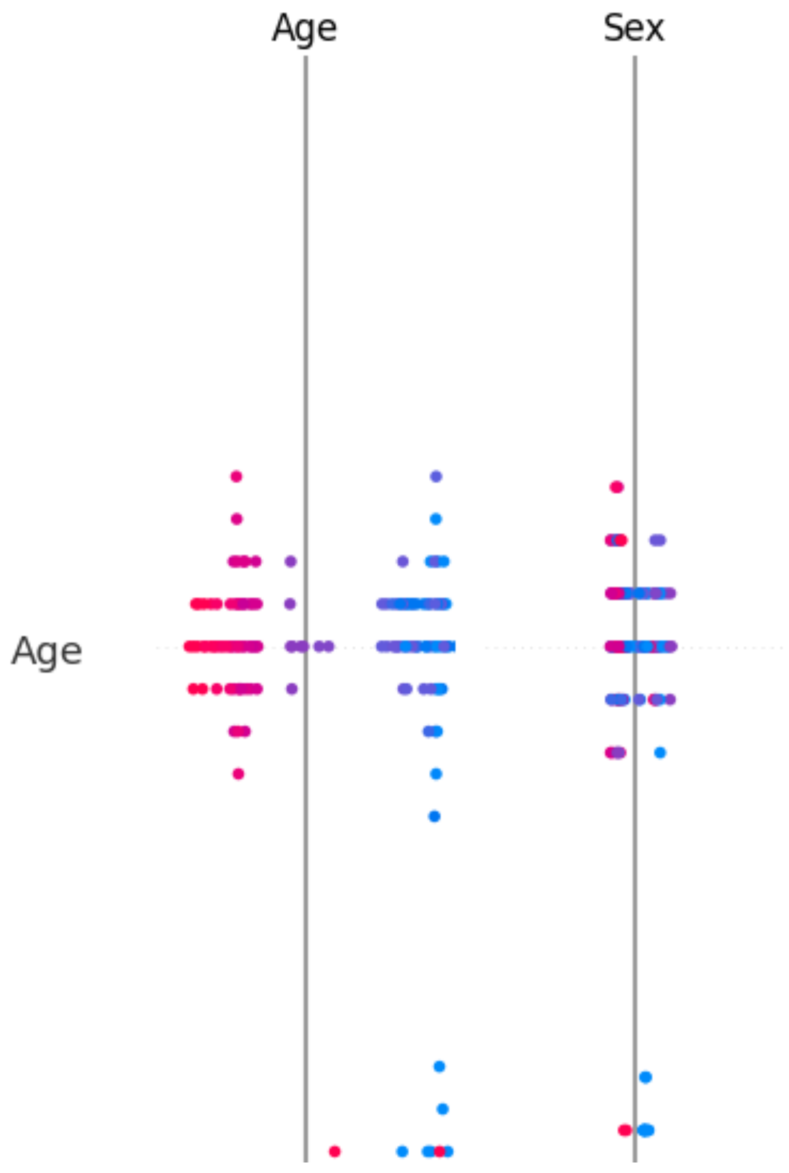

Confusion Matrix Random Forest
al Success
64
24
60
55
50

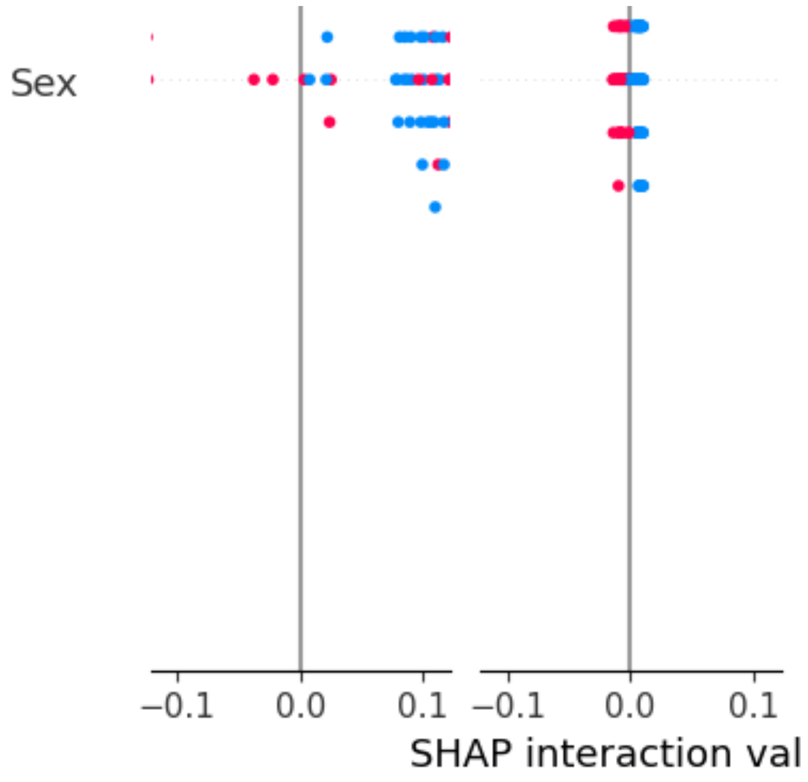


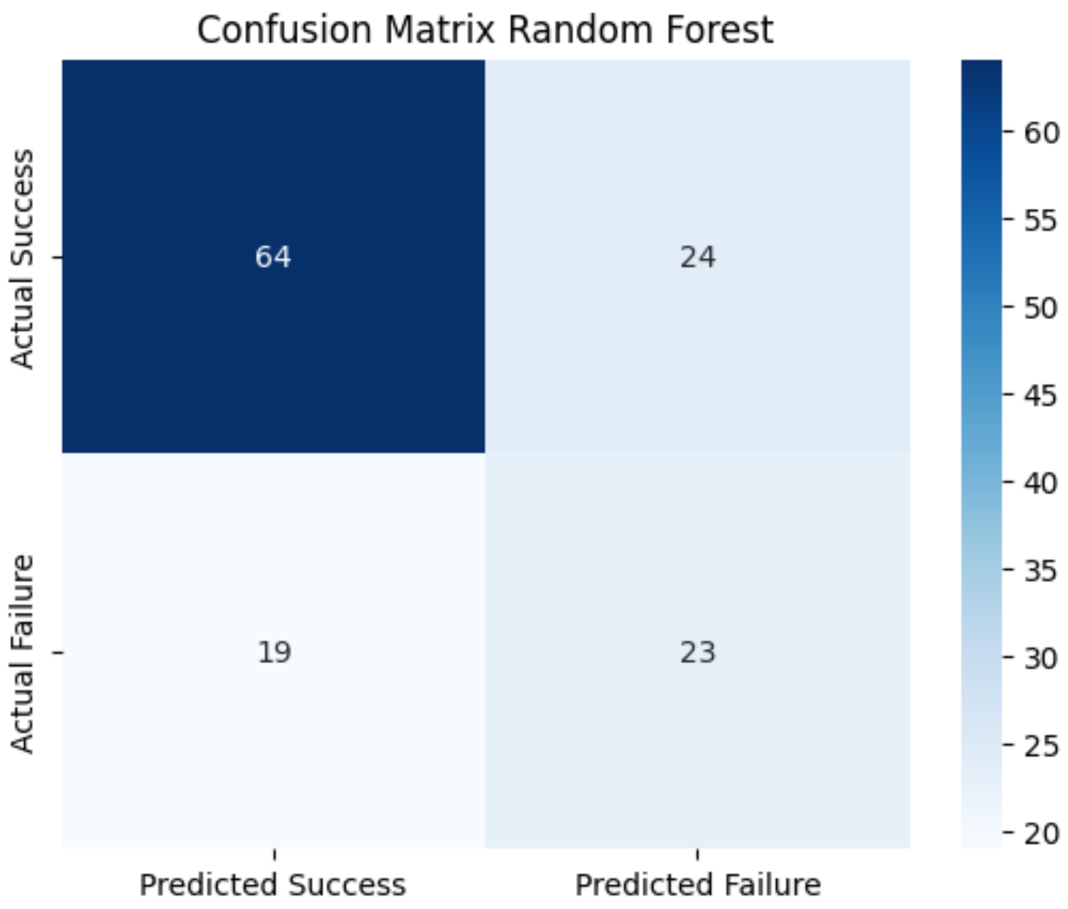


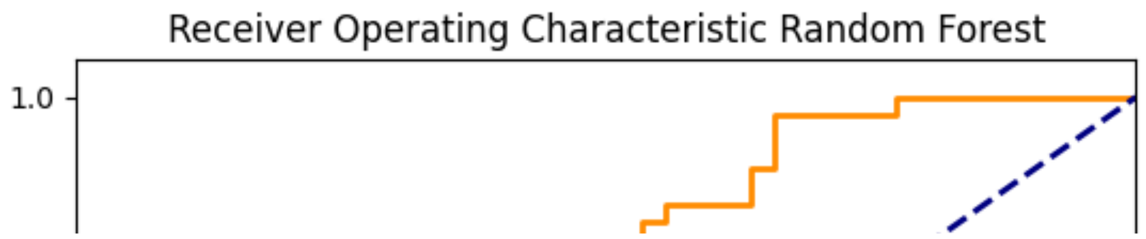
```
Running evaluation with seed 42
Evaluating Random Forest with seed 42...

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.03409090909090909, 0.03409090909090909, 0.034090909
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7229437229437229
--- Fim dos Dados ROC ---

Training - Accuracy: 0.727810650887574, Sensitivity: 0.08609271523178808, S
```

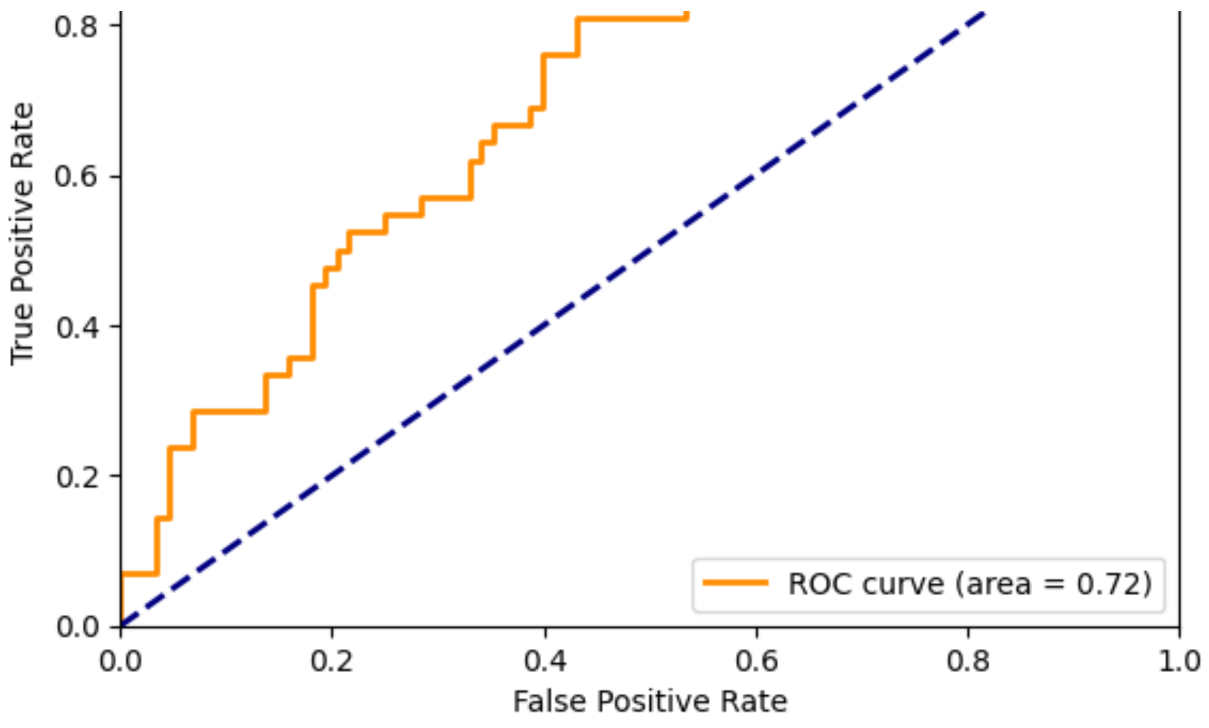
```
Metrics for manual threshold 0.35:
Accuracy: 0.6692307692307692, Sensitivity: 0.5476190476190477, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.33076923076923076, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5461538461538461, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6307692307692307, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6692307692307692, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6923076923076923, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7076923076923077, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.6923076923076923, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Random Forest
```



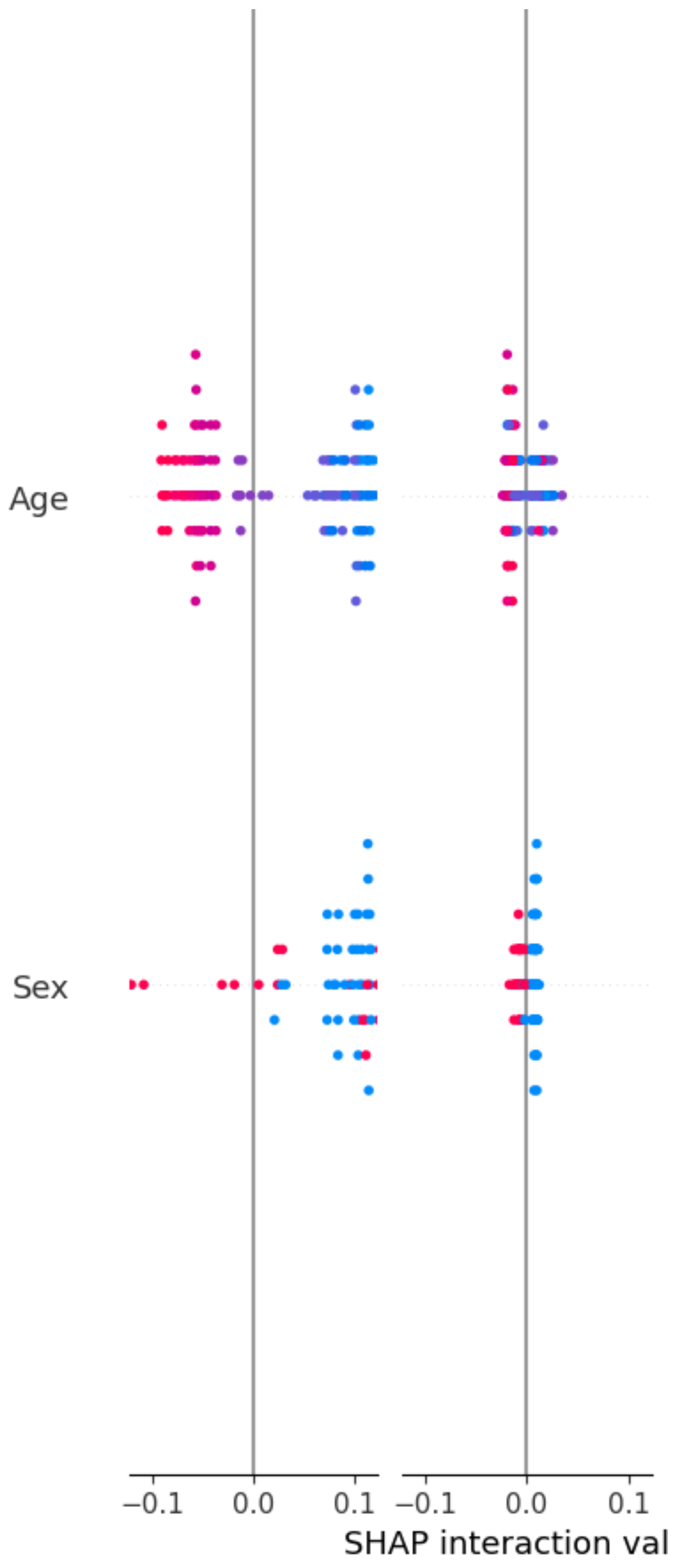

Sex
−0.1
0.0
0.1
−0.1
0.0
0.1
SHAP interaction val


Confusion Matrix Random Forest
Actual Success
Actual Failure
Predicted Success
Predicted Failure
64
24
19
23
60
55
50
45
40
35
30
25
20

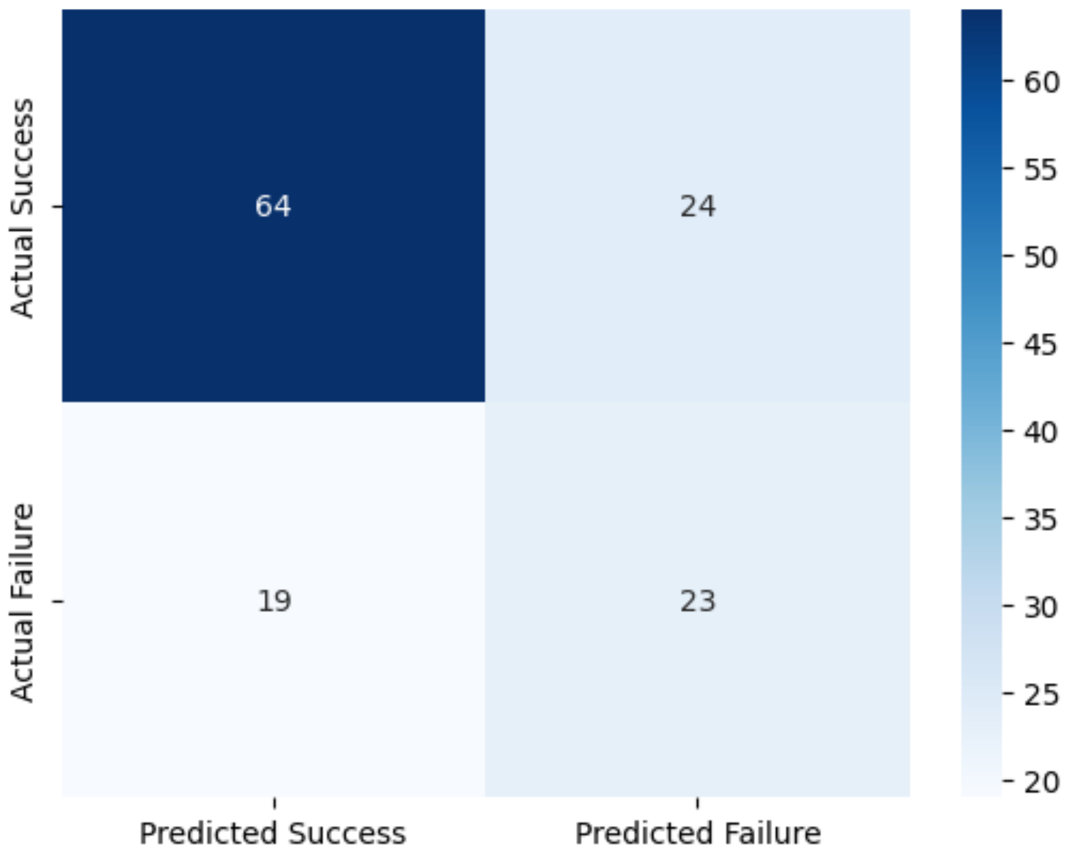

Receiver Operating Characteristic Random Forest
1.0

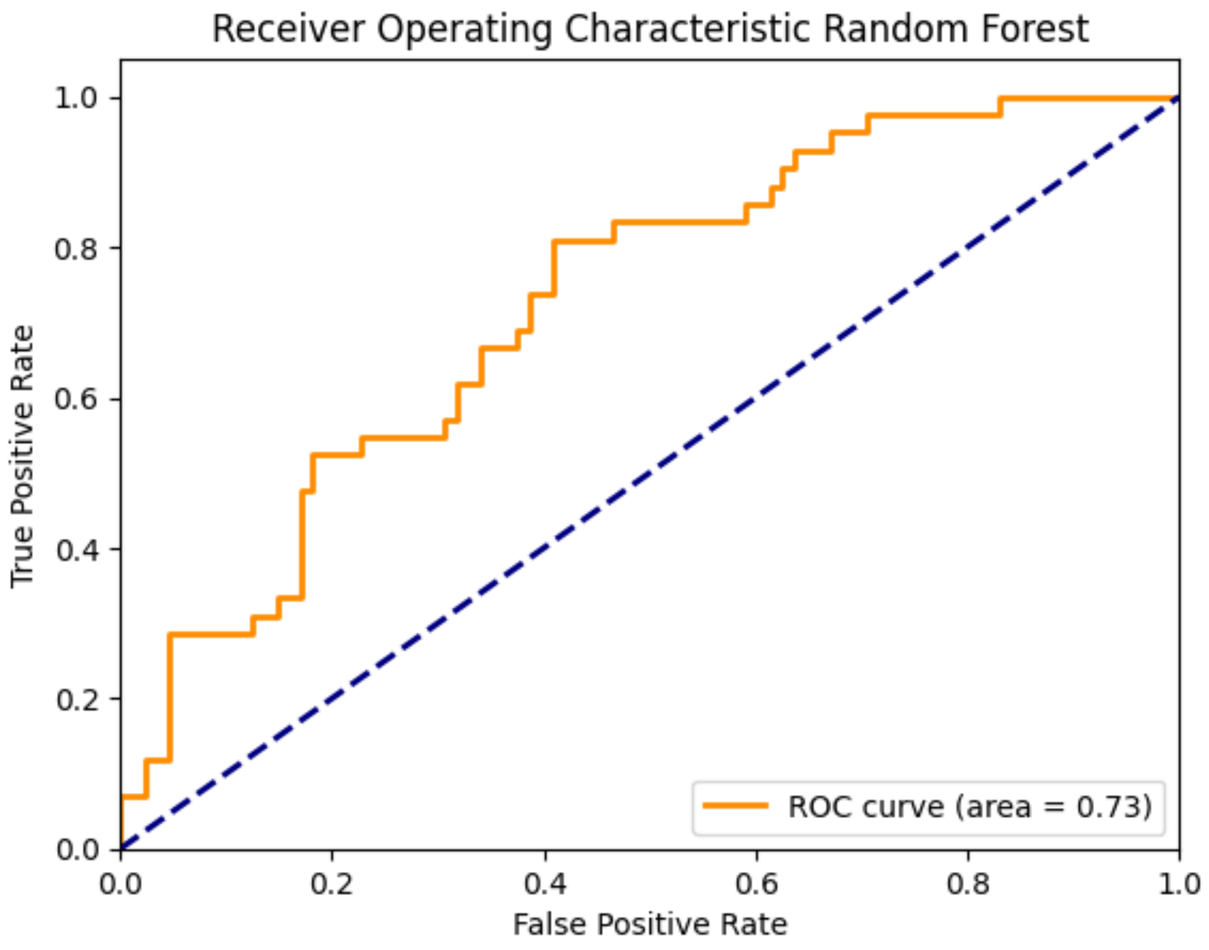


```
Running evaluation with seed 43
Evaluating Random Forest with seed 43...

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.0454545
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7288961038961039
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7258382642998028, Sensitivity: 0.07947019867549669,
Metrics for manual threshold 0.35:
Accuracy: 0.6692307692307692, Sensitivity: 0.5476190476190477, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.33076923076923076, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5461538461538461, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6384615384615384, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6692307692307692, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6692307692307692, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7076923076923077, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.6923076923076923, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Random Forest
```



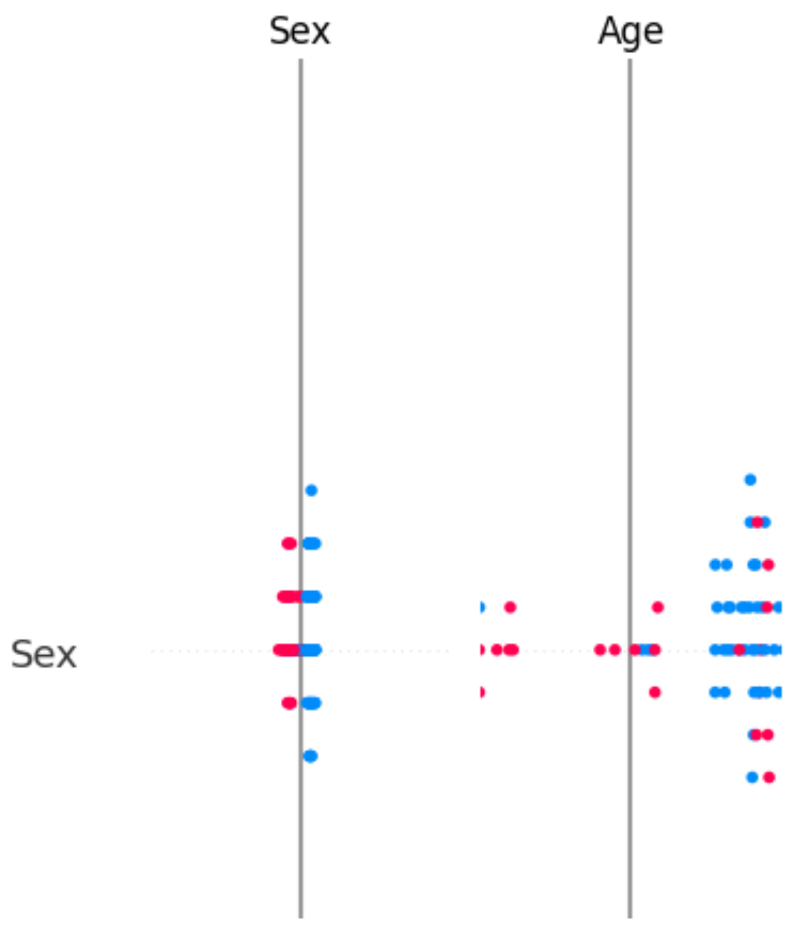
Confusion Matrix Random Forest

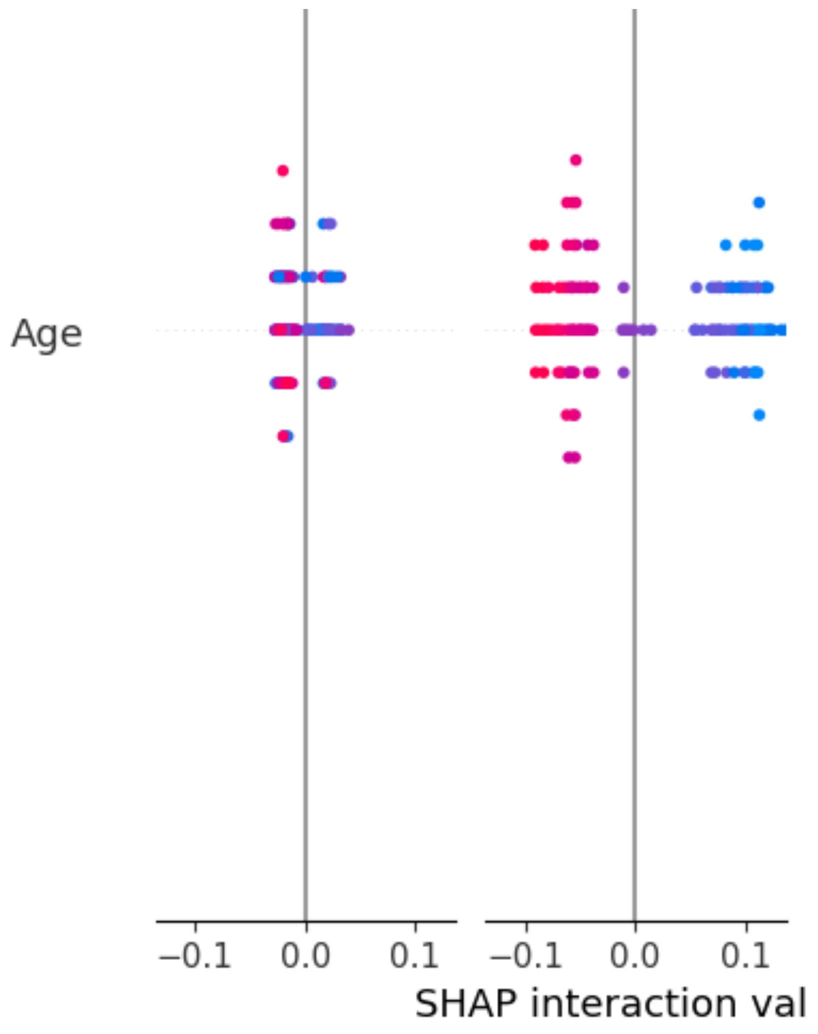


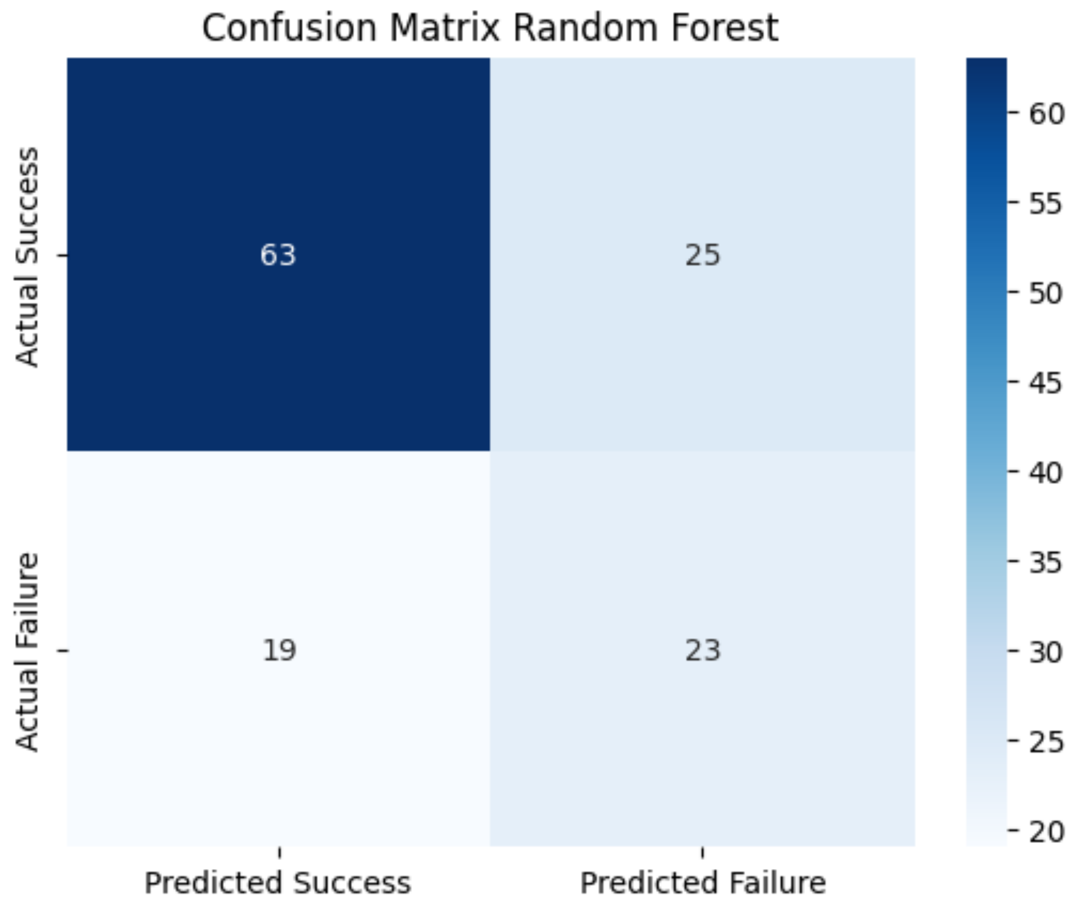


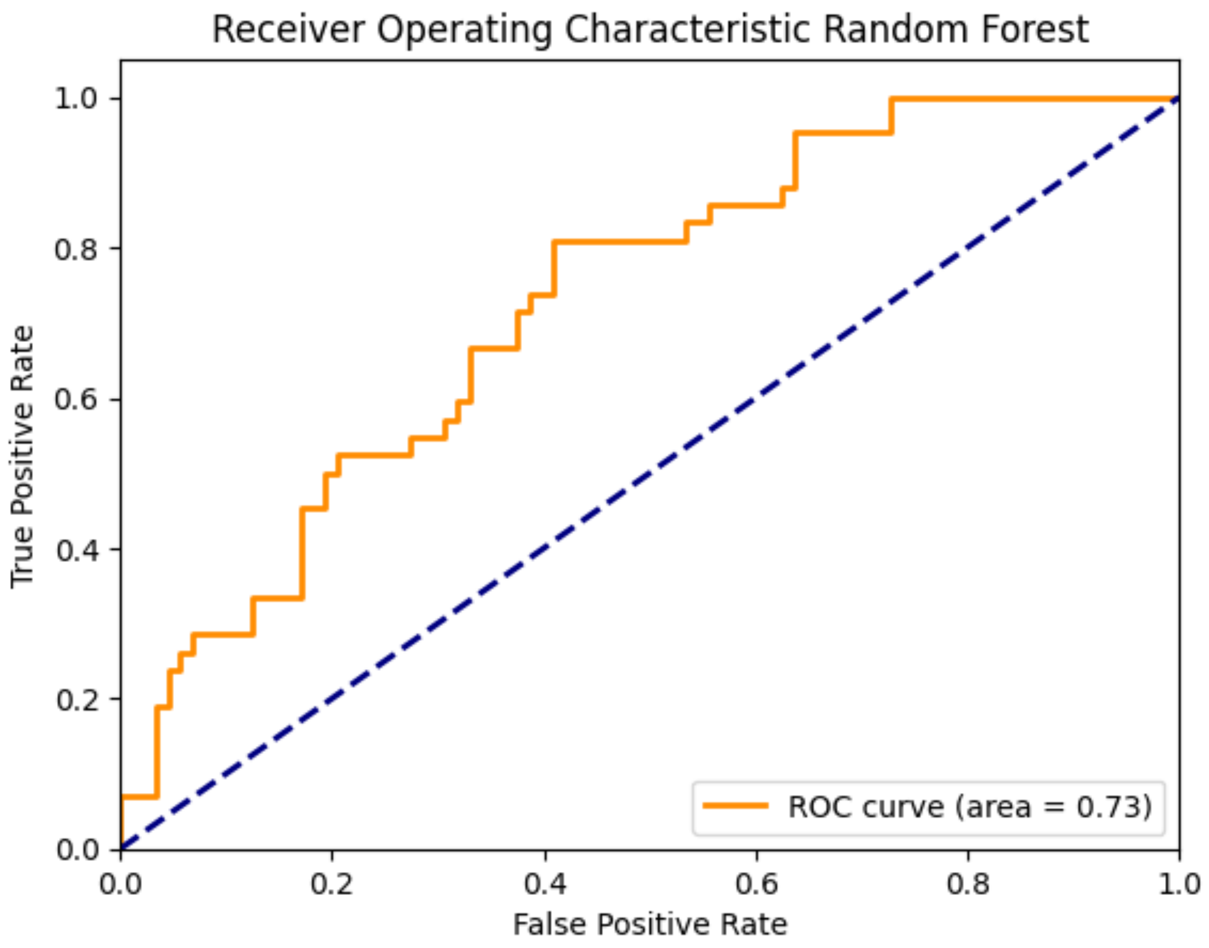
```
Running evaluation with seed 44
Evaluating Random Forest with seed 44...

    Dados ROC para copiar
```

```
--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.03409090909090909, 0.03409090909090909, 0.034090909
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7283549783549783
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7337278106508875, Sensitivity: 0.11920529801324503,
Metrics for manual threshold 0.35:
Accuracy: 0.6615384615384615, Sensitivity: 0.5476190476190477, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.34615384615384615, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5461538461538461, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6461538461538462, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6615384615384615, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7153846153846154, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7, 'Sensitivity': 0.07142857142857
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Random Forest
```



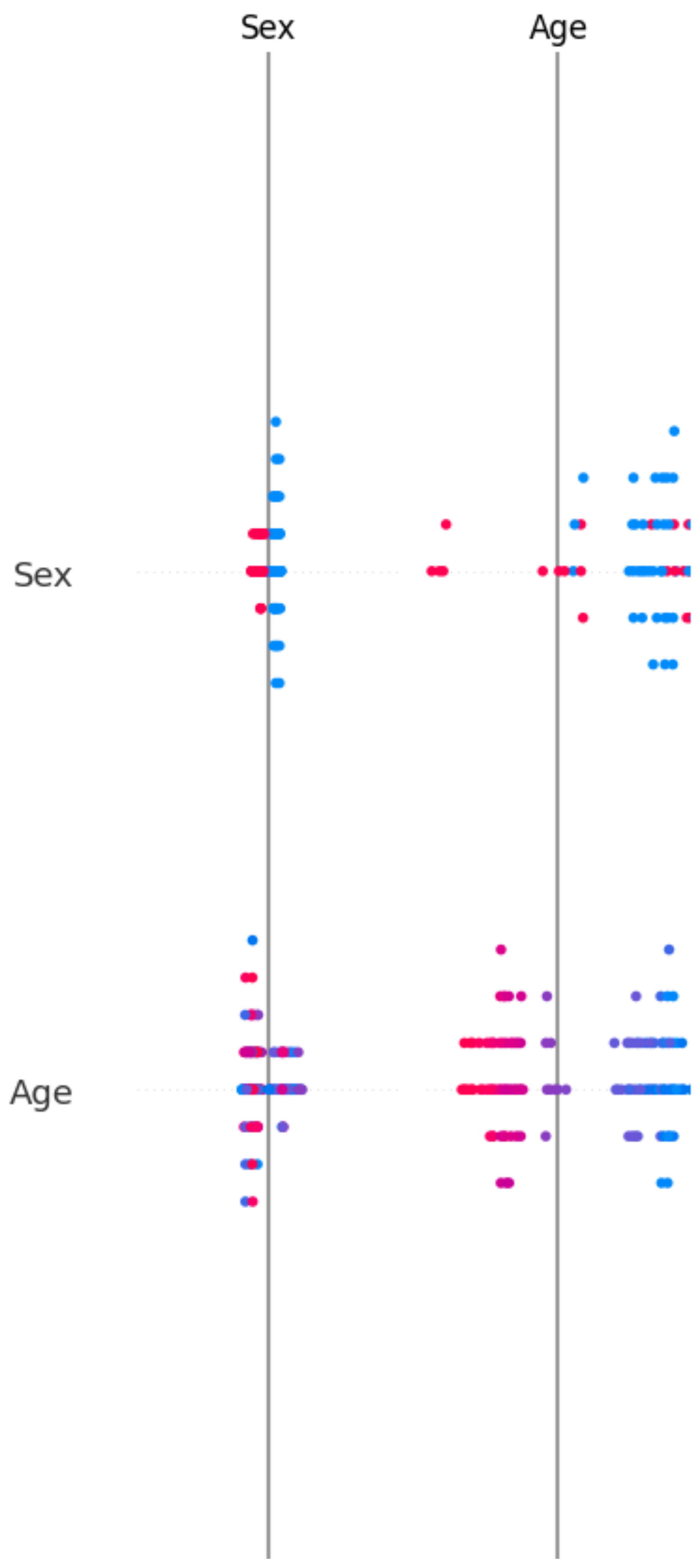

Age
−0.1
0.0
0.1
−0.1
0.0
0.1
SHAP interaction val

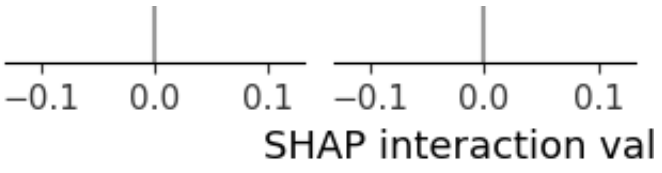

Confusion Matrix Random Forest
Actual Success
Actual Failure
Predicted Success
Predicted Failure
63
25
19
23
60
55
50
45
40
35
30
25
20

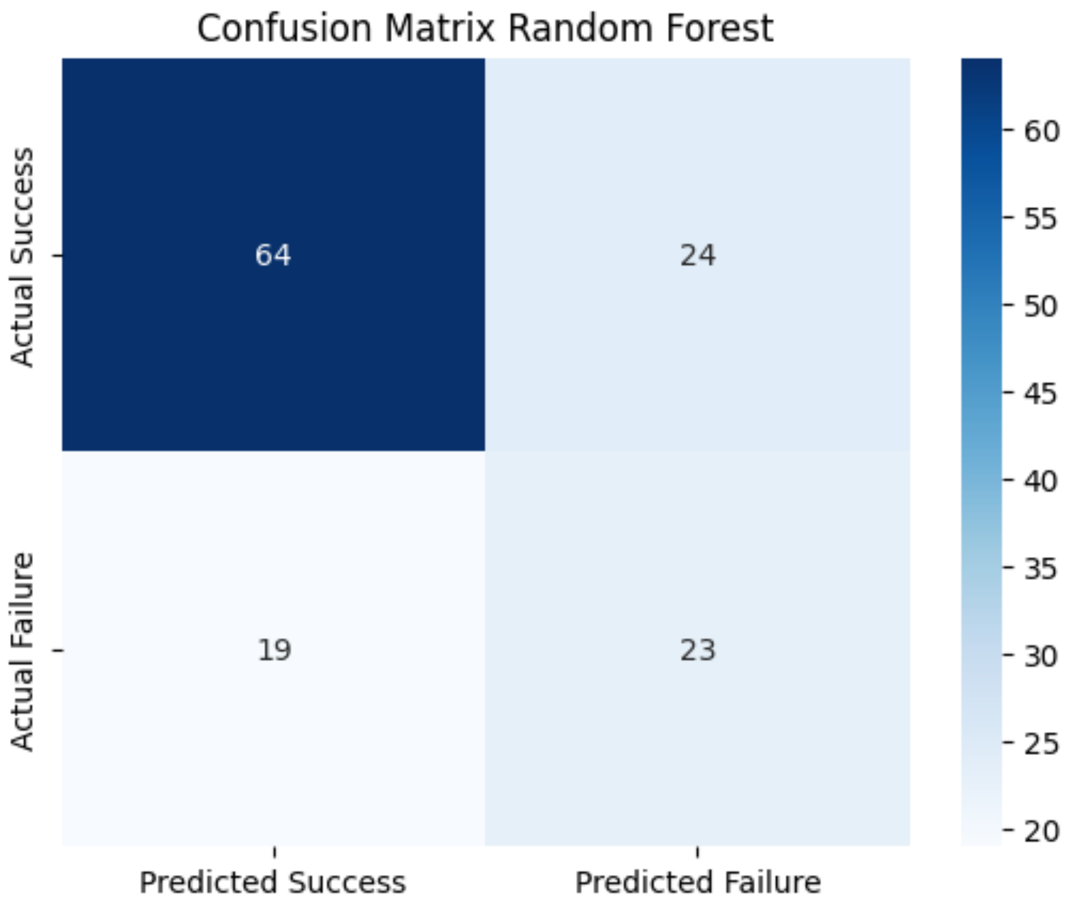


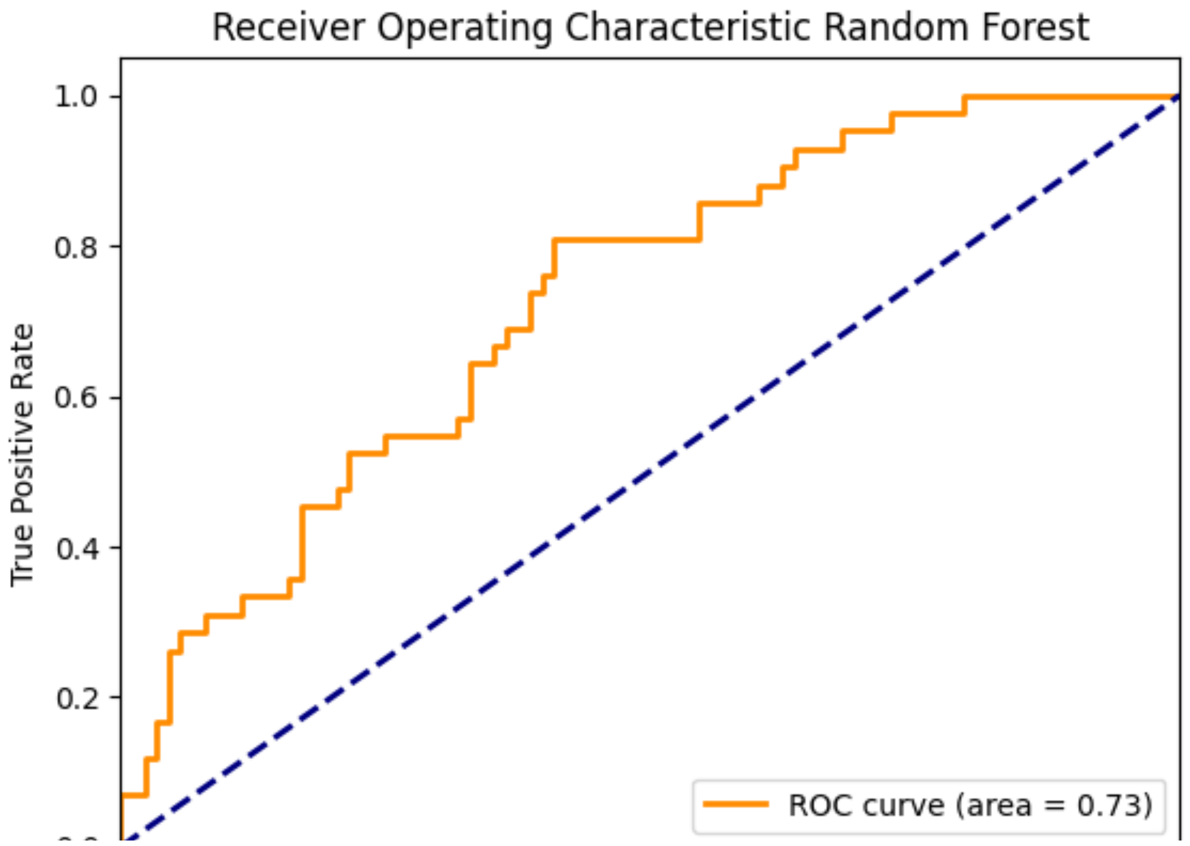
```
Running evaluation with seed 45
Evaluating Random Forest with seed 45...

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.0340909
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.72754329004329
--- Fim dos Dados ROC ---

Training - Accuracy: 0.727810650887574, Sensitivity: 0.09271523178807947, S
Metrics for manual threshold 0.35:
Accuracy: 0.6692307692307692, Sensitivity: 0.5476190476190477, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.34615384615384615, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5461538461538461, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6461538461538462, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6692307692307692, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.45, Metrics: {'Accuracy': 0.7153846153846154, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.6923076923076923, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
```

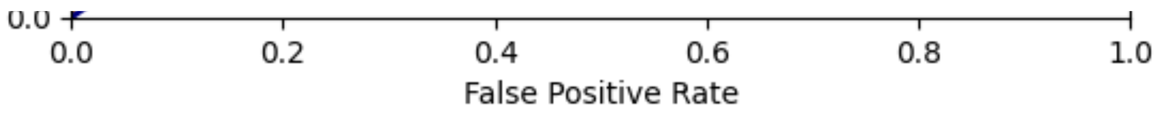
```
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Random Forest
```



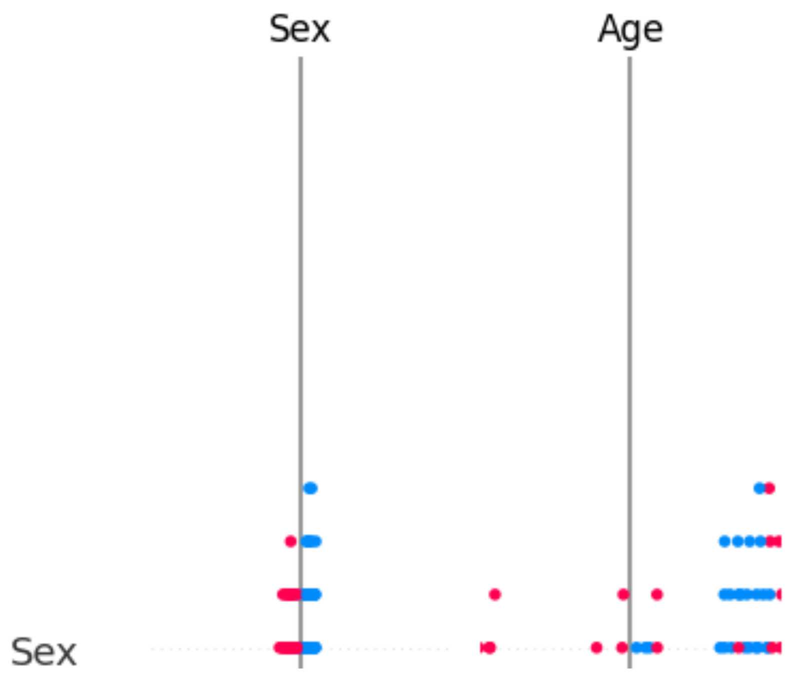

−0.1
0.0
0.1
−0.1
0.0
0.1
SHAP interaction val

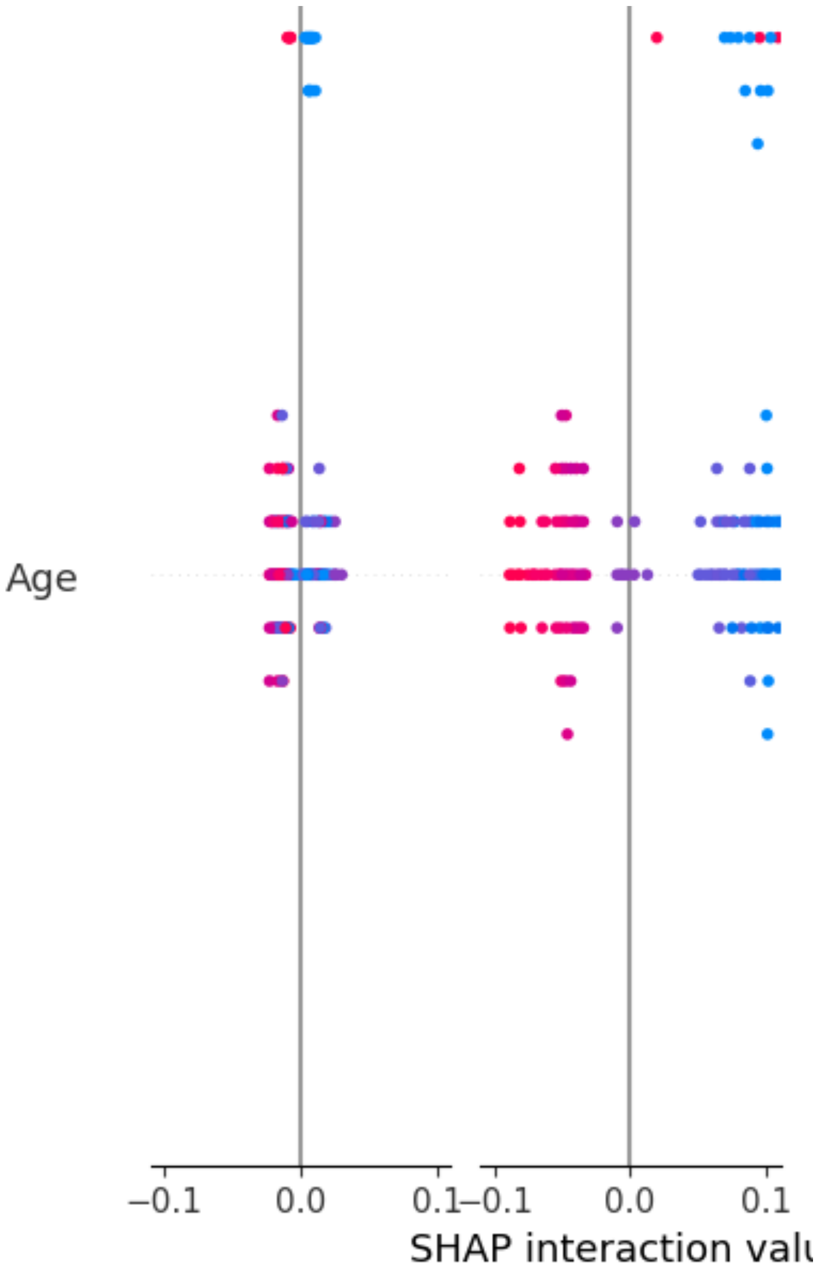

Confusion Matrix Random Forest
Actual Success
Actual Failure
Predicted Success
Predicted Failure
64
24
19
23
60
55
50
45
40
35
30
25
20

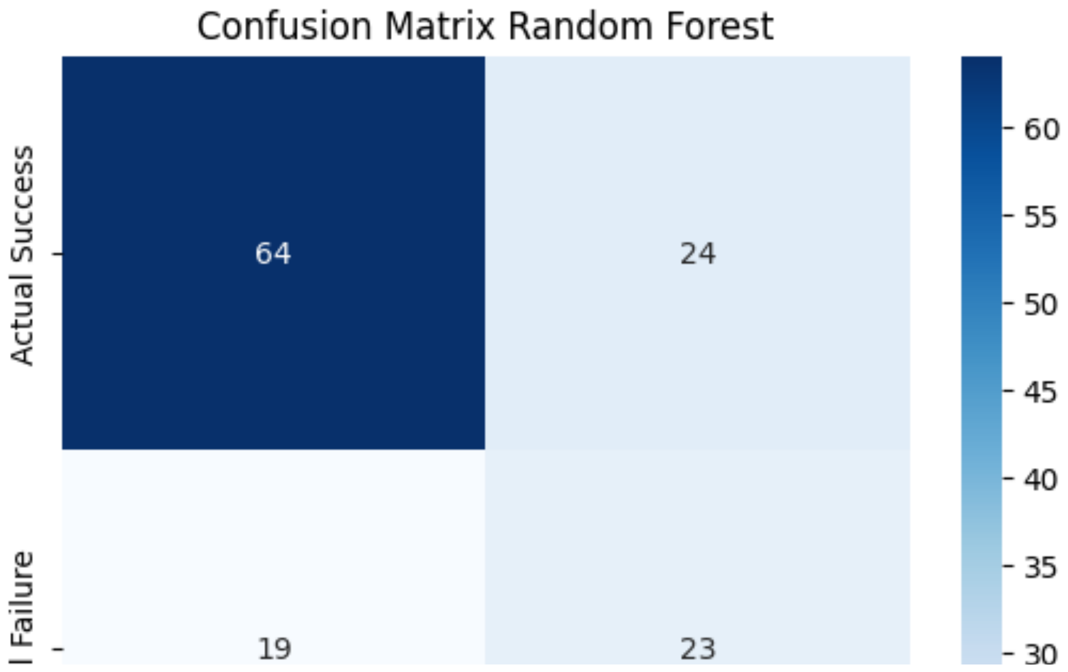

Receiver Operating Characteristic Random Forest
True Positive Rate
1.0
0.8
0.6
0.4
0.2
ROC curve (area = 0.73)

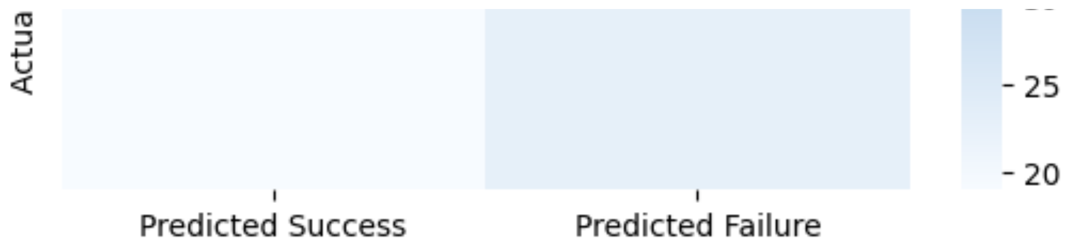


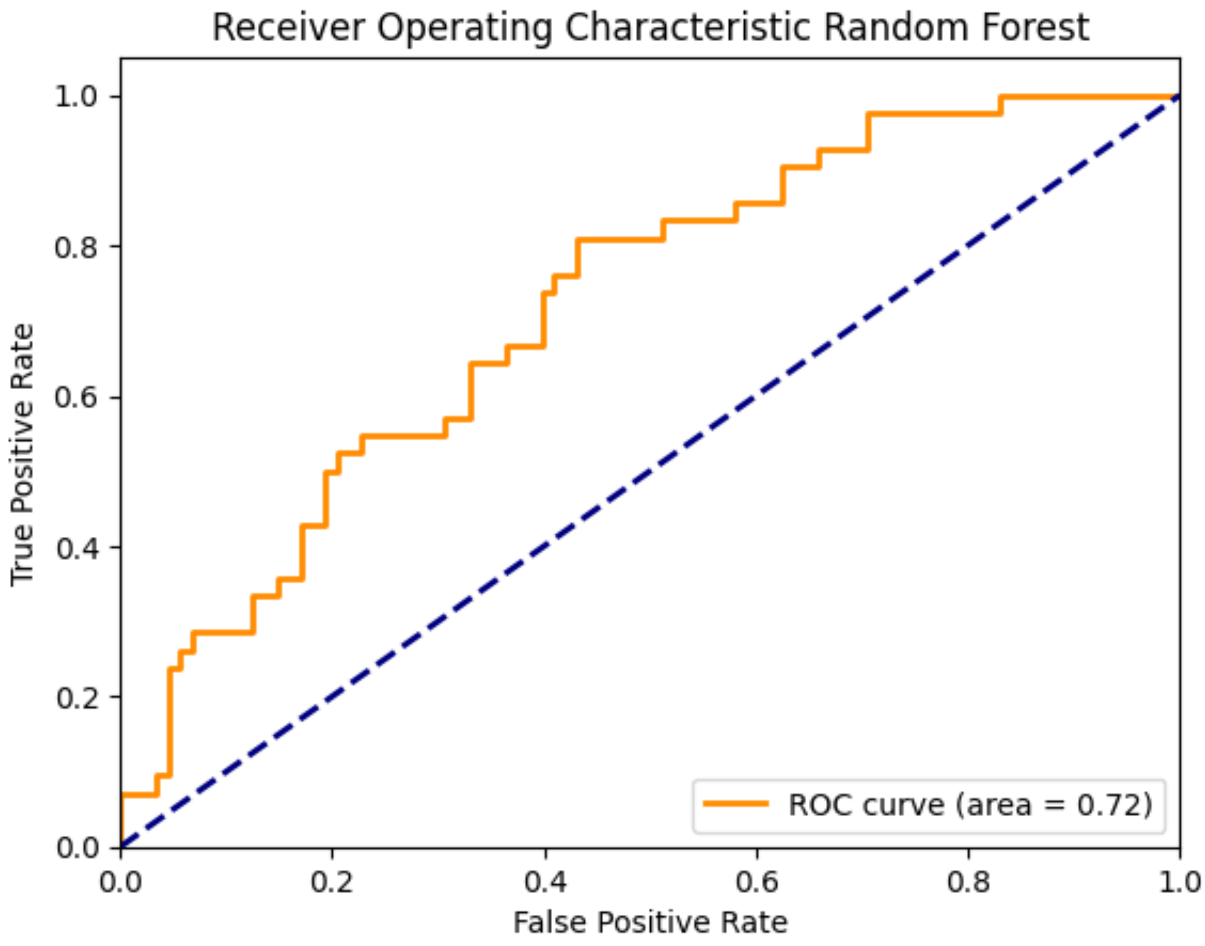
```
Running evaluation with seed 46
Evaluating Random Forest with seed 46...

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.03409090909090909, 0.03409090909090909, 0.045454545
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7210497835497836
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7238658777120316, Sensitivity: 0.0728476821192053, S
Metrics for manual threshold 0.35:
Accuracy: 0.6692307692307692, Sensitivity: 0.5476190476190477, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.33076923076923076, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5461538461538461, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6307692307692307, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6692307692307692, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6923076923076923, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7153846153846154, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7, 'Sensitivity': 0.07142857142857
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Random Forest
```




Age
–0.1
0.0
0.1–0.1
0.0
0.1
SHAP interaction valu

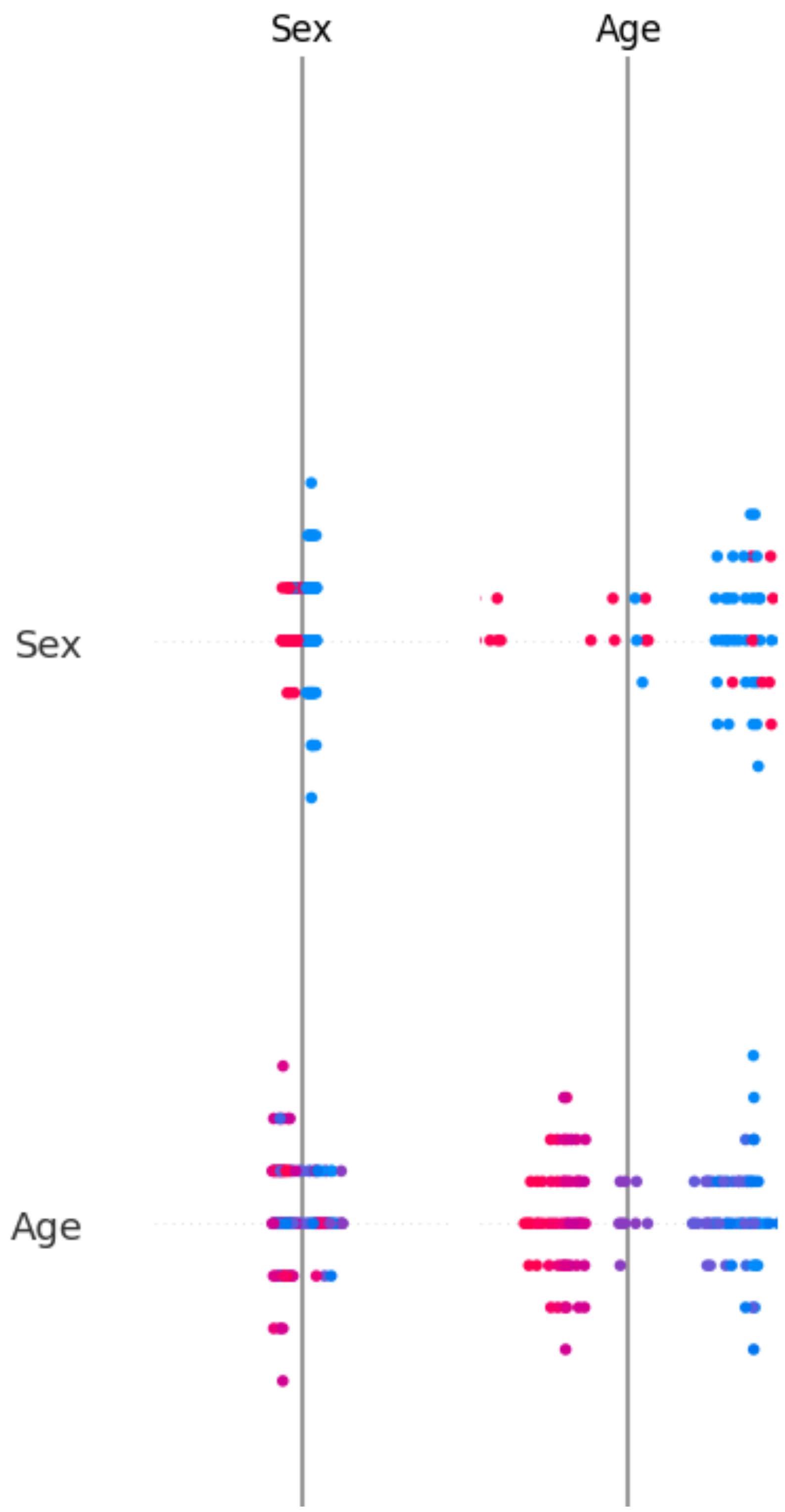

Confusion Matrix Random Forest
Actual Success
l Failure
64
24
19
23
60
55
50
45
40
35
30

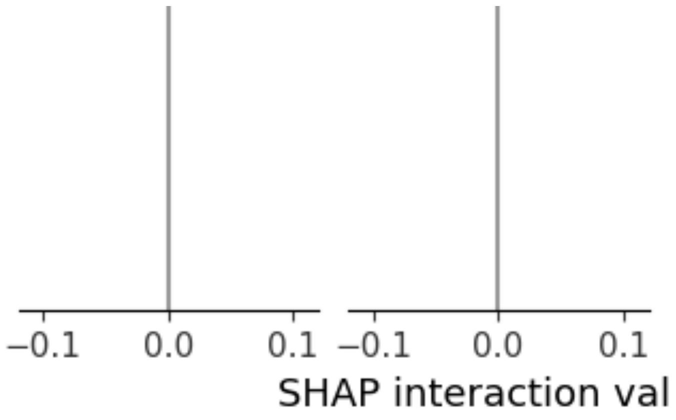


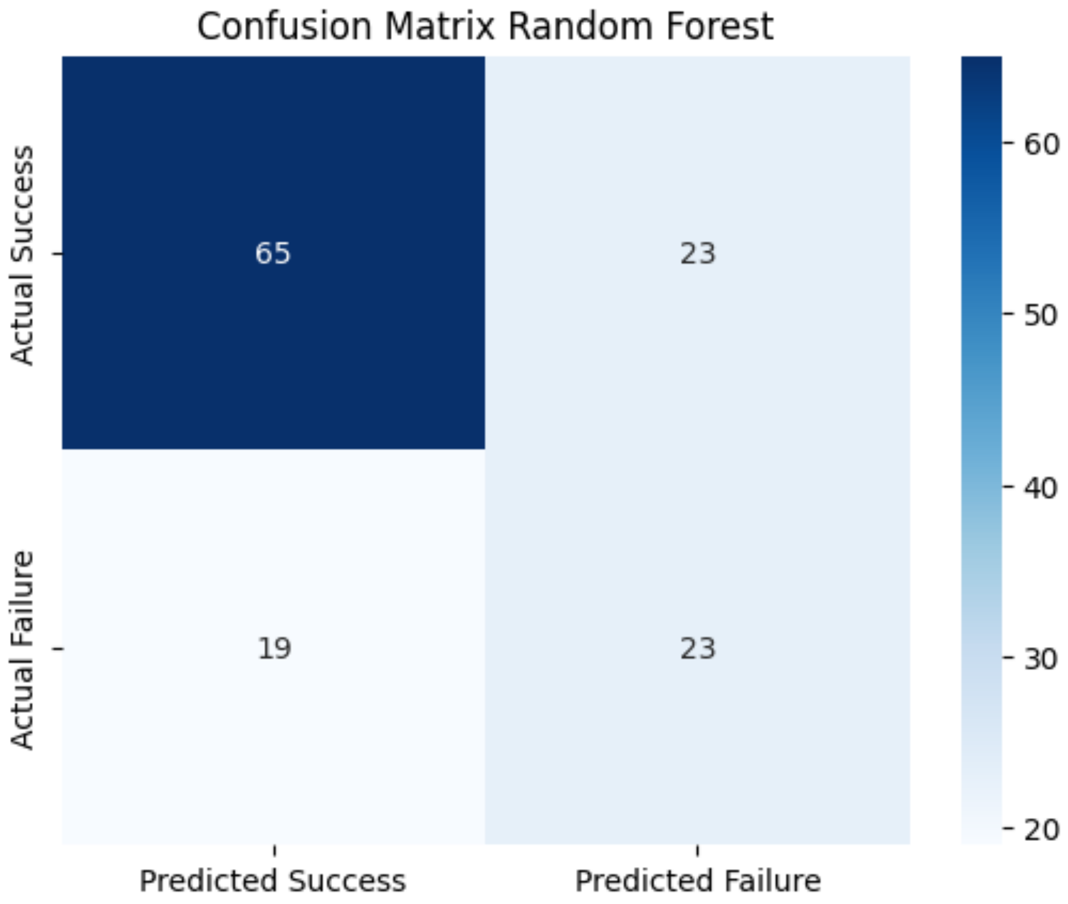


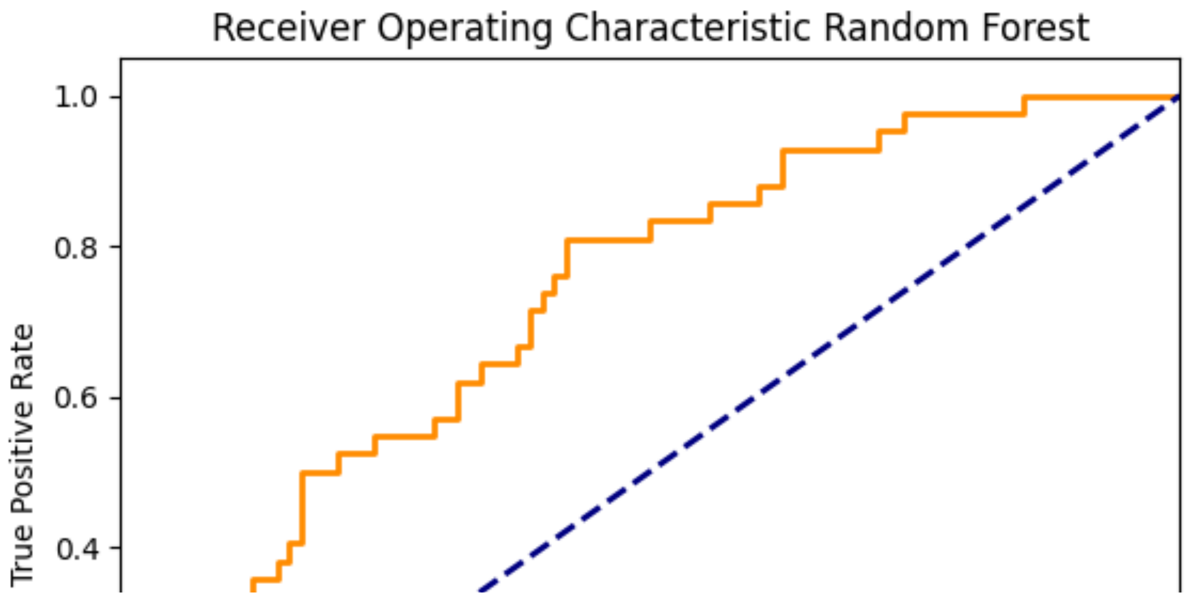
```
Running evaluation with seed 47
Evaluating Random Forest with seed 47...

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.03409090909090909, 0.03409090909090909, 0.034090909
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7270021645021646
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7238658777120316, Sensitivity: 0.07947019867549669,
Metrics for manual threshold 0.35:
Accuracy: 0.676923076923077, Sensitivity: 0.5476190476190477, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.33076923076923076, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5461538461538461, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6384615384615384, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.40, Metrics: {'Accuracy': 0.7, 'Sensitivity': 0.38095238095238
Threshold: 0.45, Metrics: {'Accuracy': 0.7153846153846154, 'Sensitivity': 0
```

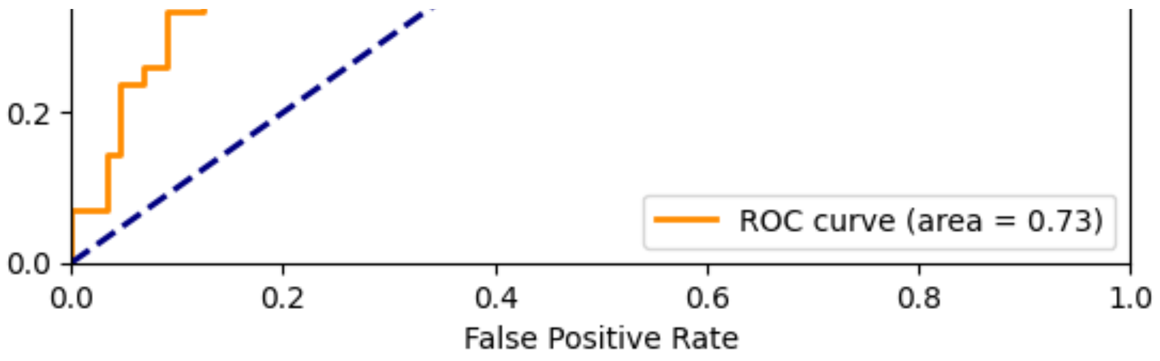
```
Threshold: 0.50, Metrics: {'Accuracy': 0.7, 'Sensitivity': 0.07142857142857
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Random Forest
```



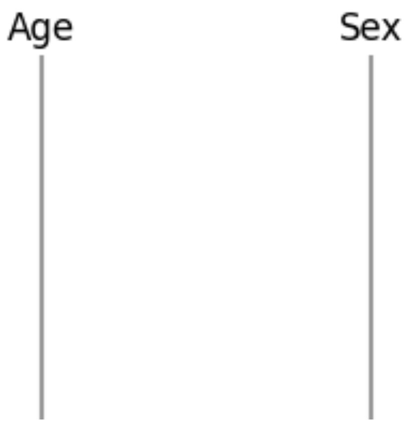

−0.1
0.0
0.1
−0.1
0.0
0.1
SHAP interaction val

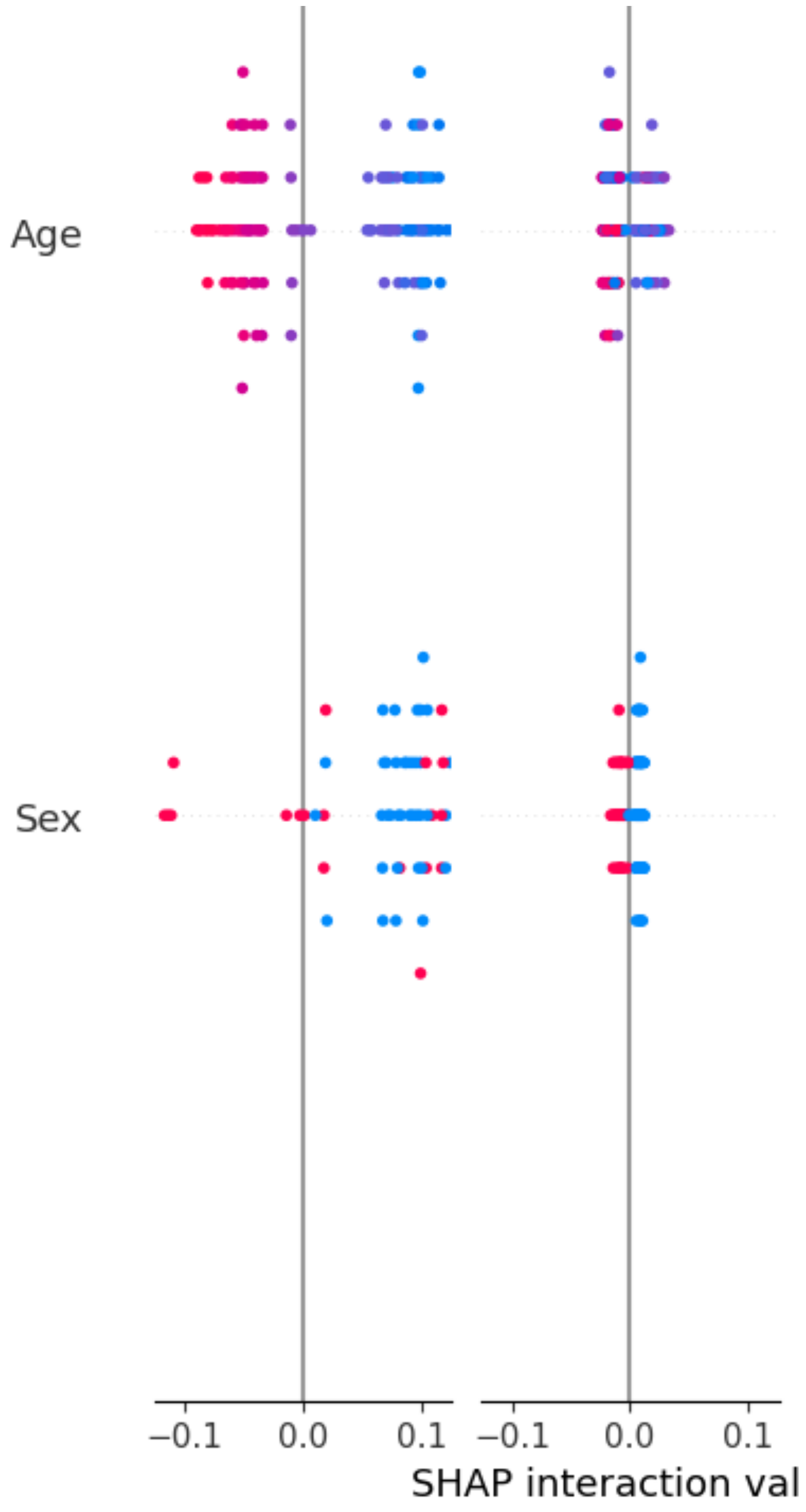

Confusion Matrix Random Forest
Actual Success
Actual Failure
Predicted Success
Predicted Failure
65
23
19
23
60
50
40
30
20

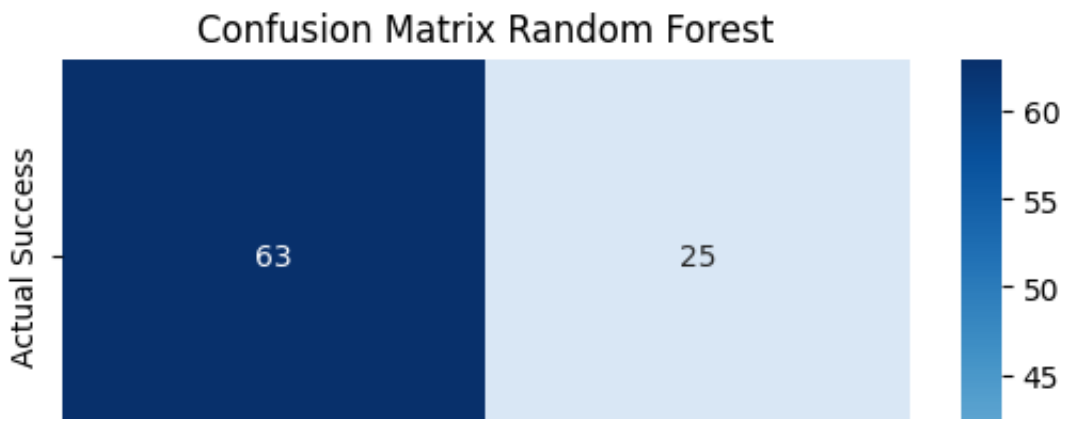

Receiver Operating Characteristic Random Forest
True Positive Rate
1.0
0.8
0.6
0.4

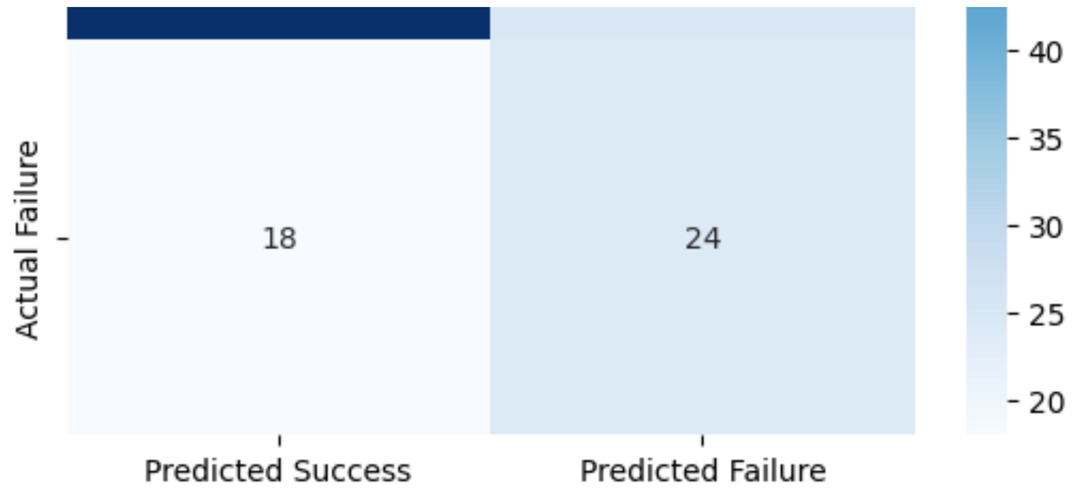


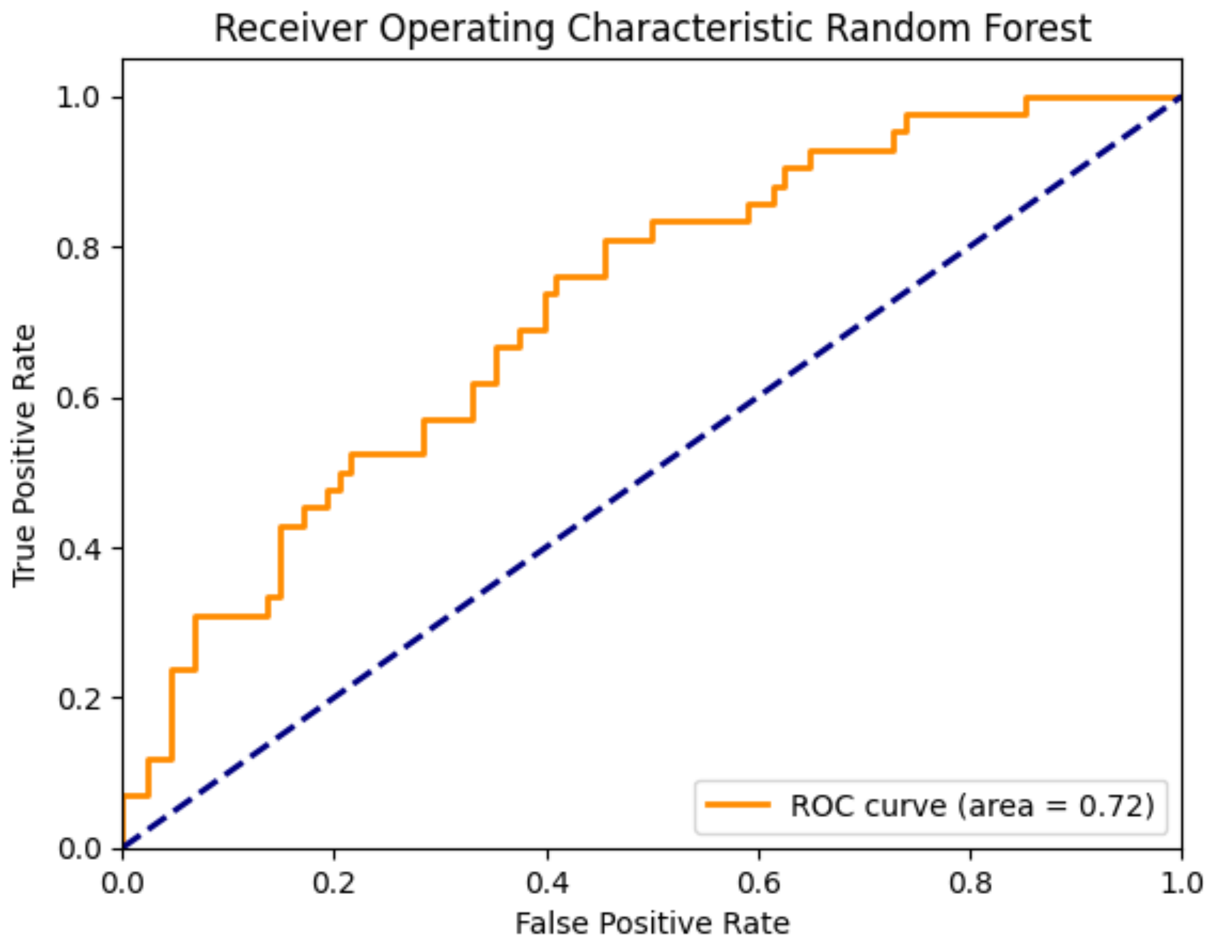
```
Running evaluation with seed 48
Evaluating Random Forest with seed 48...

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.0454545
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7213203463203464
--- Fim dos Dados ROC ---

Training - Accuracy: 0.727810650887574, Sensitivity: 0.09271523178807947, S
Metrics for manual threshold 0.35:
Accuracy: 0.6692307692307692, Sensitivity: 0.5714285714285714, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.36923076923076925, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5461538461538461, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6461538461538462, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6692307692307692, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6923076923076923, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7230769230769231, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7, 'Sensitivity': 0.07142857142857
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for Random Forest
```




Age
Sex
−0.1
0.0
0.1
−0.1
0.0
0.1
SHAP interaction val

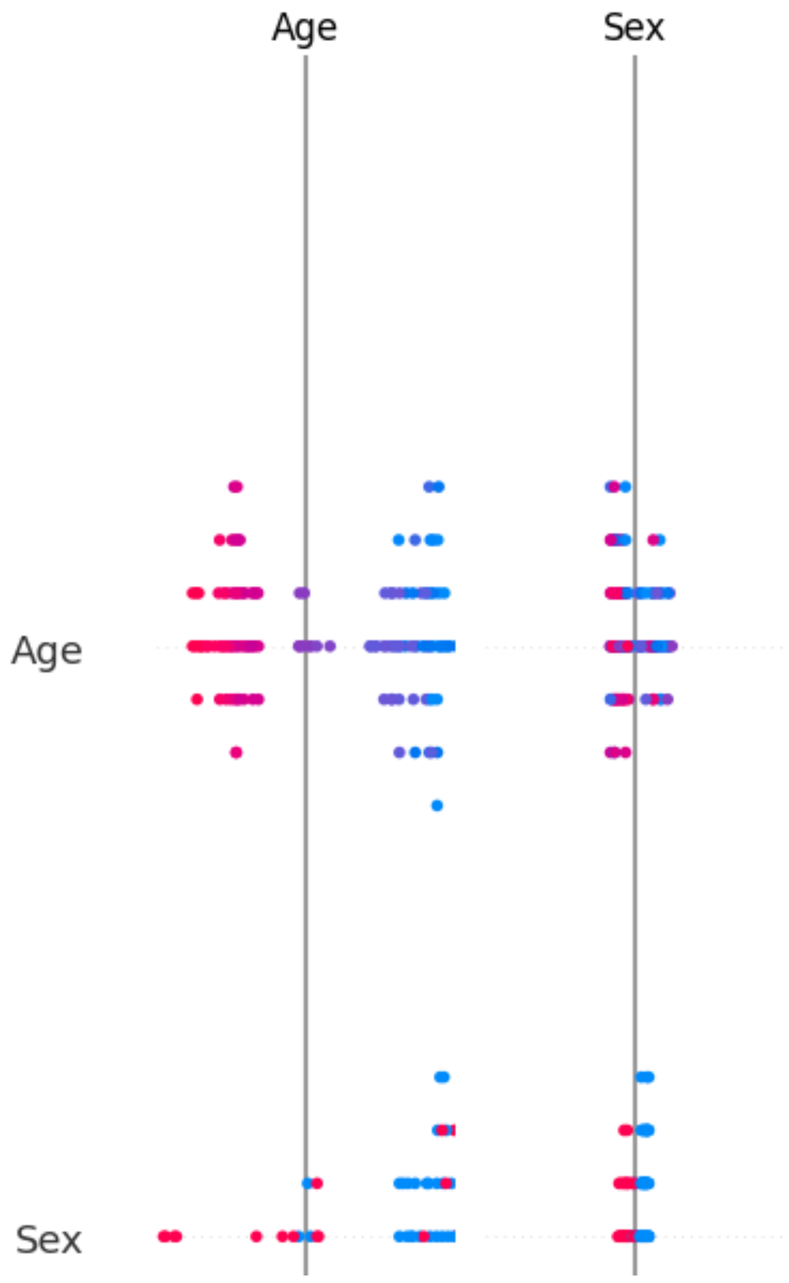

Confusion Matrix Random Forest
Actual Success
63
25
60
55
50
45

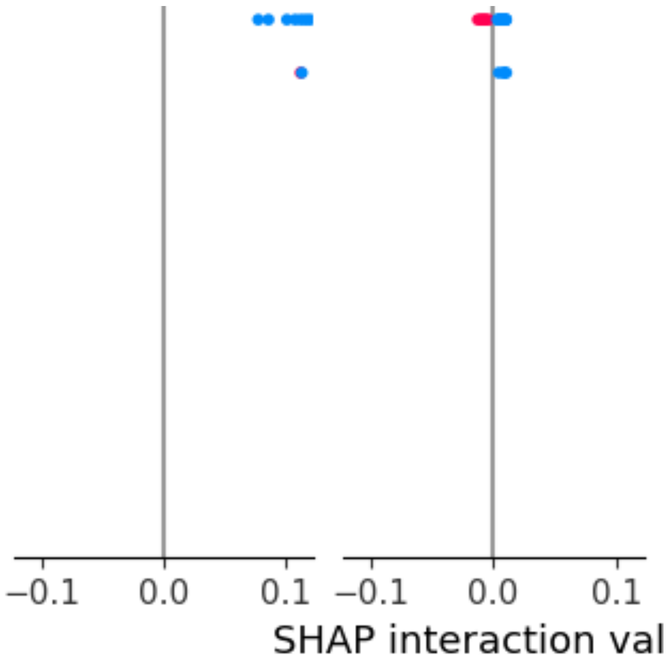


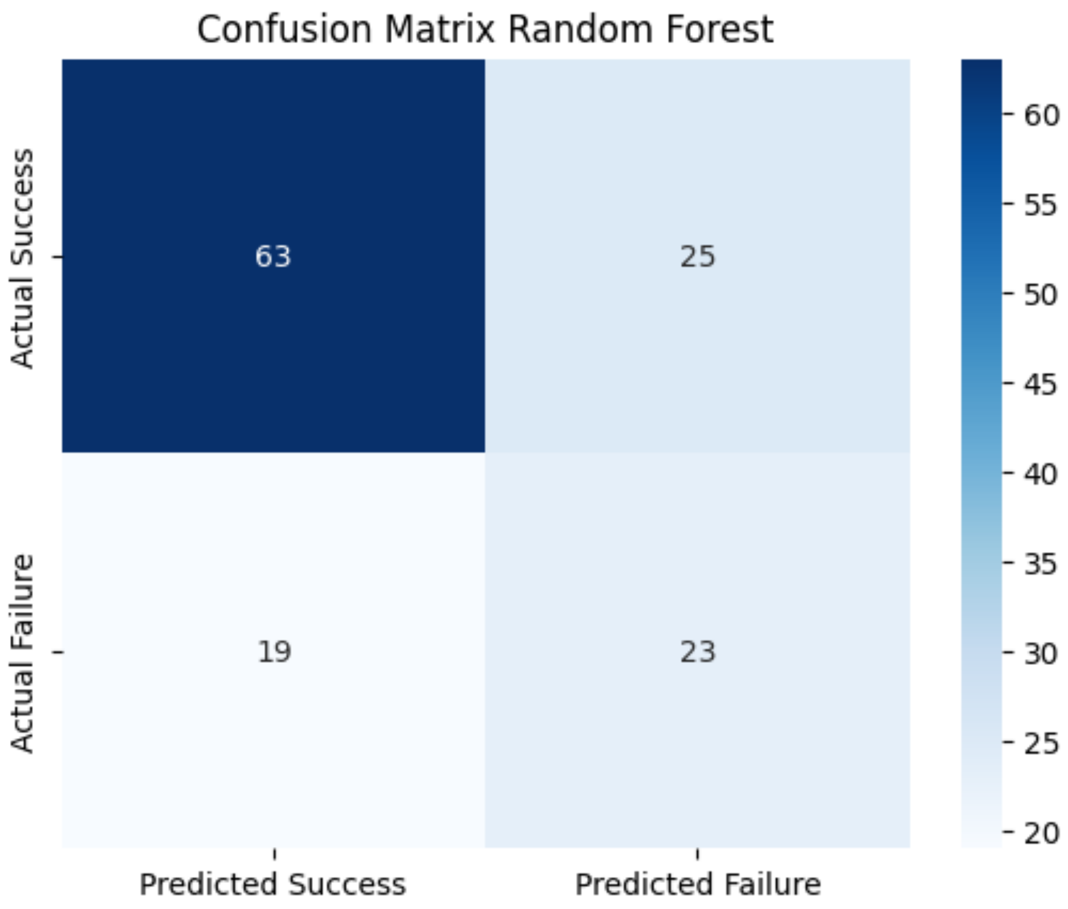


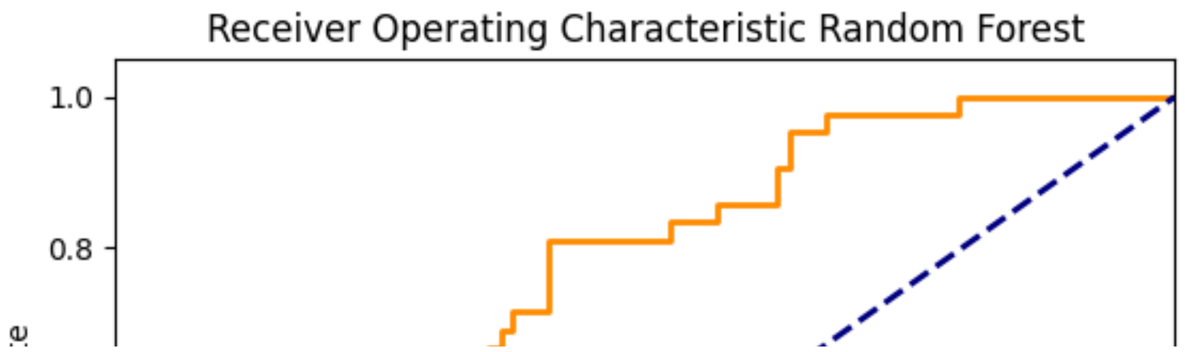
```
Running evaluation with seed 49
Evaluating Random Forest with seed 49...

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.0454545
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7297077922077922
--- Fim dos Dados ROC ---

Training - Accuracy: 0.717948717948718, Sensitivity: 0.052980132450331126,
Metrics for manual threshold 0.35:
Accuracy: 0.6615384615384615, Sensitivity: 0.5476190476190477, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
```

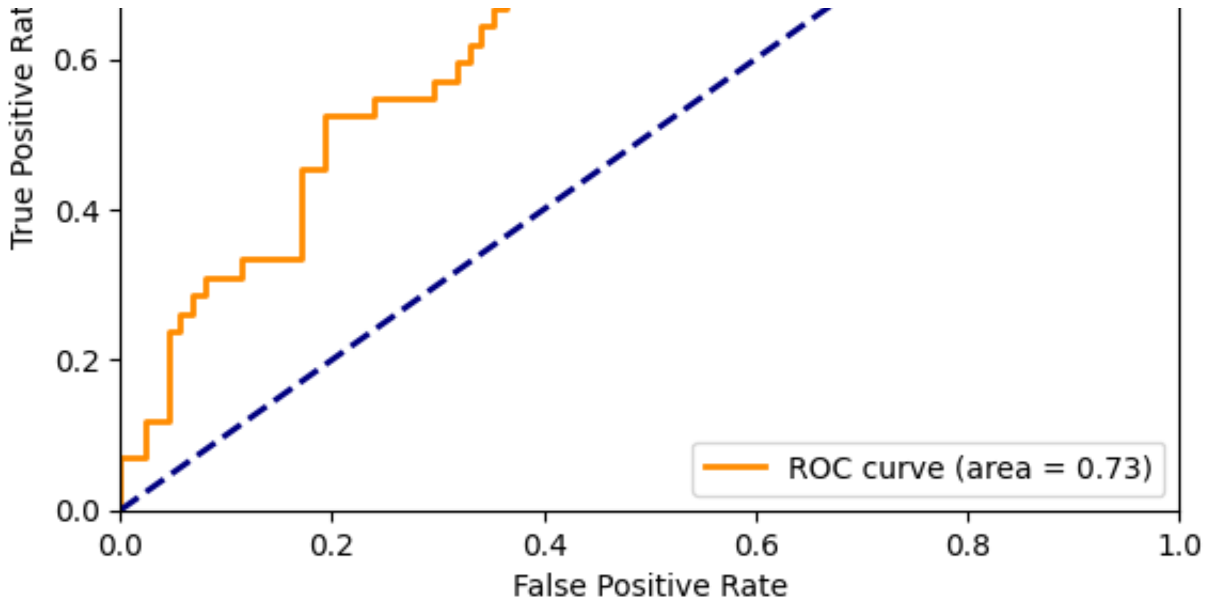
```
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.34615384615384615, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5461538461538461, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6461538461538462, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6615384615384615, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.45, Metrics: {'Accuracy': 0.7230769230769231, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.6923076923076923, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, '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
Predicted Success
Predicted Failure
63
25
19
23
60
55
50
45
40
35
30
25
20


Receiver Operating Characteristic Random Forest
1.0
0.8



```
Aggregated Test Set Metrics Across Seeds:
   accuracy  sensitivity  specificity        f1   roc_auc
0  0.646154     0.547619     0.693182  0.500000  0.726461
1  0.669231     0.547619     0.727273  0.516854  0.729437
2  0.669231     0.547619     0.727273  0.516854  0.722944
3  0.669231     0.547619     0.727273  0.516854  0.728896
4  0.661538     0.547619     0.715909  0.511111  0.728355
5  0.669231     0.547619     0.727273  0.516854  0.727543
6  0.669231     0.547619     0.727273  0.516854  0.721050
7  0.676923     0.547619     0.738636  0.522727  0.727002
8  0.669231     0.571429     0.715909  0.527473  0.721320
9  0.661538     0.547619     0.715909  0.511111  0.729708

Summary of Test Set Metrics (Mean, Standard Error, 95% Confidence Interval)
Accuracy: Mean = 0.666, SE = 0.003, 95% CI = [0.660, 0.672]
Sensitivity: Mean = 0.550, SE = 0.002, 95% CI = [0.545, 0.555]
Specificity: Mean = 0.722, SE = 0.004, 95% CI = [0.713, 0.730]
F1: Mean = 0.516, SE = 0.002, 95% CI = [0.510, 0.521]
Roc_auc: Mean = 0.726, SE = 0.001, 95% CI = [0.724, 0.729]
```

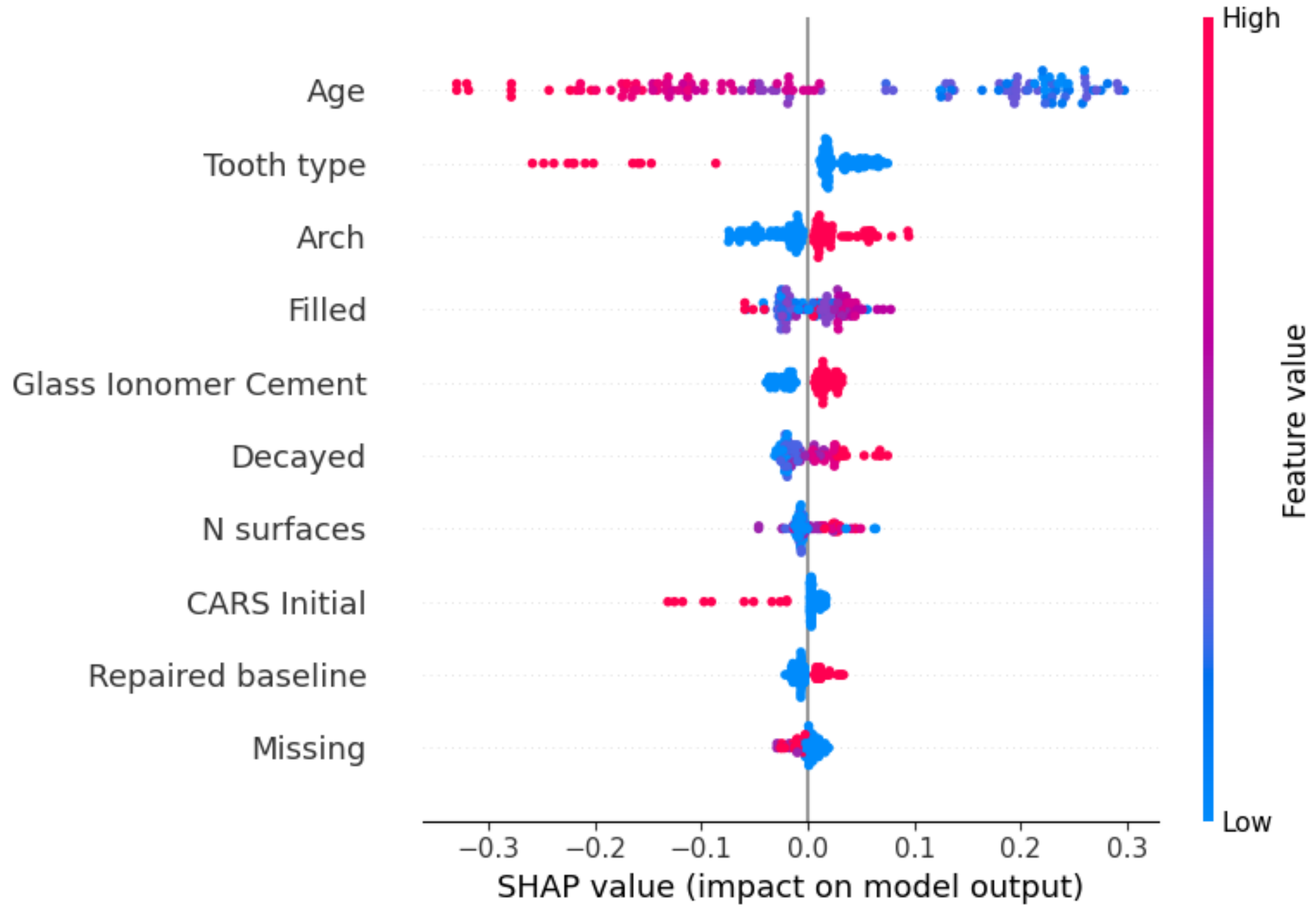
```
def evaluate_model(model, name, grid, X_train, y_train, X_test, y_test, cv, scor
    print(f"Evaluating {name}...")

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

    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

    clf.fit(X_train, y_train)
    best_model = clf.best_estimator_
    best_params = clf.best_params_
    print(f"Best parameters for {name}: {best_params}")

    calibrated_clf = CalibratedClassifierCV(estimator=best_model, method='sigmoi
    calibrated_clf.fit(X_train, y_train)

    y_probs = calibrated_clf.predict_proba(X_test)[:, 1]

    # Calculate FPR, TPR, and AUC
    fpr, tpr, thresholds = roc_curve(y_test, y_probs)
    roc_auc = auc(fpr, tpr)

    print("\n--- Dados ROC para copiar ---")
    print("FPR =", fpr.tolist())
    print("TPR =", tpr.tolist())
    print("AUC =", roc_auc)
    print("--- Fim dos Dados ROC ---\n")

    # Calculate metrics for the training set
    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}, Sensitivity: {train_sens}, Specifi

    # 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"Metrics for manual threshold {manual_threshold}:")
    print(f"Accuracy: {manual_acc}, Sensitivity: {manual_sens}, Specificity: {ma

    # 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}")

    calculate_and_plot_shap(best_model, X_train, X_test, name)

    # Prepare dictionary of test metrics for later 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):
    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)
    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 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()

def evaluate_xgboost(X_train, y_train, X_test, y_test, cv, scoring, manual_thres
    print("Inside evaluate_xgboost function")
    model = XGBClassifier(use_label_encoder=False, eval_metric='logloss', random
    grid = {
        'max_depth': [5],
        'gamma': [0.1],
        'learning_rate': [0.002],
        'subsample': [0.8],
        'colsample_bytree': [1],
        'reg_alpha': [0],
        'reg_lambda': [1],
        'n_estimators': [200]
    }
    return evaluate_model(model, "XGBoost", grid, X_train, y_train, X_test, y_te

def main(X_train, y_train, 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)

    aggregated_metrics = []

    for seed in range(40, 50):
        print(f"Running evaluation with seed {seed}")
        (best_model, manual_threshold, best_params, nested_scores, calibrated_cl
         threshold_metrics, test_metrics) = evaluate_xgboost(X_train, y_train, X
                                                              cv, scoring, manual

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

        plot_confusion_matrix(y_test, y_pred_manual)
        plot_roc_curve(y_test, y_probs)
        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 for
    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)
        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}

if __name__ == '__main__':
    main(X_train, y_train, X_test, y_test)
```

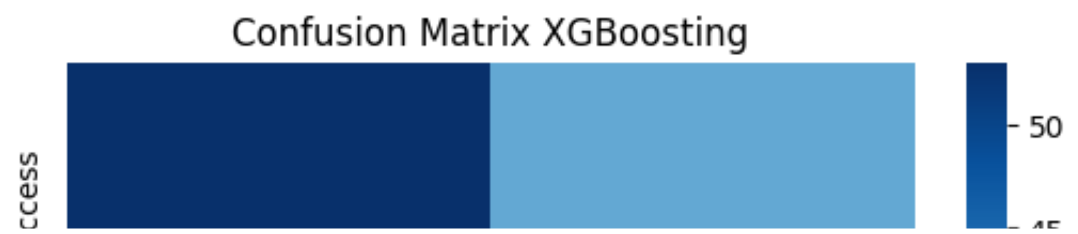
```
Running evaluation with seed 40
Inside evaluate_xgboost function
Evaluating XGBoost...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

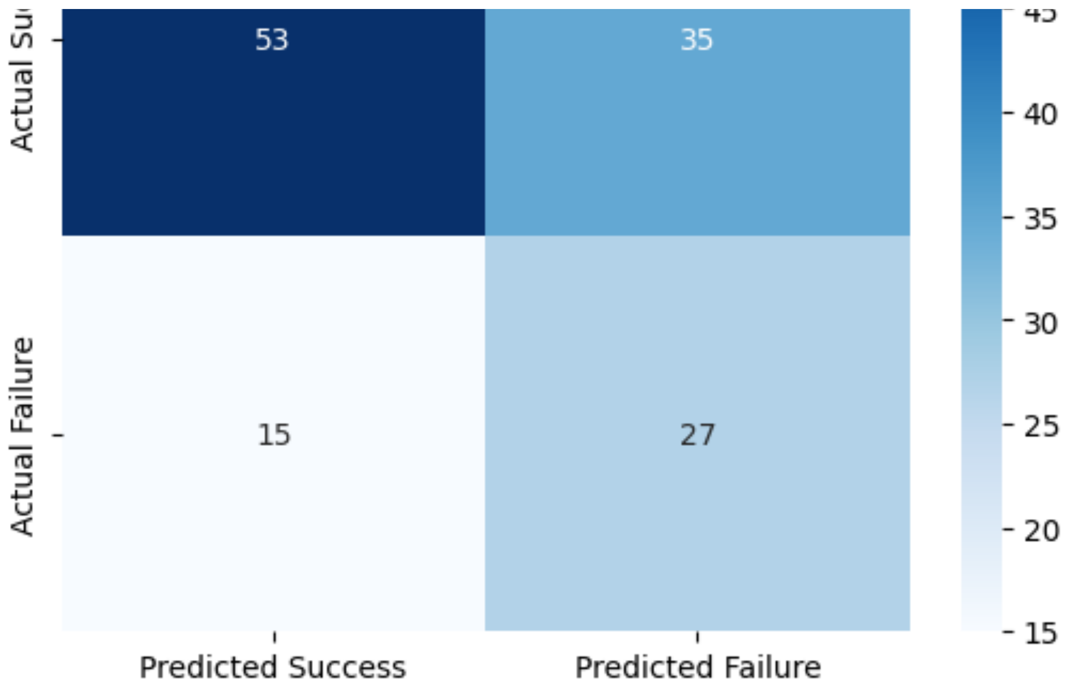
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

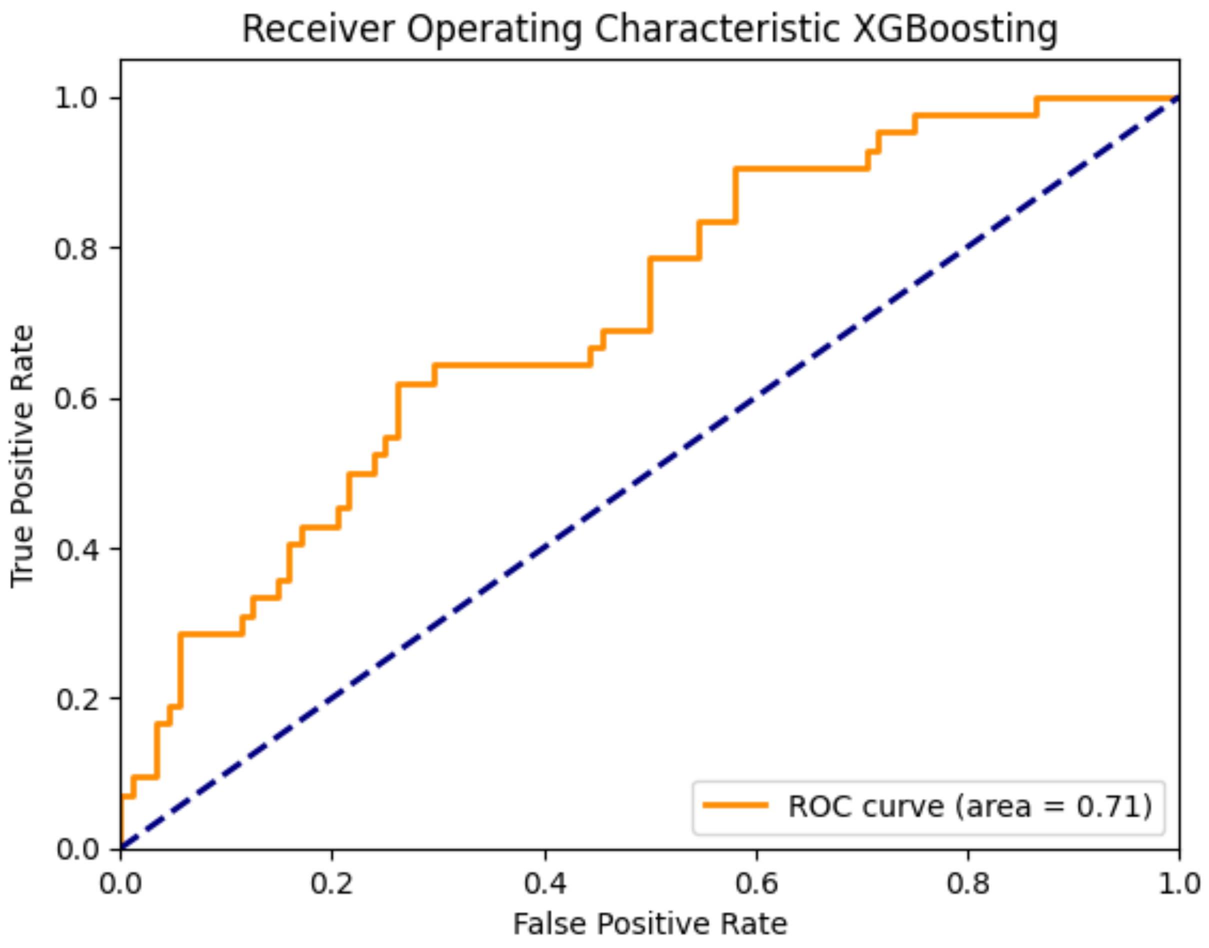
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

  warnings.warn(smsg, 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': 0.1, 'learnin
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

  warnings.warn(smsg, UserWarning)

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.0340909
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7088744588744589
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7021696252465484, Sensitivity: 0.0, Specificity: 1.0
```

```
Metrics for manual threshold 0.3:
Accuracy: 0.6153846153846154, Sensitivity: 0.6428571428571429, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5230769230769231, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6153846153846154, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7230769230769231, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for XGBoost
```





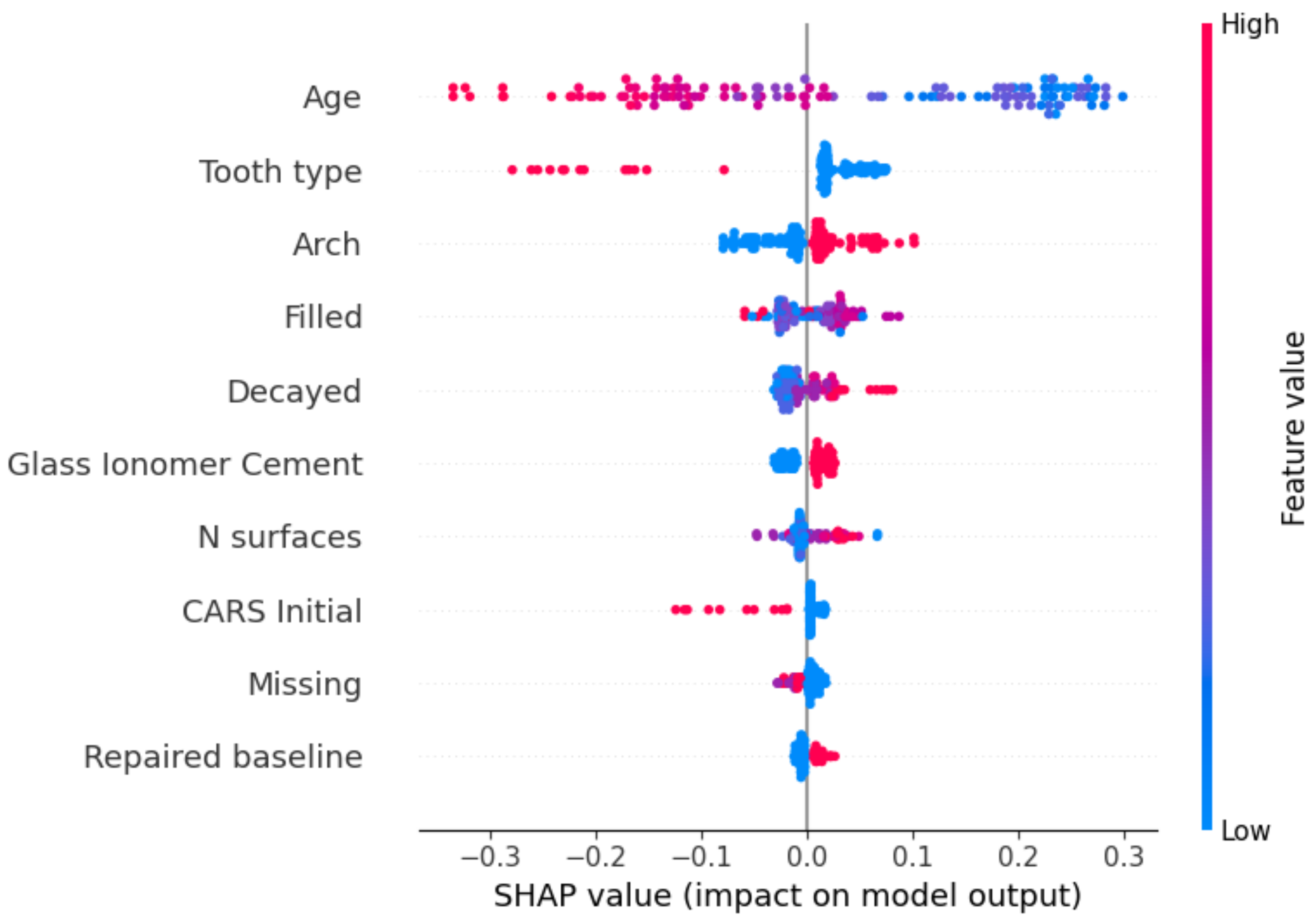


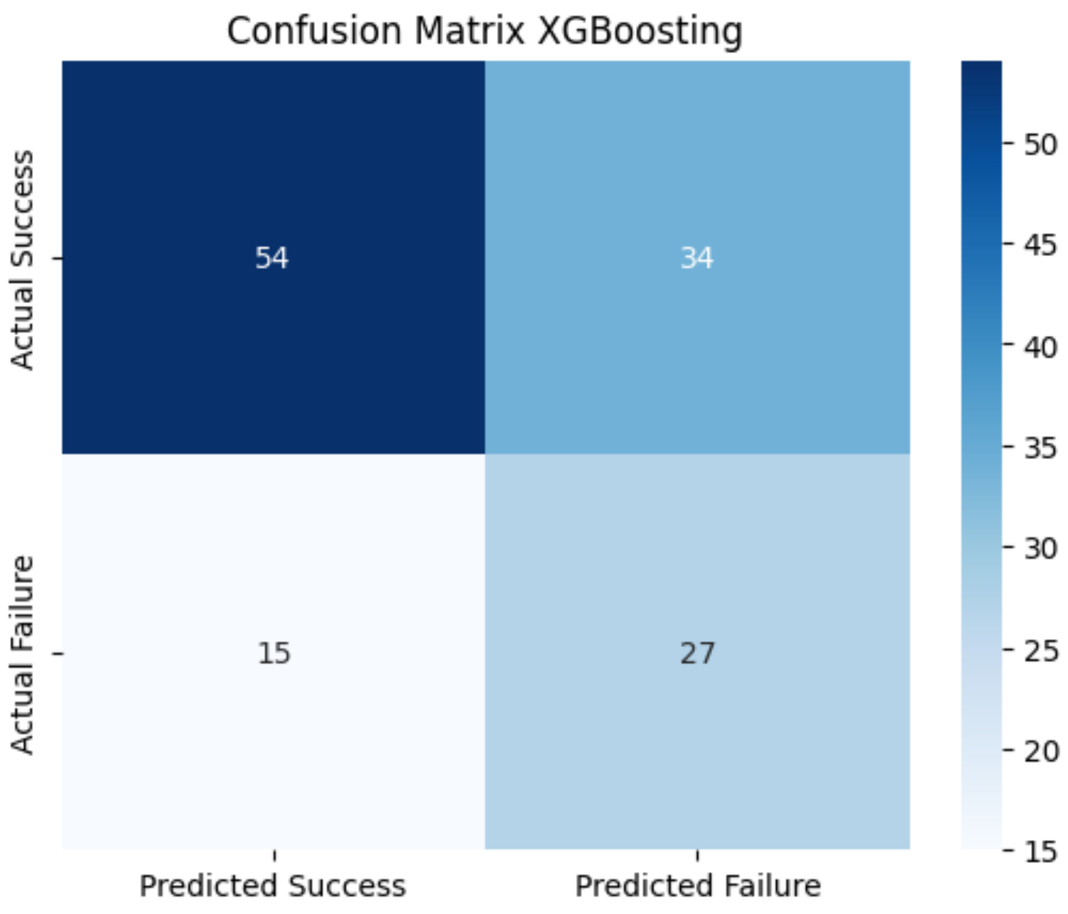


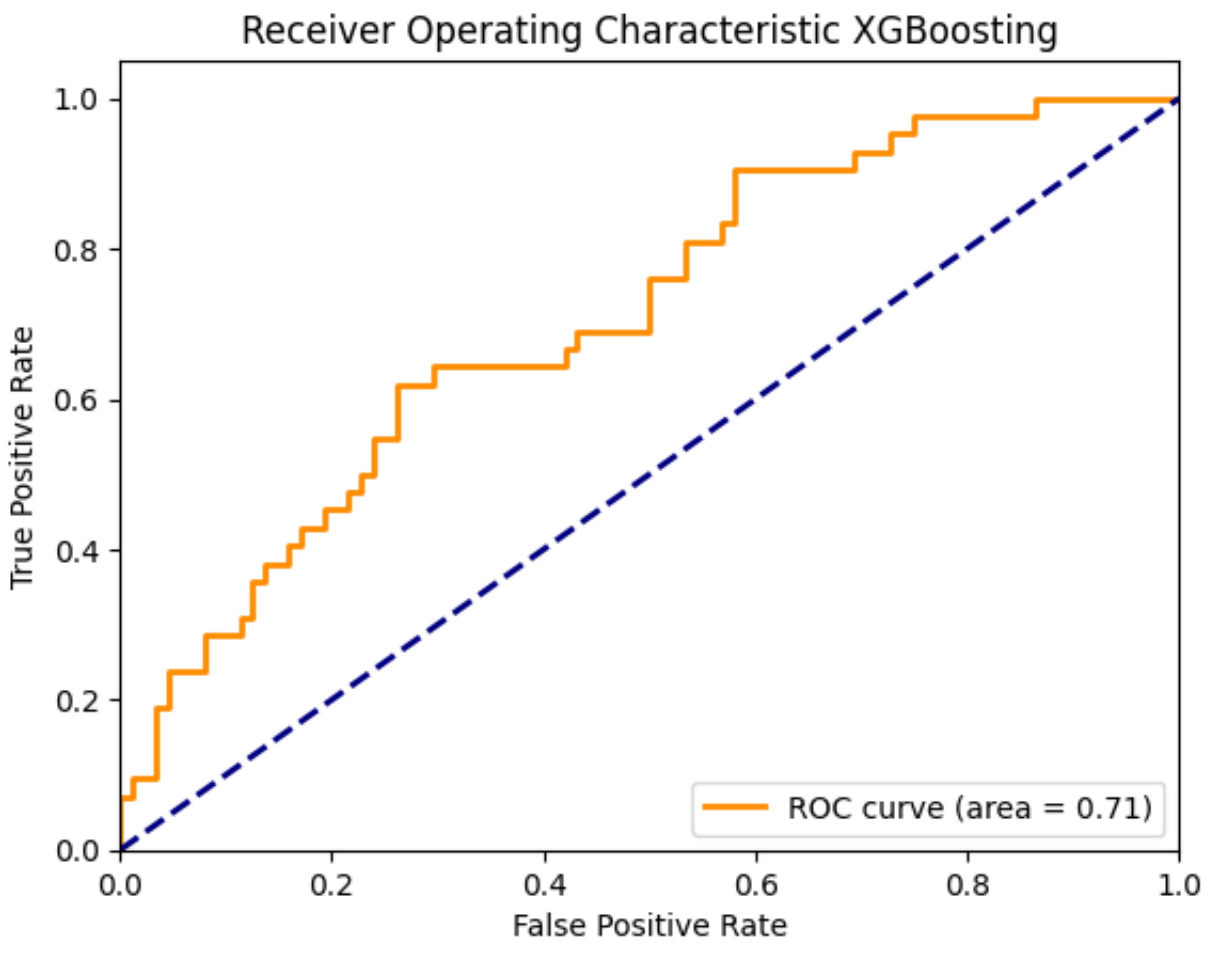
```
Running evaluation with seed 41
Inside evaluate_xgboost function
Evaluating XGBoost...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

  warnings.warn(smsg, 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': 0.1, 'learnin
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

  warnings.warn(smsg, UserWarning)

--- Dados ROC para copiar ---
```

```
FPR = [0.0, 0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.0340909
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.70995670995671
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7021696252465484, Sensitivity: 0.0, Specificity: 1.0
Metrics for manual threshold 0.3:
Accuracy: 0.6230769230769231, Sensitivity: 0.6428571428571429, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5384615384615384, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6230769230769231, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7153846153846154, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for XGBoost
```





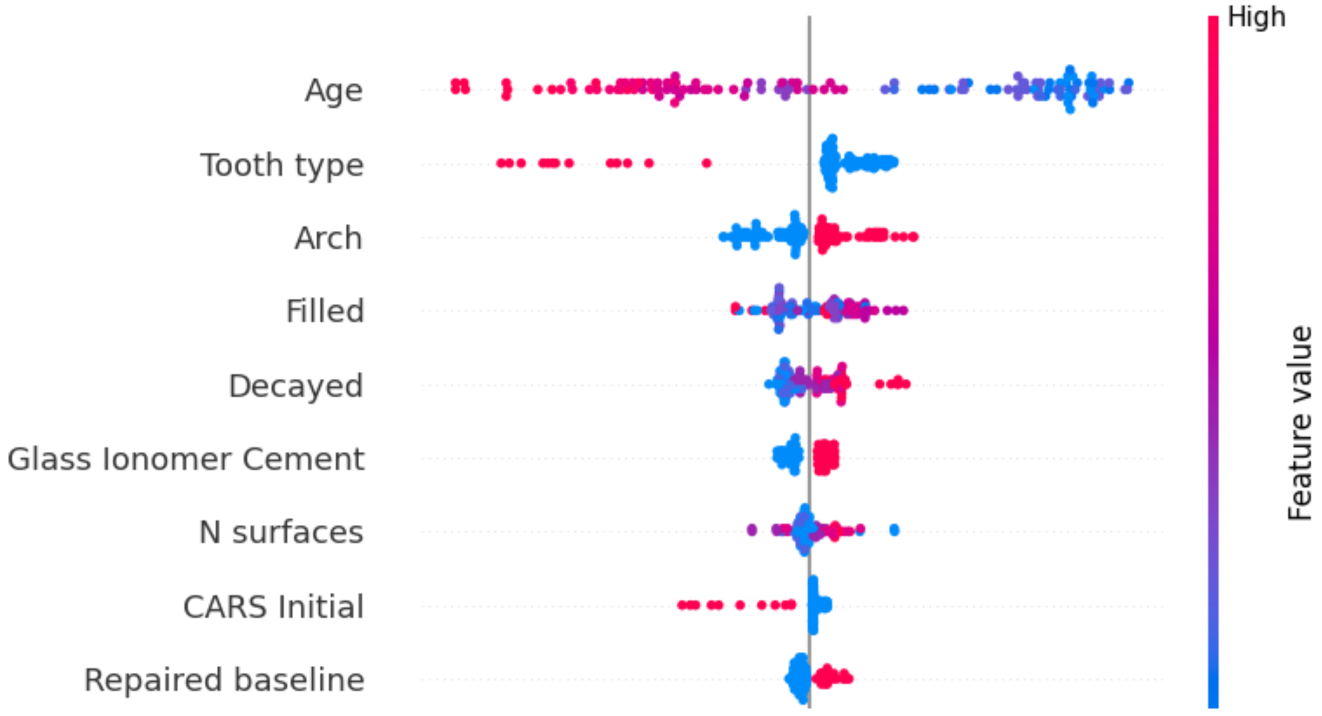


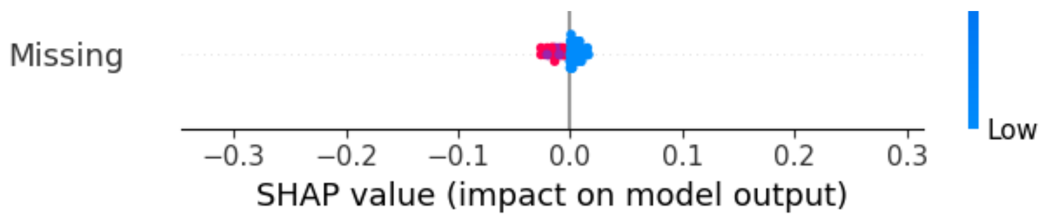
```
Running evaluation with seed 42
Inside evaluate xgboost function
```

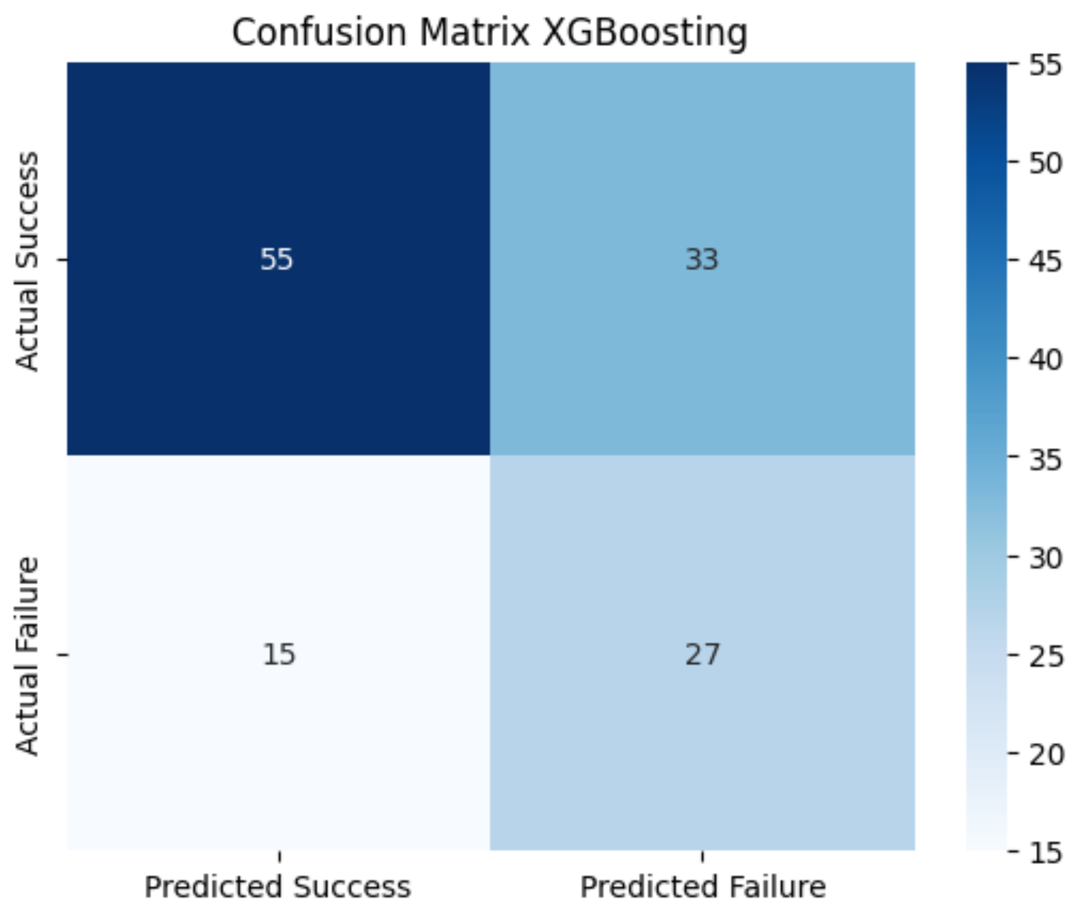
```
Evaluating XGBoost...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

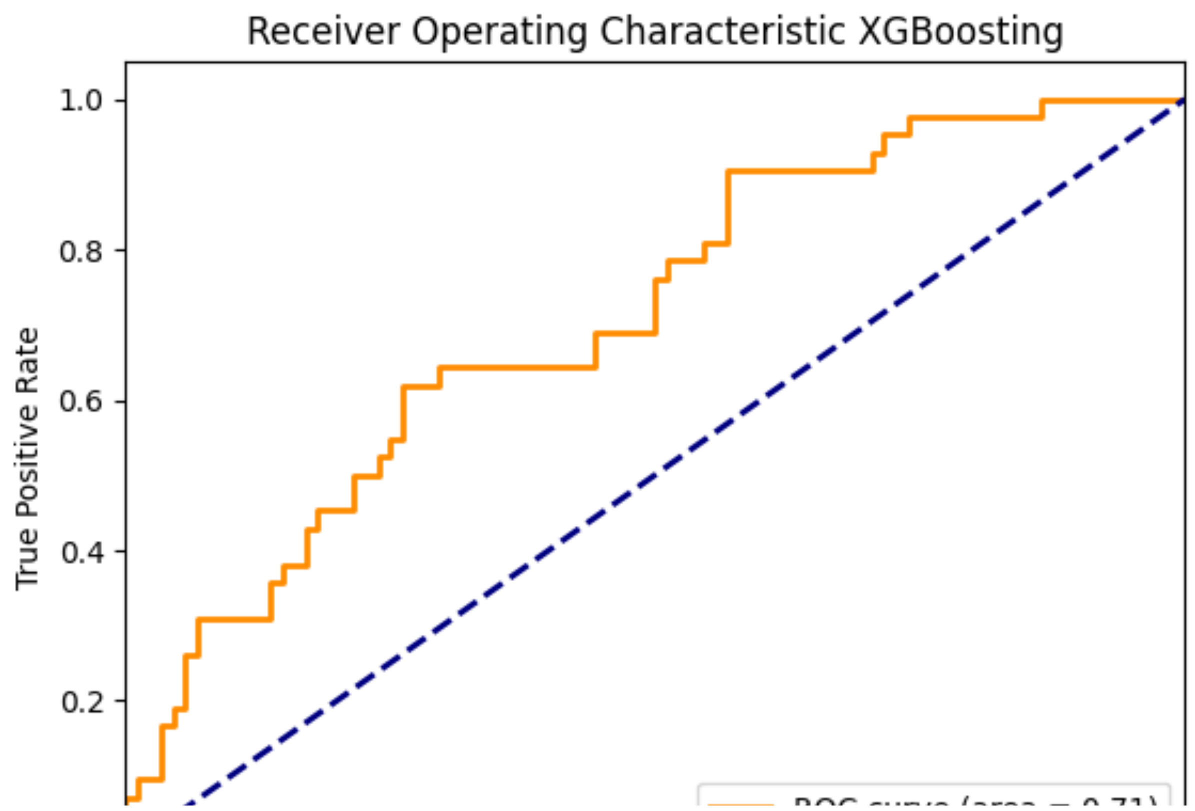
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

  warnings.warn(smsg, 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': 0.1, 'learnin
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

  warnings.warn(smsg, UserWarning)
```

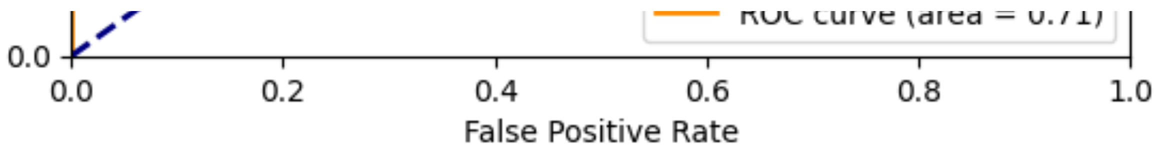
```
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.0340909
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7107683982683983
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7021696252465484, Sensitivity: 0.0, Specificity: 1.0
Metrics for manual threshold 0.3:
Accuracy: 0.6307692307692307, Sensitivity: 0.6428571428571429, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3384615384615385, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5538461538461539, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6307692307692307, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.40, Metrics: {'Accuracy': 0.7307692307692307, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.6923076923076923, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for XGBoost
```




Missing
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
SHAP value (impact on model output)
Low


Confusion Matrix XGBoosting
Actual Success
Actual Failure
Predicted Success
Predicted Failure
55
33
15
27
55
50
45
40
35
30
25
20
15

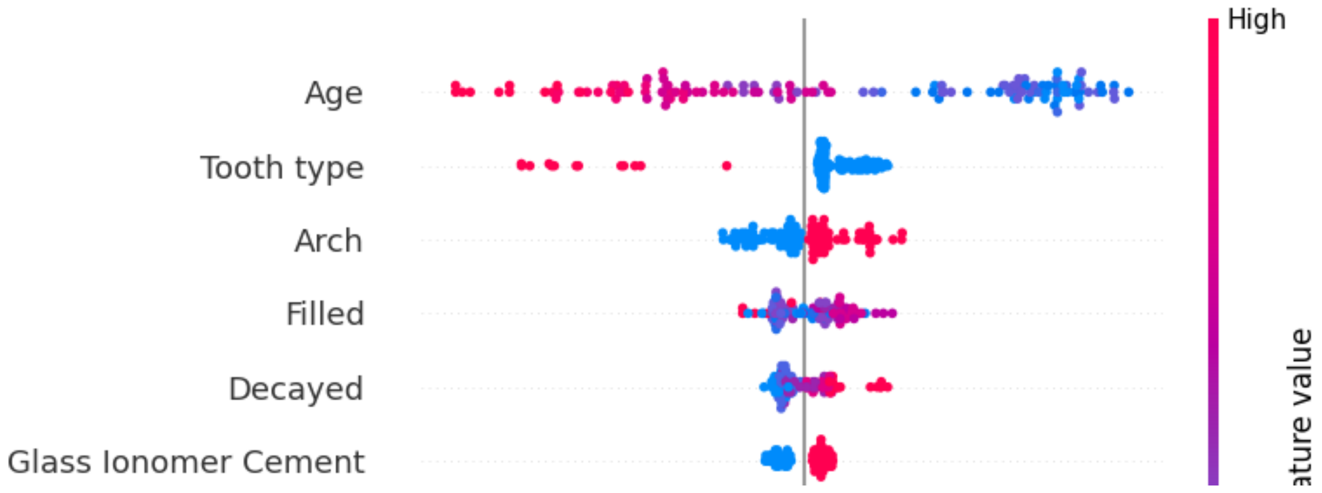

Receiver Operating Characteristic XGBoosting
True Positive Rate
1.0
0.8
0.6
0.4
0.2
ROC curve (area = 0.71)

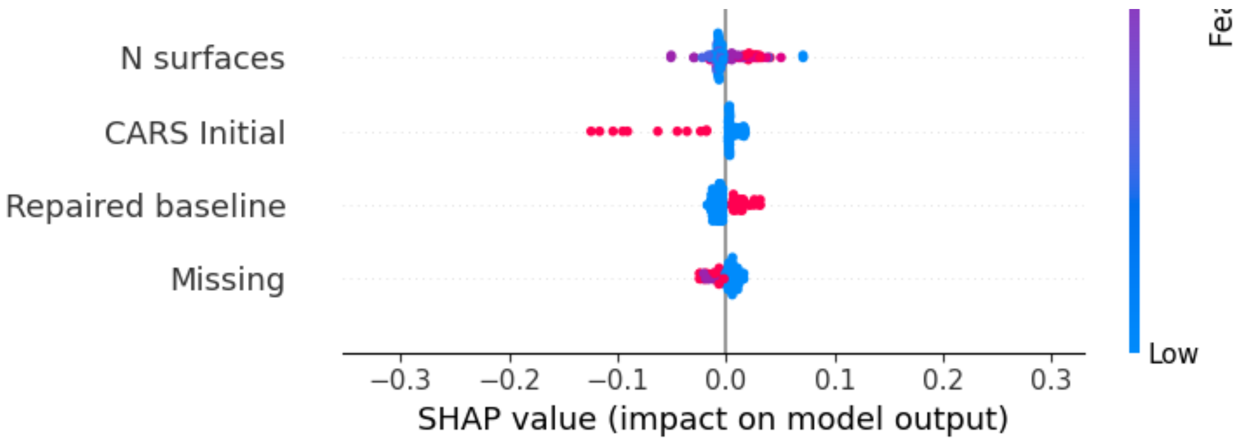


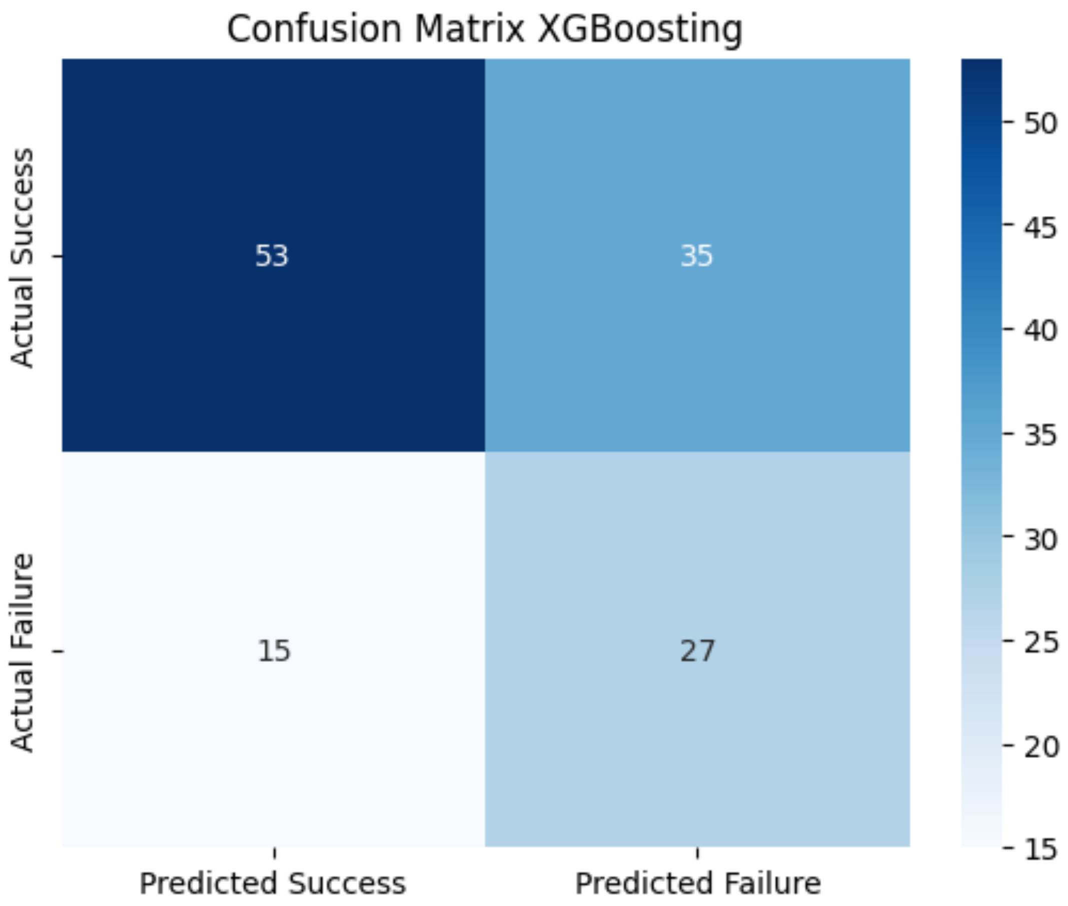
```
Running evaluation with seed 43
Inside evaluate_xgboost function
Evaluating XGBoost...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

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

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

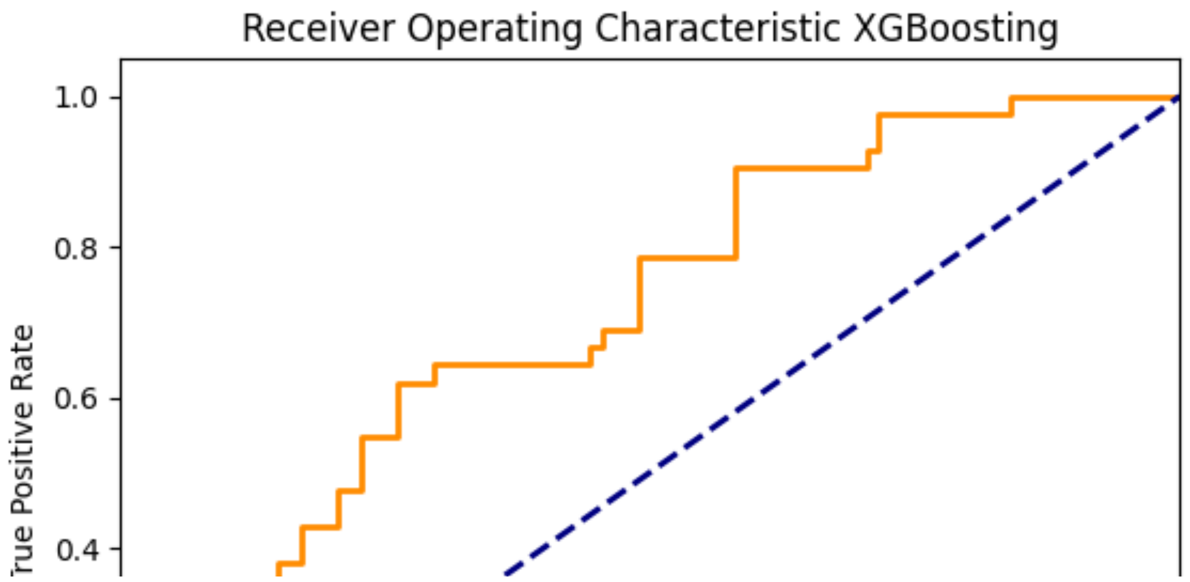
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

  warnings.warn(smsg, 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': 0.1, 'learnin
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

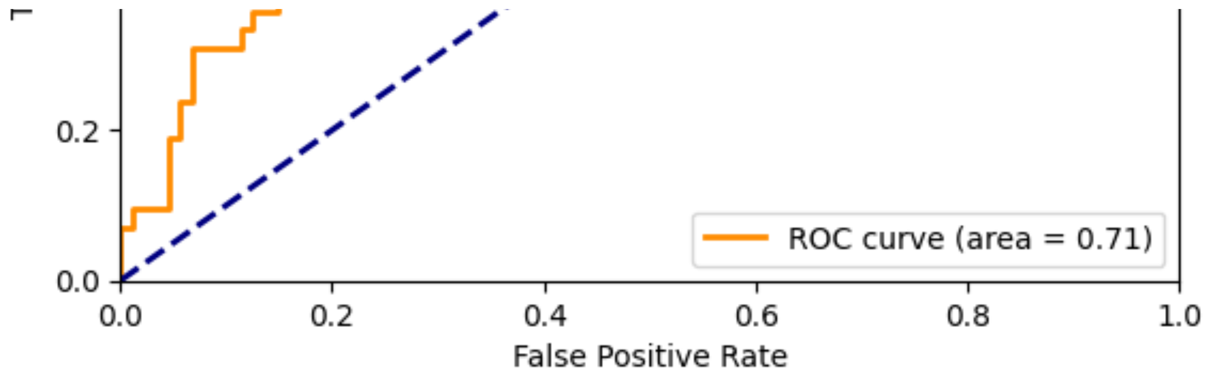
```
Parameters: { "use_label_encoder" } are not used.

  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)

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.0454545
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7110389610389611
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7021696252465484, Sensitivity: 0.0, Specificity: 1.0
Metrics for manual threshold 0.3:
Accuracy: 0.6153846153846154, Sensitivity: 0.6428571428571429, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5230769230769231, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6153846153846154, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7076923076923077, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for XGBoost
```




N surfaces
CARS Initial
Repaired baseline
Missing
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
SHAP value (impact on model output)
Low
Fe


Confusion Matrix XGBoosting
Actual Success
Actual Failure
53
35
15
27
Predicted Success
Predicted Failure
50
45
40
35
30
25
20
15


Receiver Operating Characteristic XGBoosting
1.0
0.8
0.6
0.4
True Positive Rate

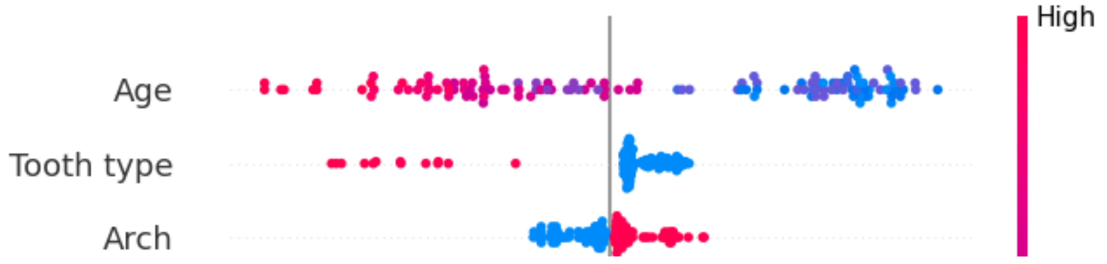


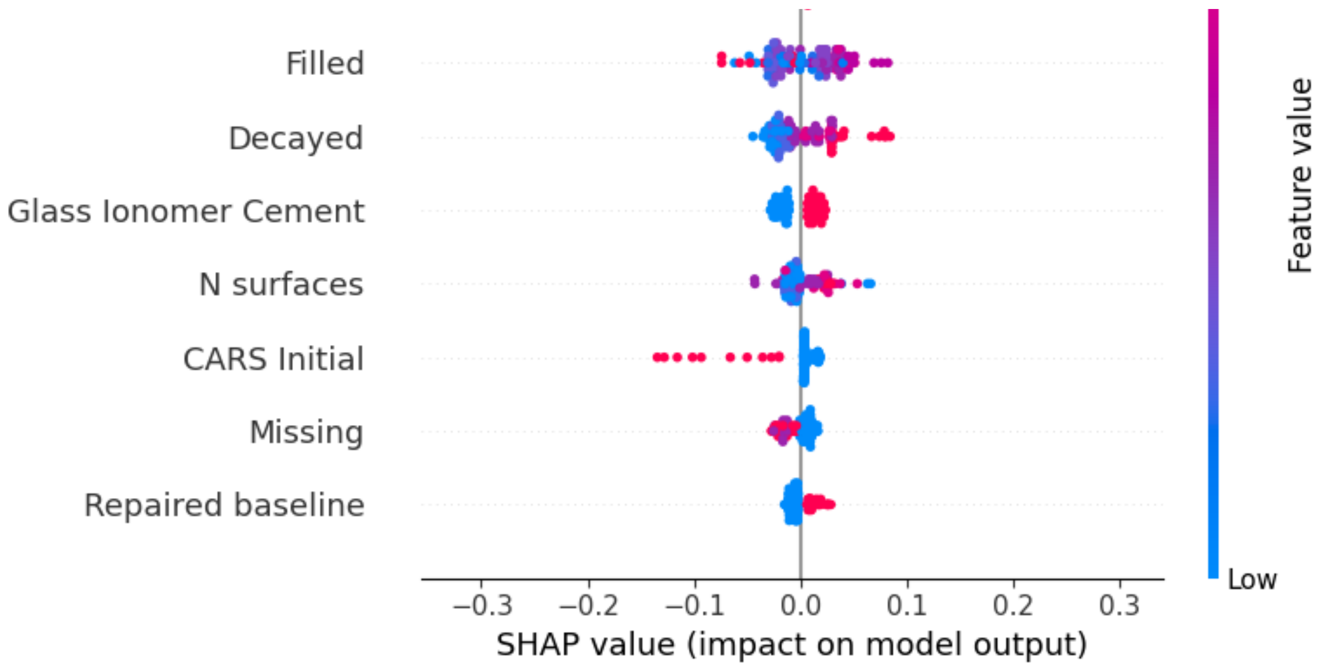
```
Running evaluation with seed 44
Inside evaluate_xgboost function
Evaluating XGBoost...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

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

  warnings.warn(smsg, UserWarning)
```

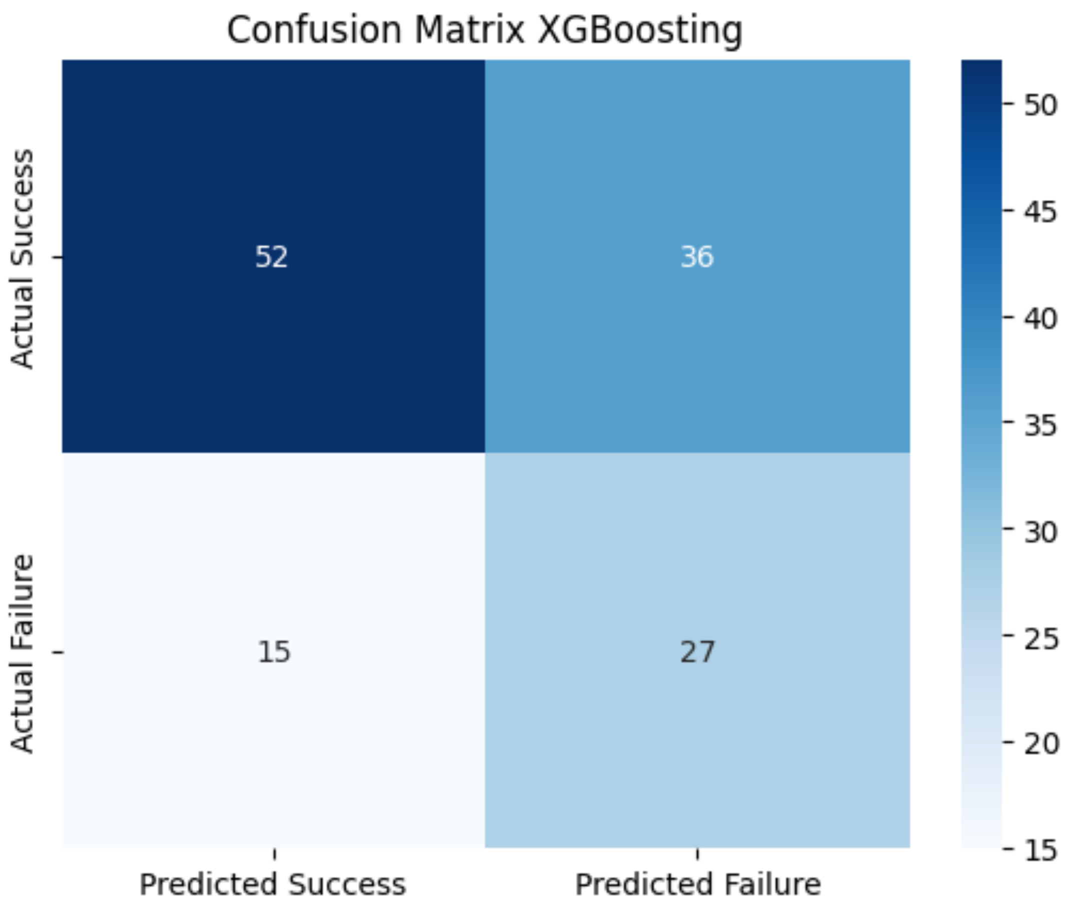
```
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

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

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

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

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

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

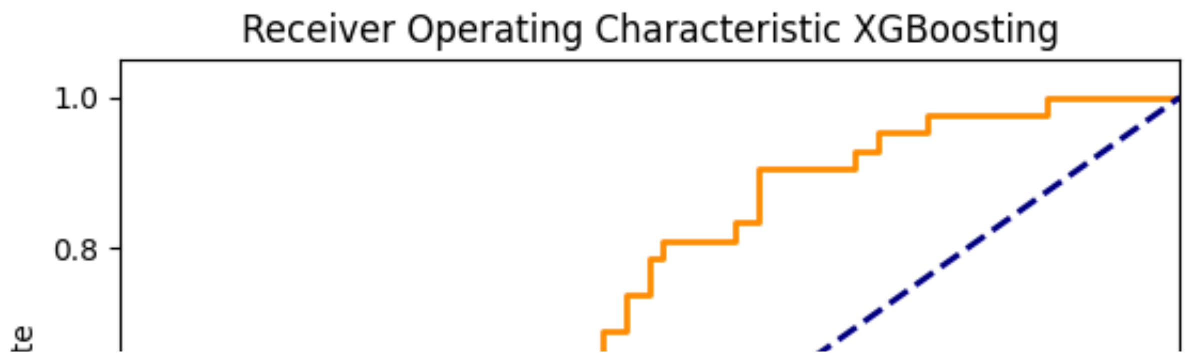
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

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

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

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

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

  warnings.warn(smsg, UserWarning)
```

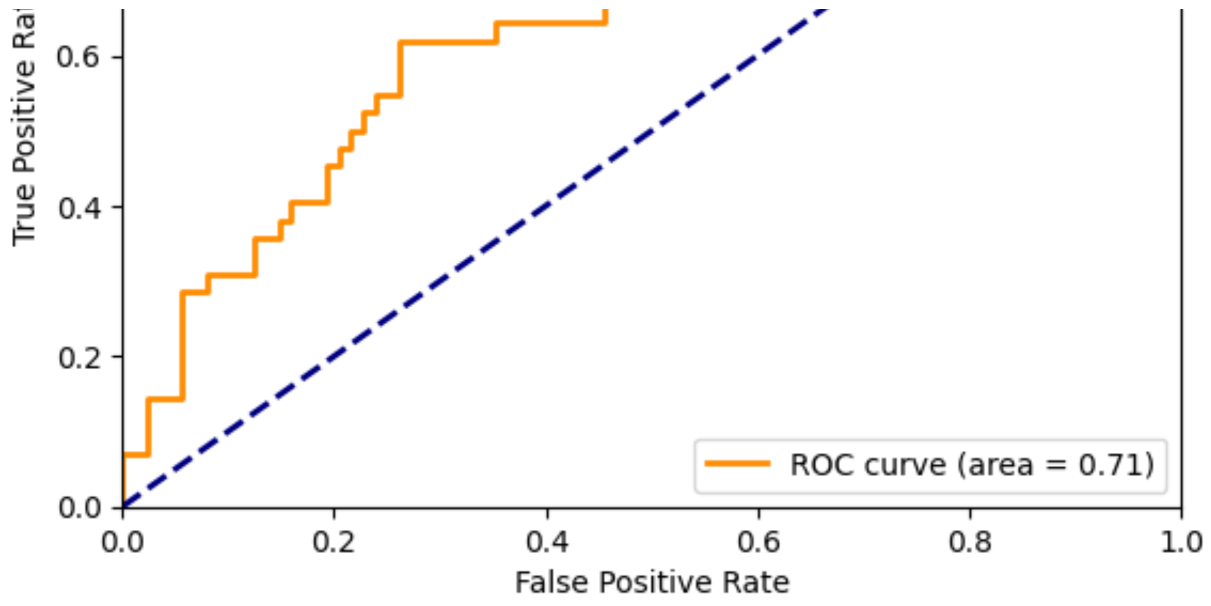
```
Best parameters for XGBoost: {'colsample_bytree': 1, 'gamma': 0.1, 'learnin
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

  warnings.warn(smsg, UserWarning)

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.0227272
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7080627705627704
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7021696252465484, Sensitivity: 0.0, Specificity: 1.0
Metrics for manual threshold 0.3:
Accuracy: 0.6076923076923076, Sensitivity: 0.6428571428571429, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5307692307692308, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6076923076923076, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7153846153846154, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for XGBoost
```




Filled
Decayed
Glass Ionomer Cement
N surfaces
CARS Initial
Missing
Repaired baseline
−0.3 −0.2 −0.1 0.0 0.1 0.2 0.3
SHAP value (impact on model output)
Feature value
Low


Confusion Matrix XGBoosting
Actual Success
Actual Failure
Predicted Success
Predicted Failure
52
36
15
27
50 45 40 35 30 25 20 15


Receiver Operating Characteristic XGBoosting
1.0
0.8
te

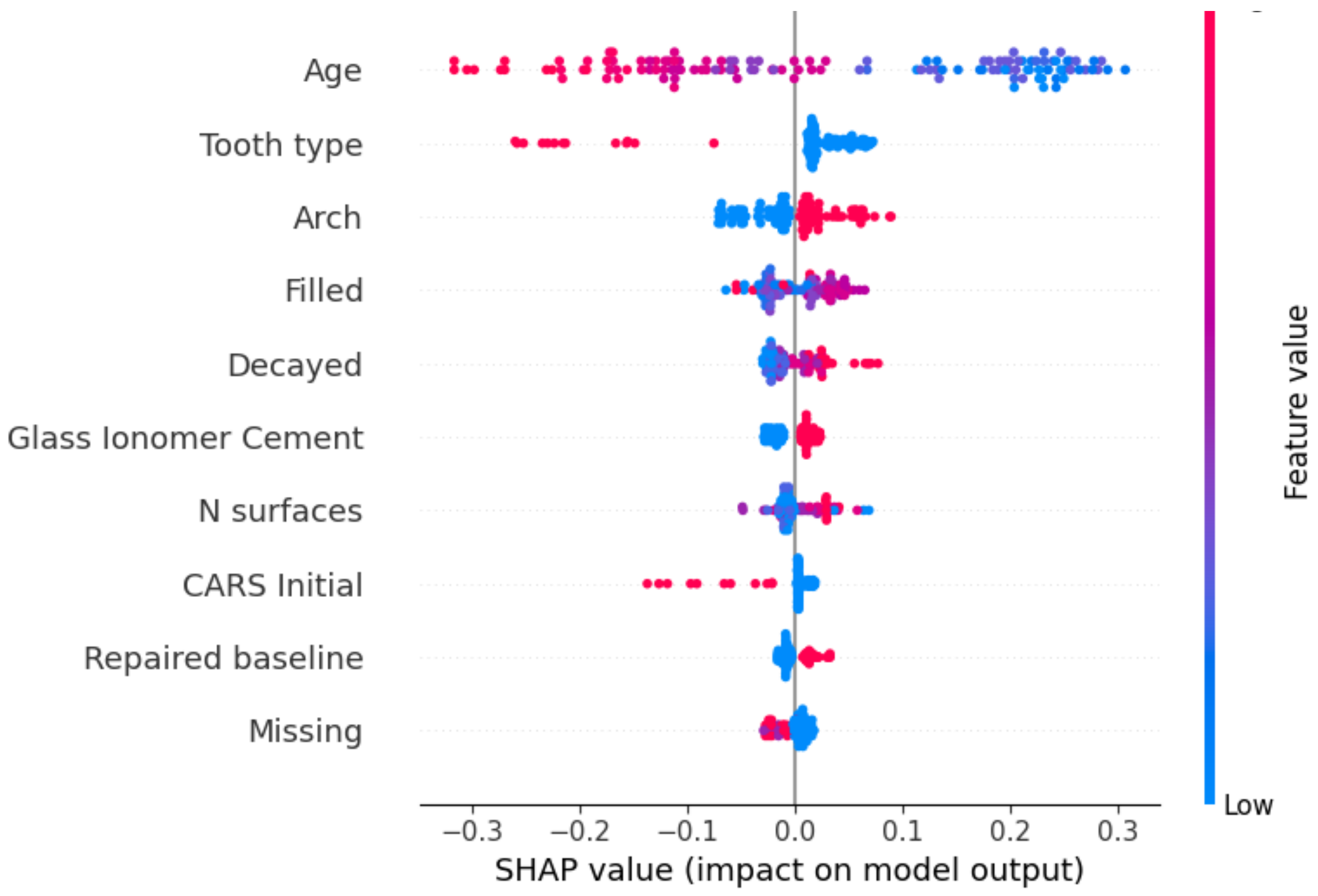


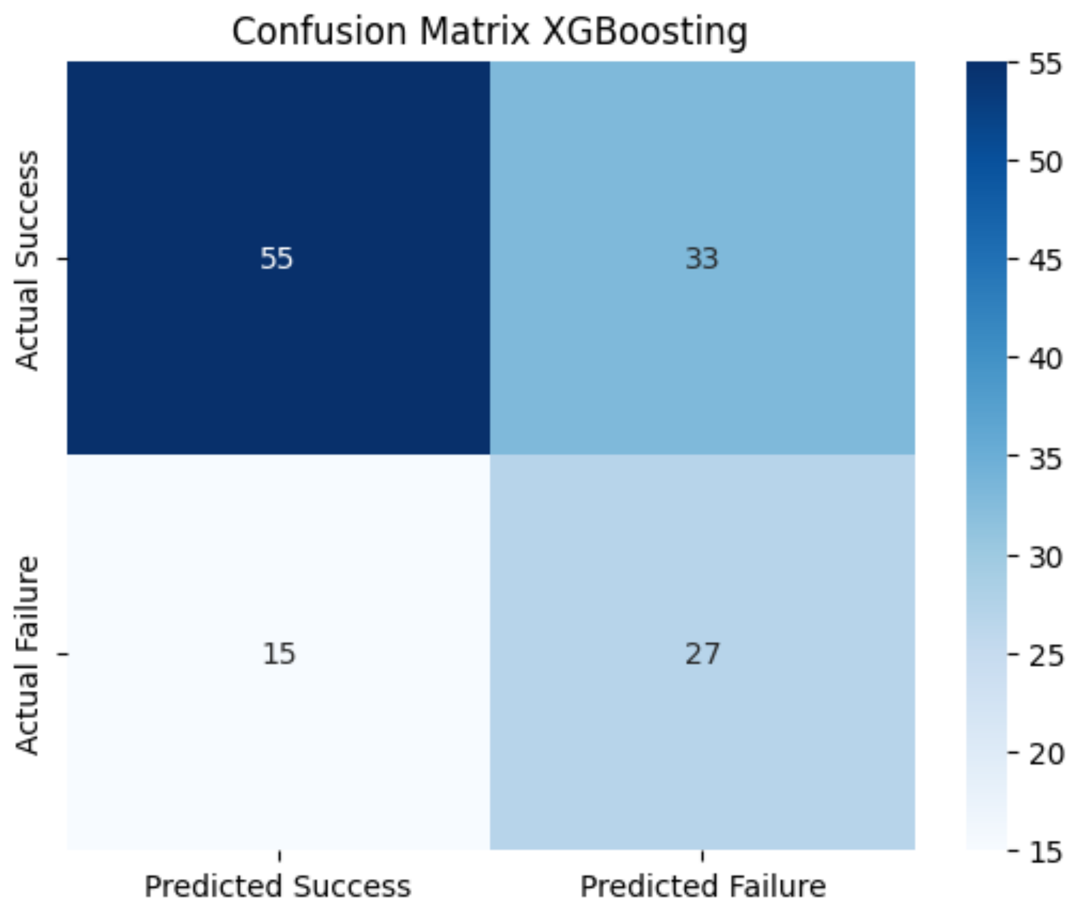
```
Running evaluation with seed 45
Inside evaluate_xgboost function
Evaluating XGBoost...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

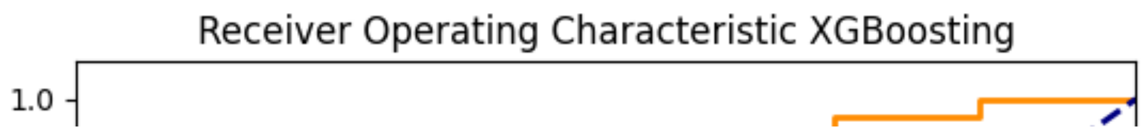
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

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

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

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

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

  warnings.warn(smsg, UserWarning)
```

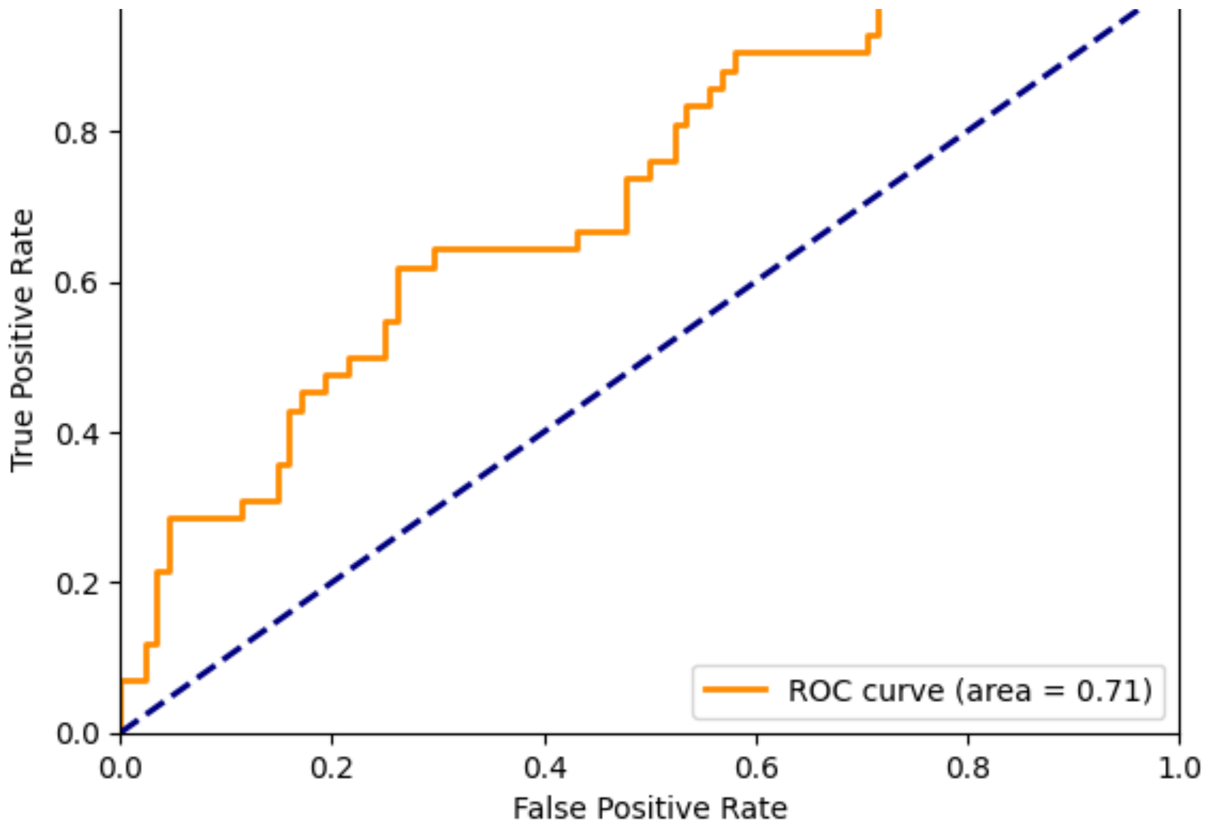
```
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

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

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

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

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

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

```
  warnings.warn(smsg, 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': 0.1, 'learnin
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

  warnings.warn(smsg, UserWarning)

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.0340909
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7142857142857143
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7021696252465484, Sensitivity: 0.0, Specificity: 1.0
Metrics for manual threshold 0.3:
Accuracy: 0.6307692307692307, Sensitivity: 0.6428571428571429, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5384615384615384, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6307692307692307, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7307692307692307, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for XGBoost
```


Age
Tooth type
Arch
Filled
Decayed
Glass Ionomer Cement
N surfaces
CARS Initial
Repaired baseline
Missing
−0.3 −0.2 −0.1 0.0 0.1 0.2 0.3
SHAP value (impact on model output)
Feature value
Low


Confusion Matrix XGBoosting
Actual Success
Actual Failure
Predicted Success
Predicted Failure
55
33
15
27
55 50 45 40 35 30 25 20 15


Receiver Operating Characteristic XGBoosting
1.0



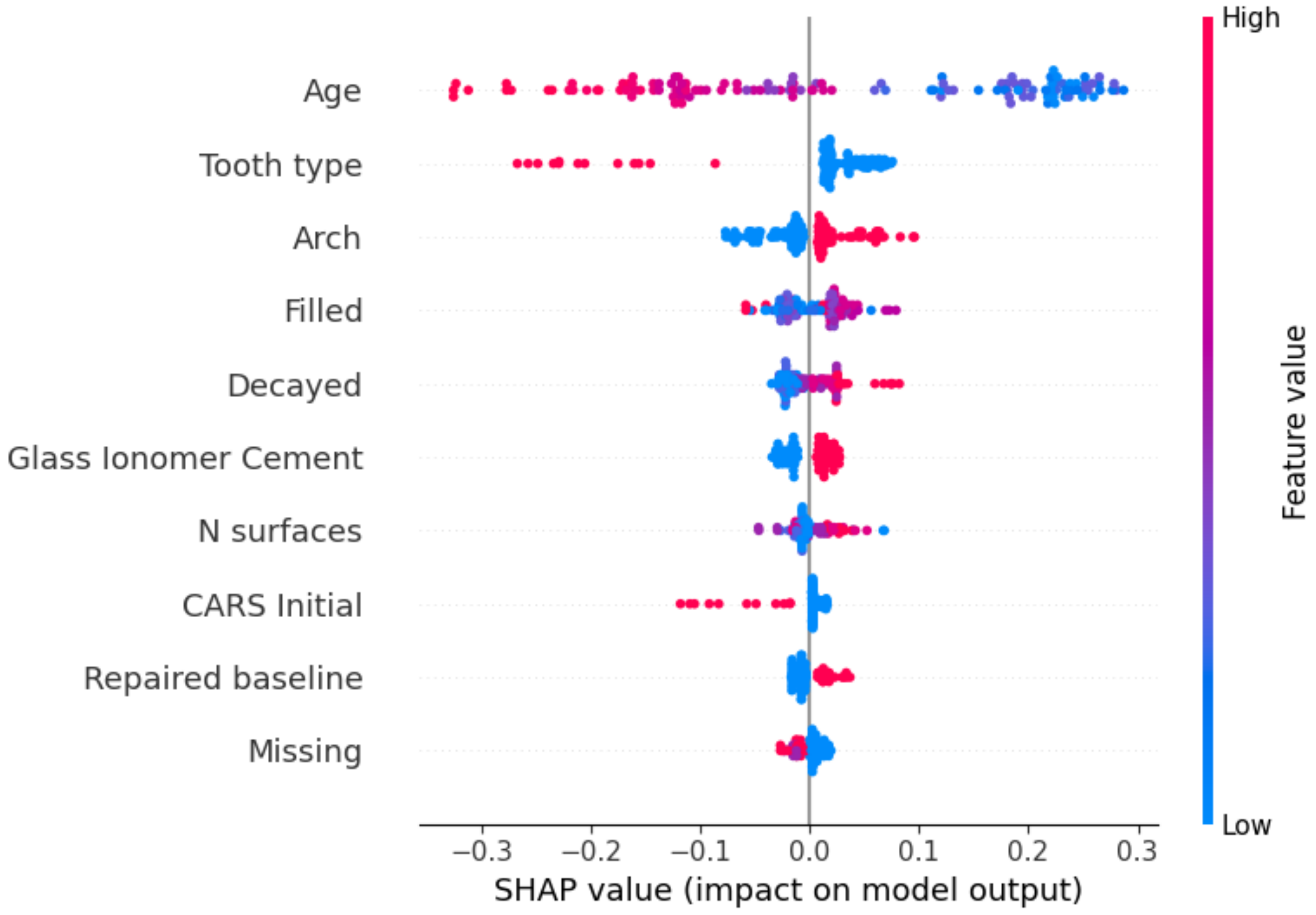
```
Running evaluation with seed 46
Inside evaluate_xgboost function
Evaluating XGBoost...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

  warnings.warn(smsg, UserWarning)
```

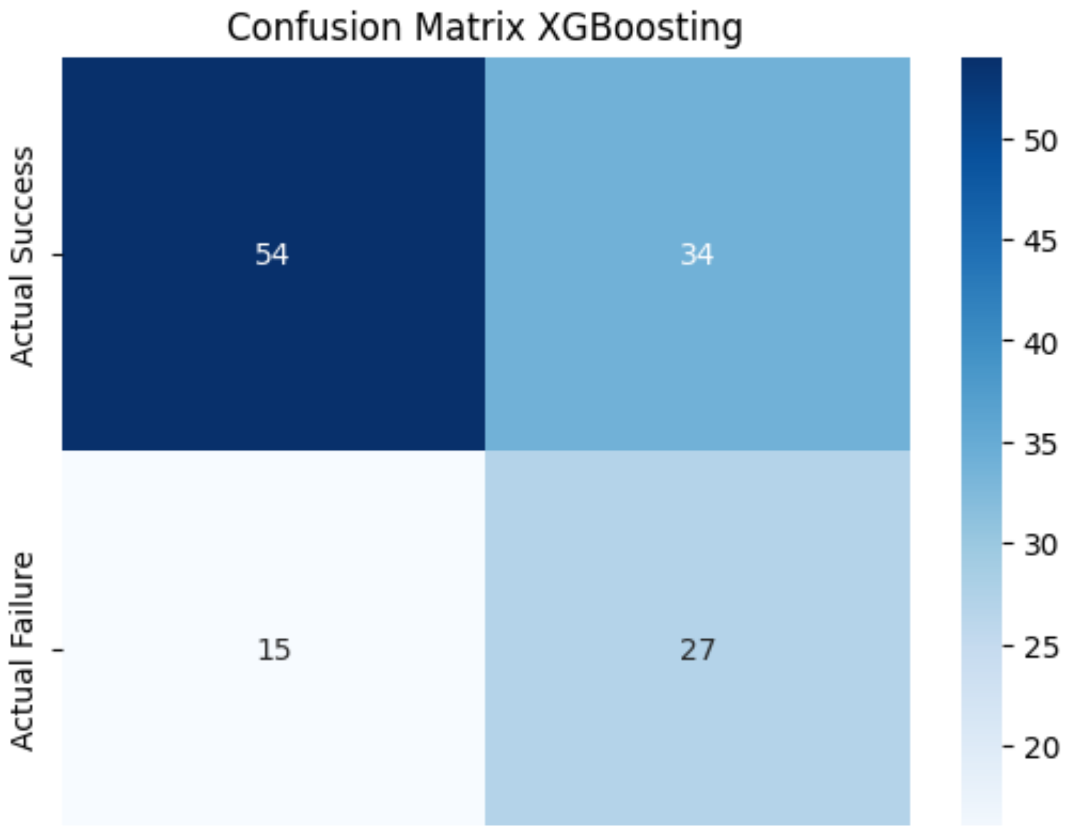
```
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

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

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

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

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

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

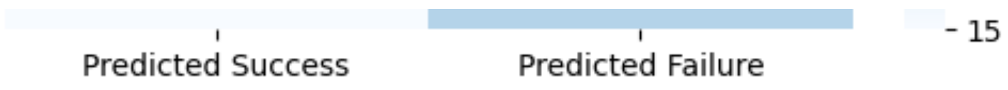
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

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

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

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

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

  warnings.warn(smsg, UserWarning)
```

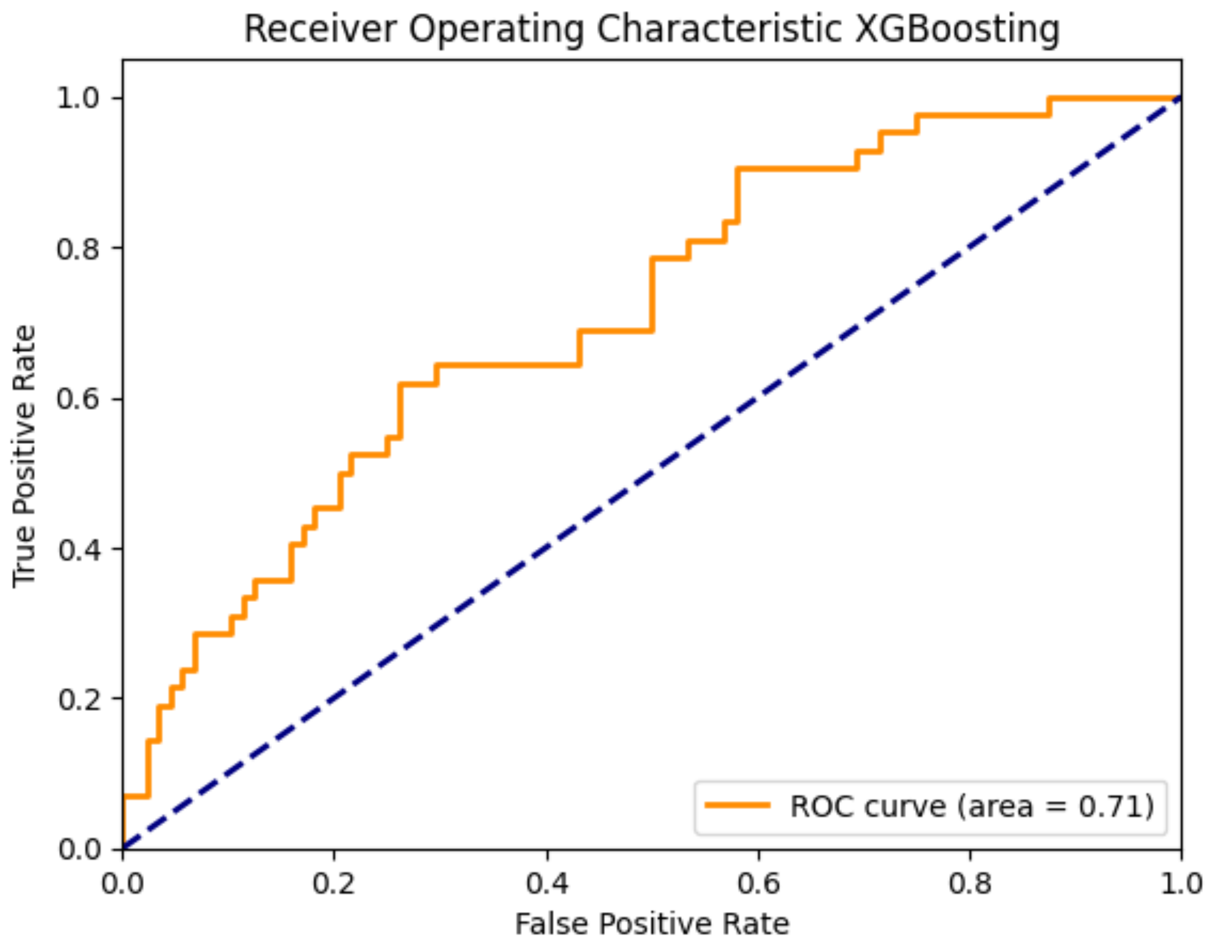
```
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

  warnings.warn(smsg, 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': 0.1, 'learnin
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

  warnings.warn(smsg, UserWarning)

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.0227272
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.712391774891775
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7021696252465484, Sensitivity: 0.0, Specificity: 1.0
Metrics for manual threshold 0.3:
Accuracy: 0.6230769230769231, Sensitivity: 0.6428571428571429, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5307692307692308, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6230769230769231, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6923076923076923, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7230769230769231, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
```

```
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for XGBoost
```







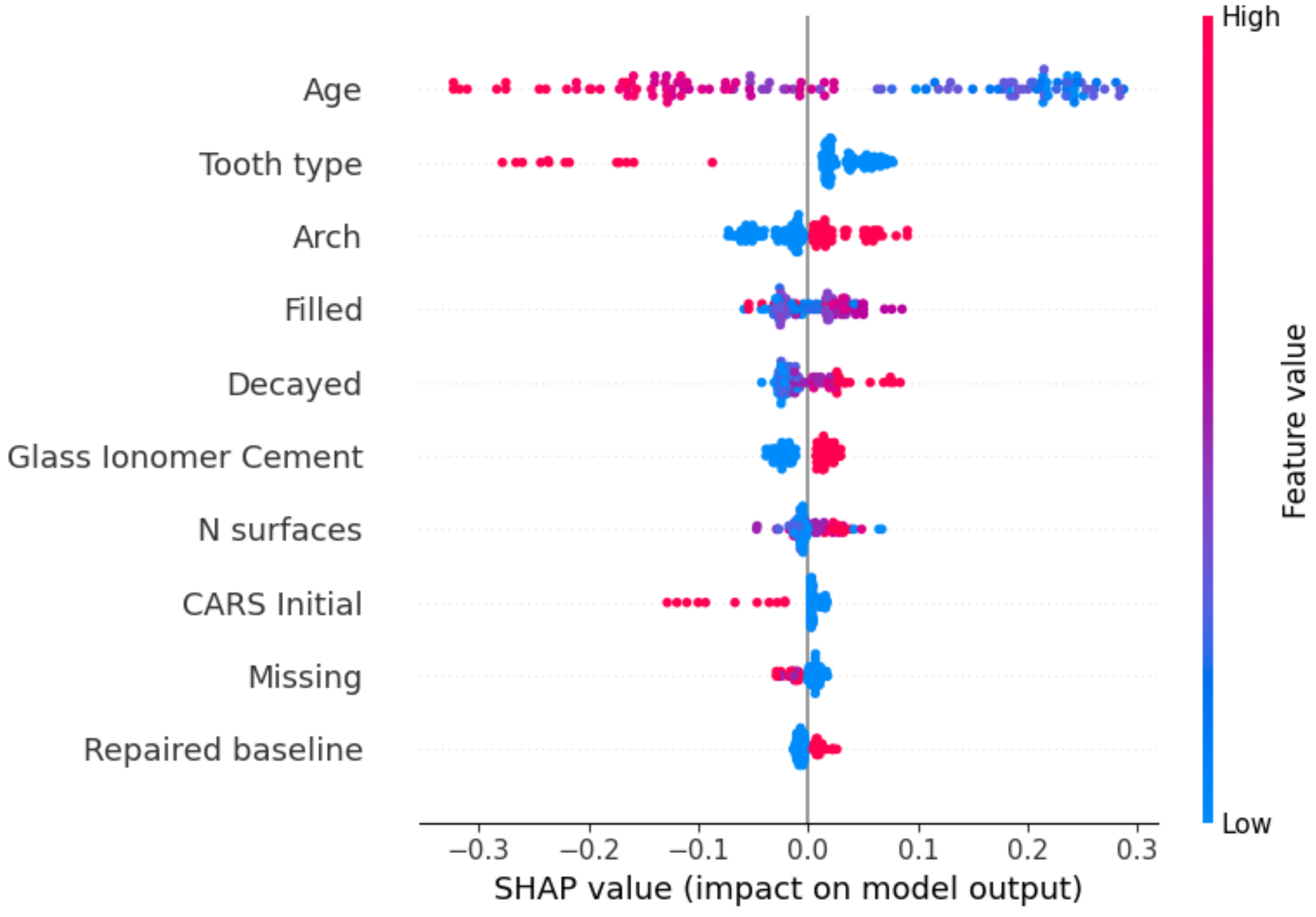


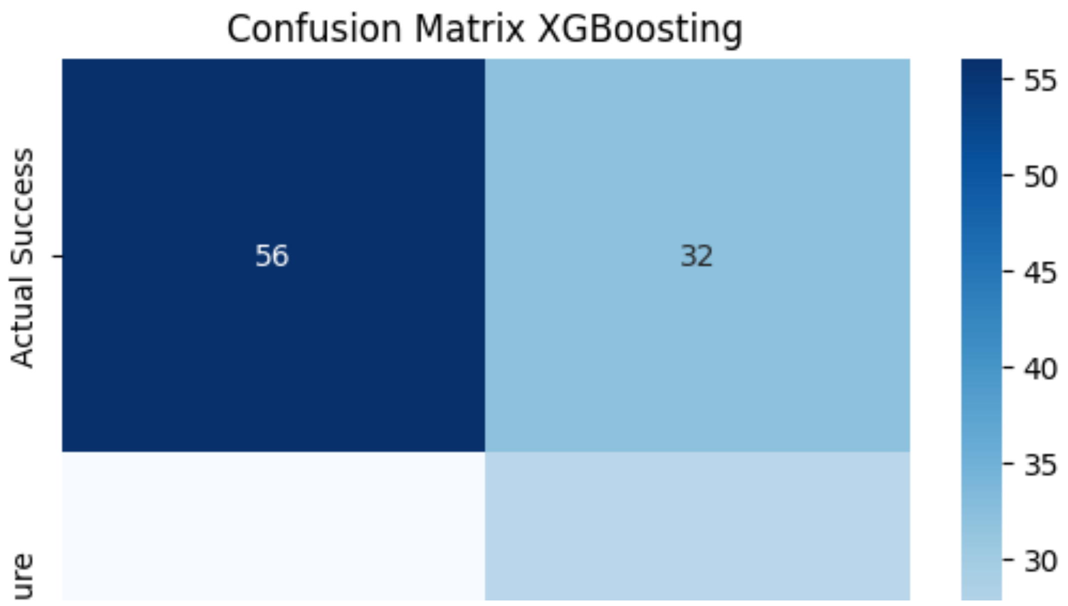
```
Running evaluation with seed 47
Inside evaluate_xgboost function
Evaluating XGBoost...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

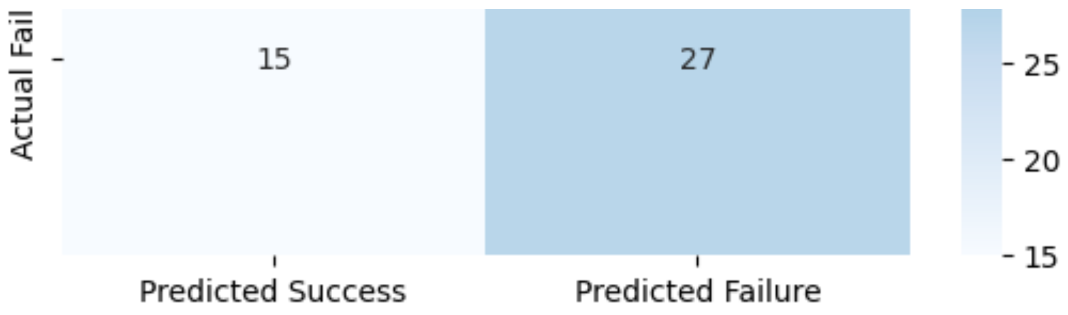
```
Parameters: { "use_label_encoder" } are not used.

  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.
```

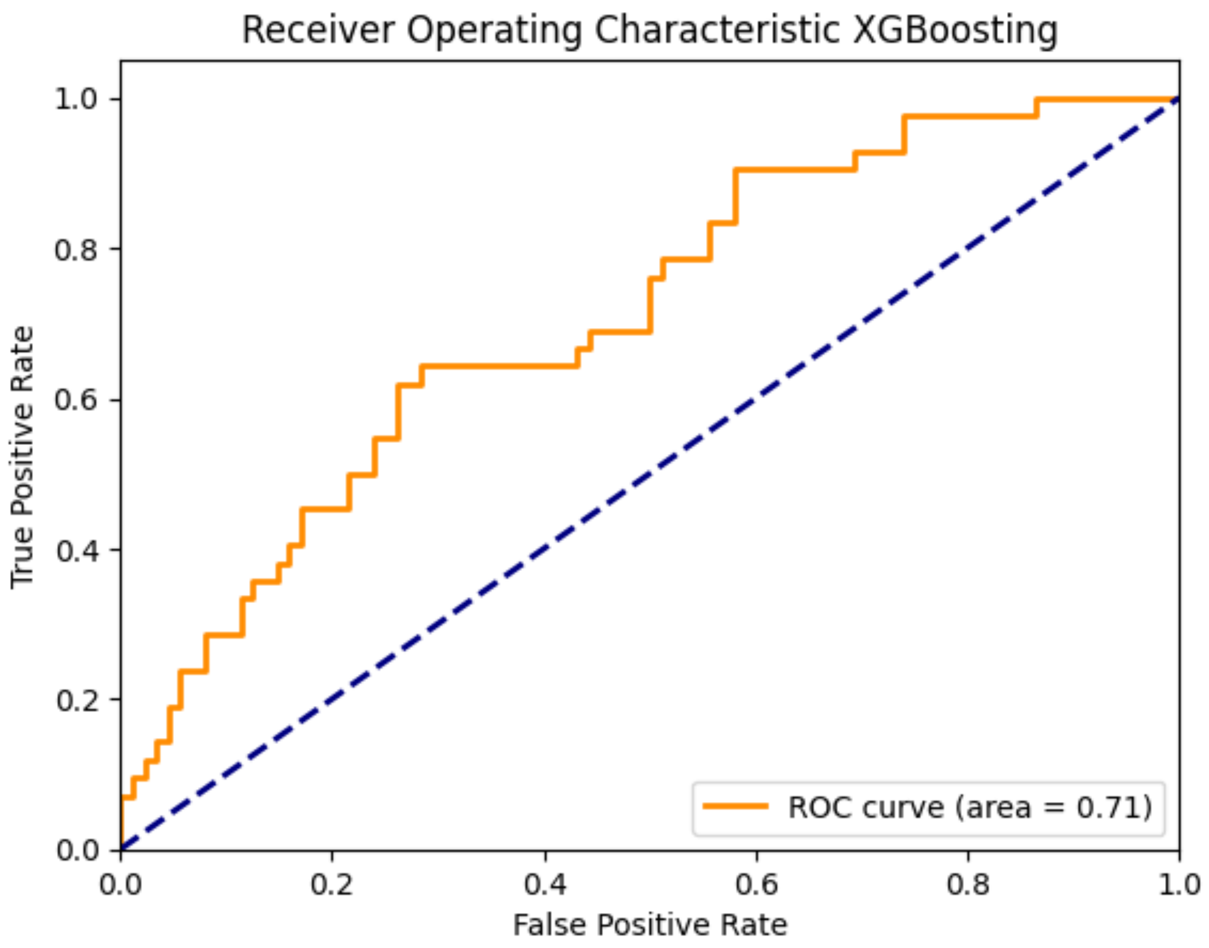
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

  warnings.warn(smsg, 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': 0.1, 'learnin
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

  warnings.warn(smsg, UserWarning)

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.0227272
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7099567099567101
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7021696252465484, Sensitivity: 0.0, Specificity: 1.0
Metrics for manual threshold 0.3:
Accuracy: 0.6384615384615384, Sensitivity: 0.6428571428571429, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5307692307692308, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6384615384615384, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7076923076923077, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
```

```
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for XGBoost
```







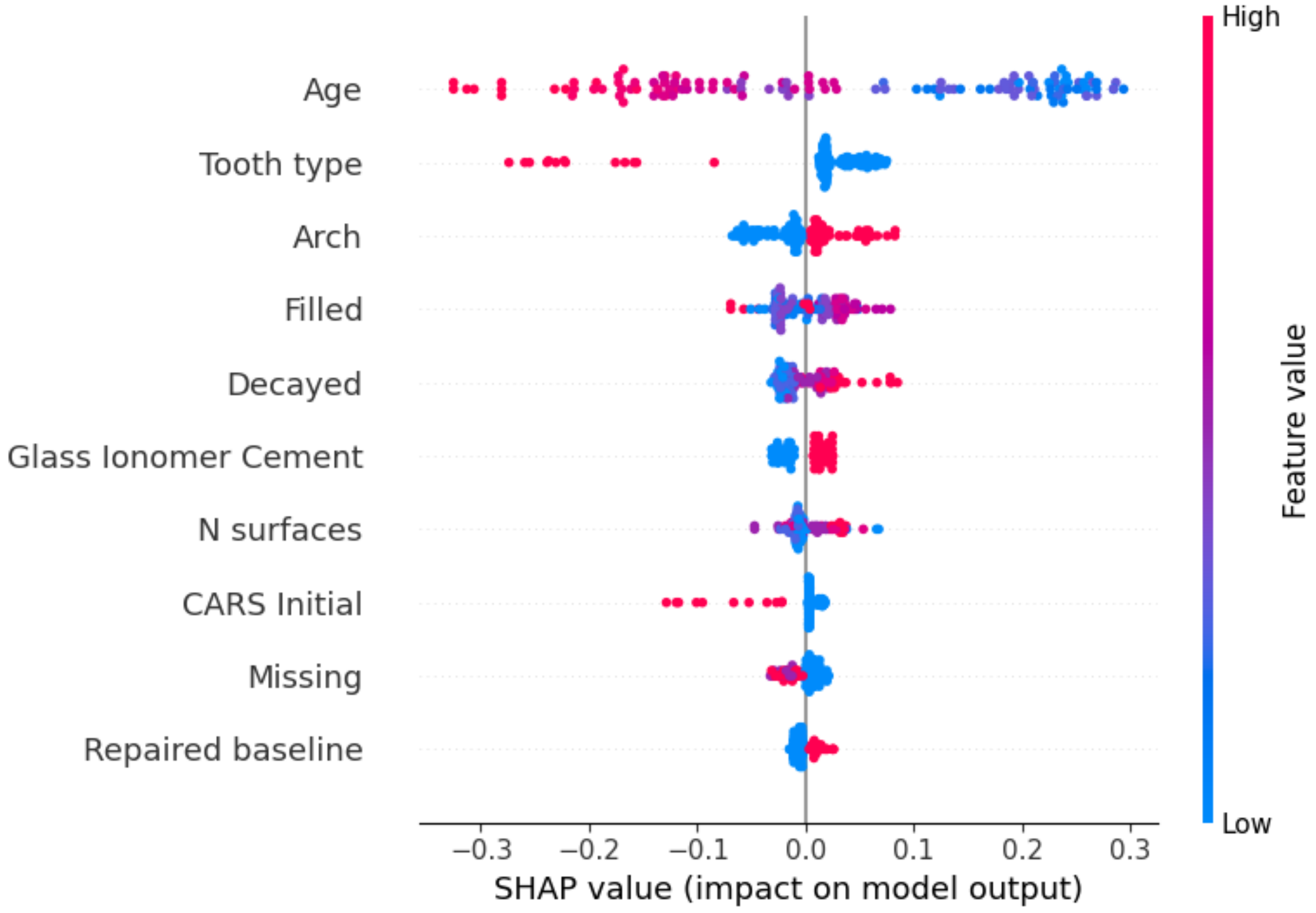


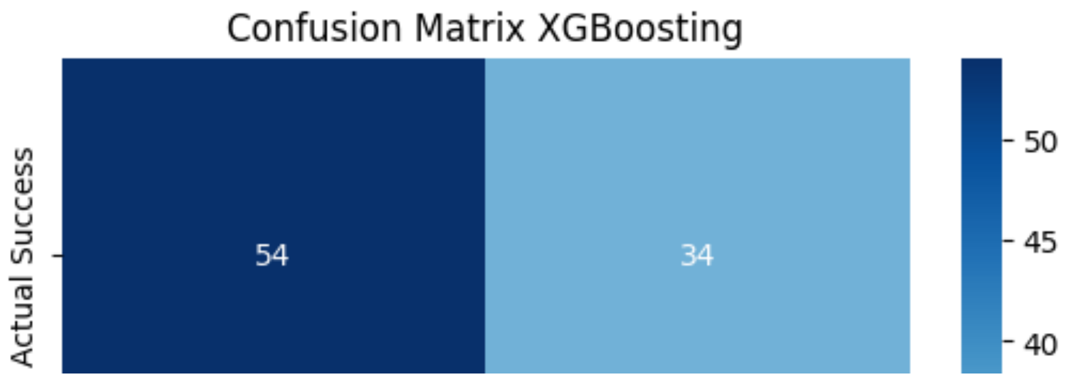
```
Running evaluation with seed 48
Inside evaluate_xgboost function
Evaluating XGBoost...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

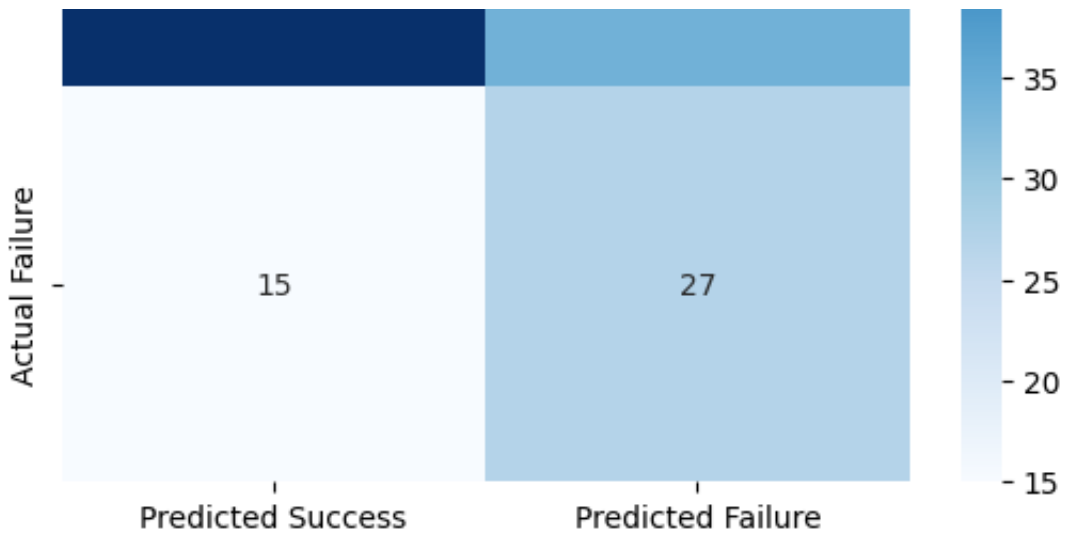
```
  warnings.warn(smsg, UserWarning)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

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

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

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

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

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

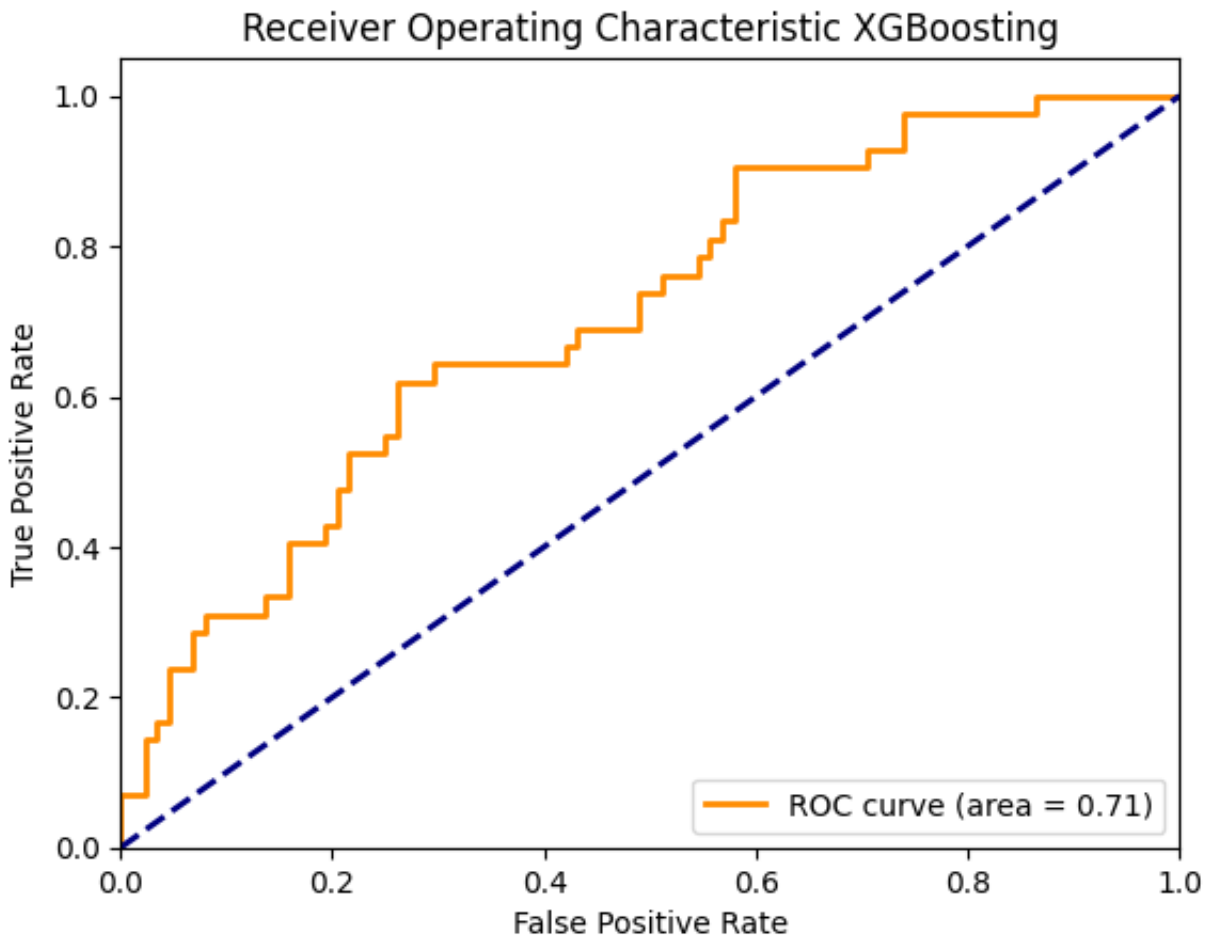
```
Parameters: { "use_label_encoder" } are not used.

  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)
Best parameters for XGBoost: {'colsample_bytree': 1, 'gamma': 0.1, 'learnin
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

  warnings.warn(smsg, UserWarning)

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.0227272
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.708874458874459
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7021696252465484, Sensitivity: 0.0, Specificity: 1.0
Metrics for manual threshold 0.3:
Accuracy: 0.6230769230769231, Sensitivity: 0.6428571428571429, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.3384615384615385, 'Sensitivity': 1
```

```
Threshold: 0.20, Metrics: {'Accuracy': 0.3384615384615385, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5384615384615384, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6230769230769231, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7076923076923077, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for XGBoost
```





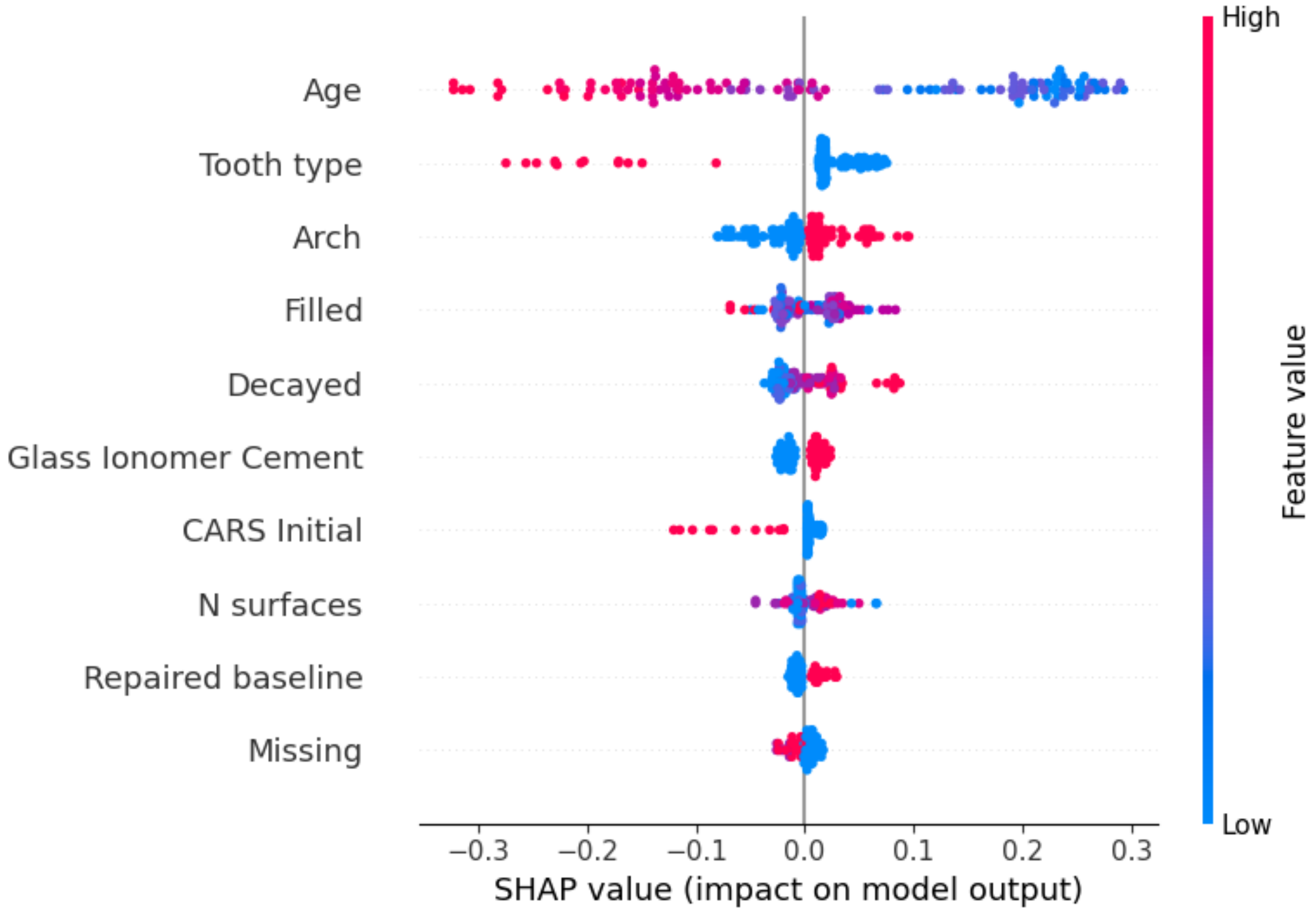


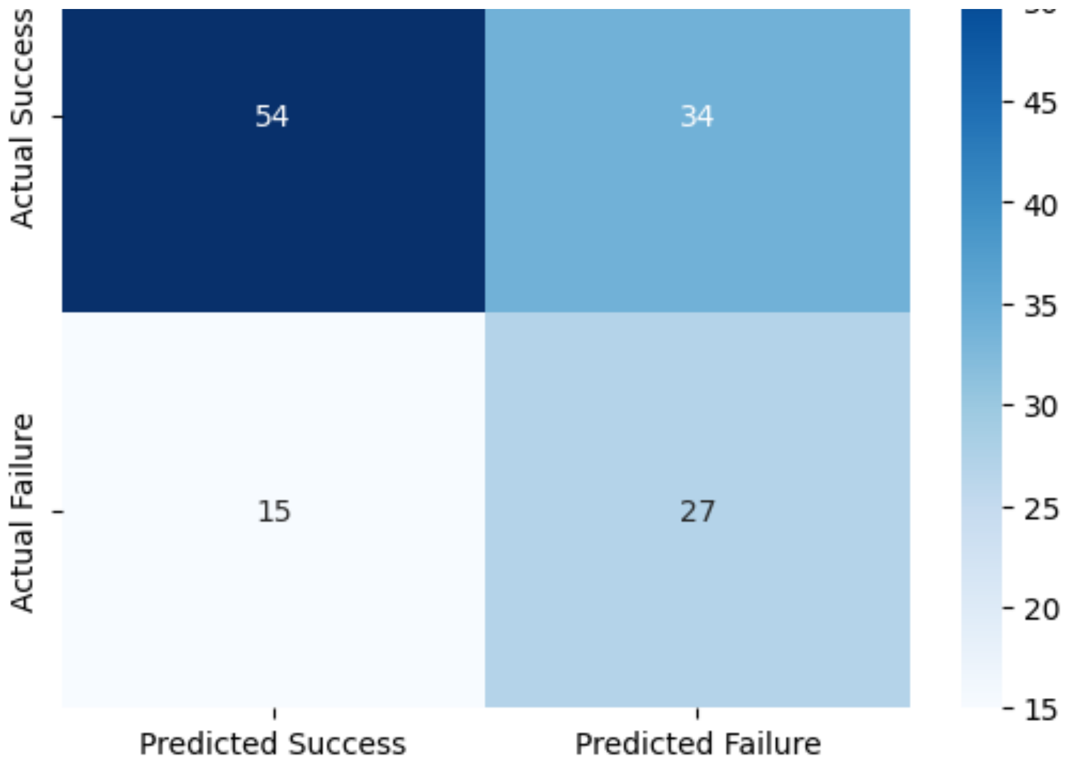


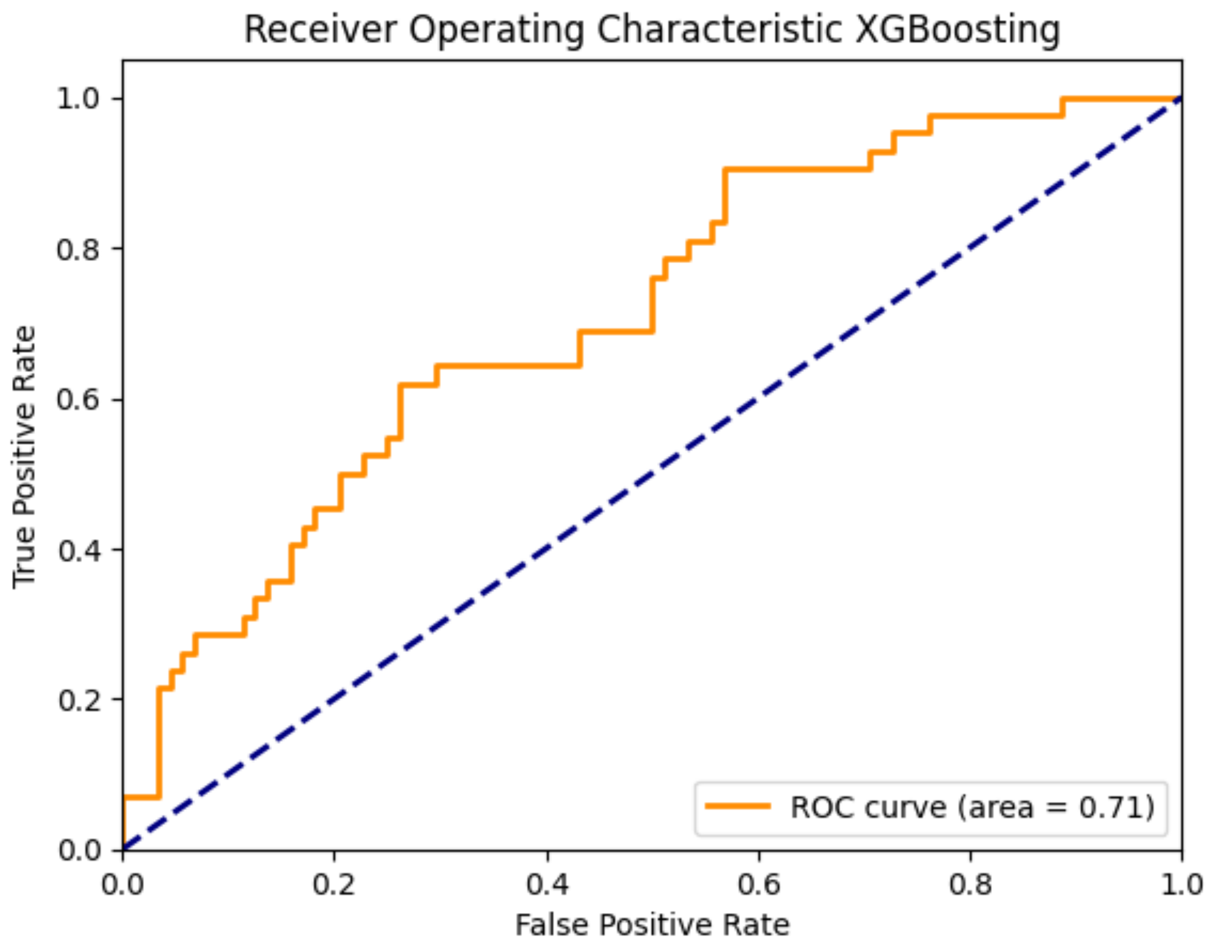
```
Running evaluation with seed 49
Inside evaluate_xgboost function
Evaluating XGBoost...
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

  warnings.warn(smsg, 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': 0.1, 'learnin
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [
Parameters: { "use_label_encoder" } are not used.

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

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

  warnings.warn(smsg, UserWarning)

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.03409090909090909, 0.03409090909090909, 0.034090909
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.711038961038961
--- Fim dos Dados ROC ---
```

```
Training - Accuracy: 0.7021696252465484, Sensitivity: 0.0, Specificity: 1.0
Metrics for manual threshold 0.3:
Accuracy: 0.6230769230769231, Sensitivity: 0.6428571428571429, Specificity:
Threshold: 0.10, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.3230769230769231, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.34615384615384615, 'Sensitivity':
Threshold: 0.25, Metrics: {'Accuracy': 0.5538461538461539, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.6230769230769231, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6923076923076923, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.6846153846153846, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.65, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.75, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.80, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.85, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.90, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 0.95, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
Threshold: 1.00, Metrics: {'Accuracy': 0.676923076923077, 'Sensitivity': 0.
SHAP Summary for XGBoost
```







```
Aggregated Test Set Metrics Across Seeds:
   accuracy  sensitivity  specificity        f1   roc_auc
0  0.615385     0.642857     0.602273  0.519231  0.708874
1  0.623077     0.642857     0.613636  0.524272  0.709957
2  0.630769     0.642857     0.625000  0.529412  0.710768
```

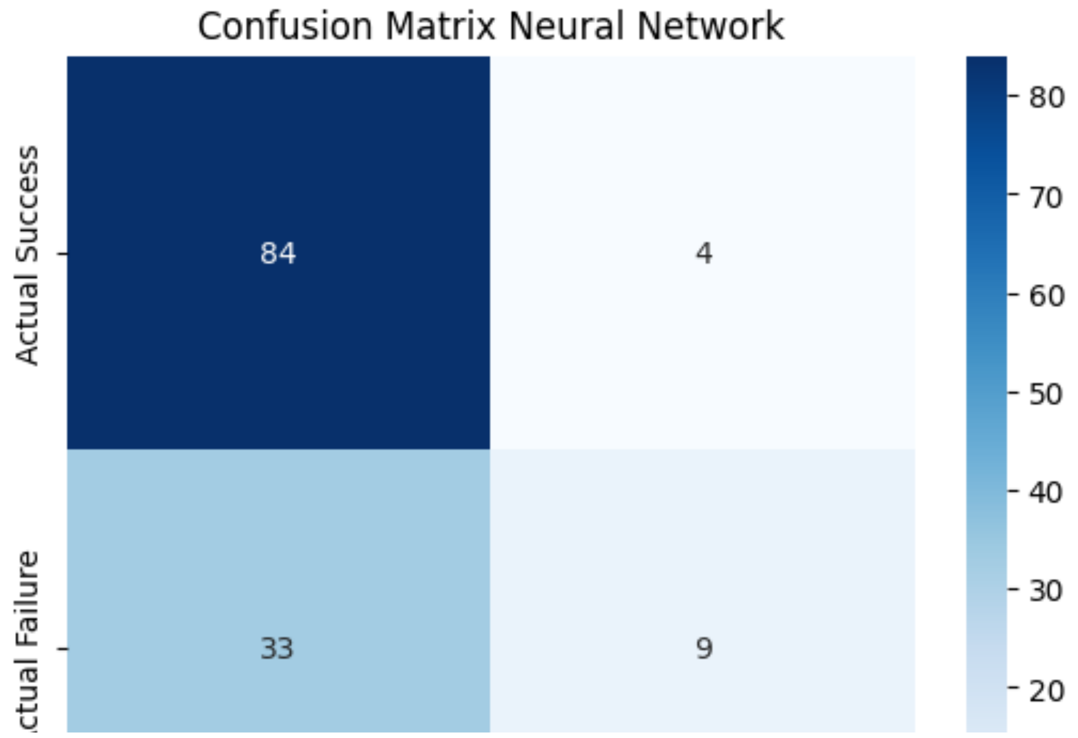
```
3  0.615385     0.642857     0.602273  0.519231  0.711039
4  0.607692     0.642857     0.590909  0.514286  0.708063
5  0.630769     0.642857     0.625000  0.529412  0.714286
6  0.623077     0.642857     0.613636  0.524272  0.712392
7  0.638462     0.642857     0.636364  0.534653  0.709957
8  0.623077     0.642857     0.613636  0.524272  0.708874
9  0.623077     0.642857     0.613636  0.524272  0.711039

Summary of Test Set Metrics (Mean, Standard Error, 95% Confidence Interval)
Accuracy: Mean = 0.623, SE = 0.003, 95% CI = [0.617, 0.629]
Sensitivity: Mean = 0.643, SE = 0.000, 95% CI = [0.643, 0.643]
Specificity: Mean = 0.614, SE = 0.004, 95% CI = [0.604, 0.623]
F1: Mean = 0.524, SE = 0.002, 95% CI = [0.520, 0.529]
Roc_auc: Mean = 0.711, SE = 0.001, 95% CI = [0.709, 0.712]
```

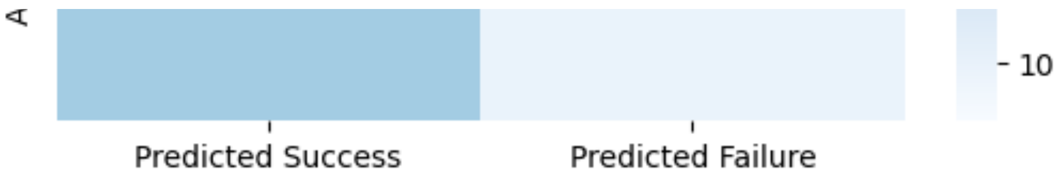
```
# Set seeds for reproducibility
seed_value = 42
np.random.seed(seed_value)
random.seed(seed_value)
tf.random.set_seed(seed_value)

# Define a function to build, train, and evaluate a neural network model.
def evaluate_neural_network(X_train, y_train, X_test, y_test, threshold_list):
    # Initialize the neural network model with specified layers.
    model = Sequential([
        Dense(128, activation='relu', kernel_regularizer=l2(0.01), input_shape=
        BatchNormalization(),
        Dropout(0.3),
        Dense(64, activation='relu', kernel_regularizer=l2(0.01)),
        BatchNormalization(),
        Dropout(0.3),
        Dense(1, activation='sigmoid')
    ])

    # Compile the model specifying the optimizer, loss function, and metrics.
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accur

    # Define callbacks for early stopping and learning rate reduction.
    early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_bes
    reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, m

    # Train the model with a validation split, epochs, batch size, and callback
    model.fit(X_train, y_train, validation_split=0.2, epochs=15, batch_size=32,
    # Training set evaluation
    y_train_probs = model.predict(X_train).ravel()
    # Use the last threshold in the list for training evaluation
    y_train_pred = (y_train_probs >= threshold_list[-1]).astype(int)
    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)

    # Test set evaluation for multiple thresholds
    y_probs = model.predict(X_test).ravel()

    # Calculate FPR, TPR and AUC
    fpr, tpr, _ = roc_curve(y_test, y_probs)
    roc_auc_val = auc(fpr, tpr)

    print("\n--- Dados ROC para copiar ---")
    print("FPR =", fpr.tolist())
    print("TPR =", tpr.tolist())
    print("AUC =", roc_auc_val)
    print("--- Fim dos Dados ROC ---\n")

    thresholds_metrics = []
    for threshold in threshold_list:
        y_pred = (y_probs >= threshold).astype(int)
        acc = accuracy_score(y_test, y_pred)
        sens = sensitivity(y_test, y_pred)
        spec = specificity(y_test, y_pred)
        thresholds_metrics.append({
            'threshold': threshold,
            'accuracy': acc,
            'sensitivity': sens,
            'specificity': spec
        })

    print(f"Training - Accuracy: {train_acc:.4f}, Sensitivity: {train_sens:.4f}

    # Test set ROC AUC (same regardless of threshold)
    test_roc_auc = roc_auc_score(y_test, y_probs)

    # Print test set metrics for each threshold
    for metrics in thresholds_metrics:
        print(f"Threshold: {metrics['threshold']:.2f}, Accuracy: {metrics['accu

    return model, train_acc, train_sens, train_spec, train_f1, train_roc_auc, t
```

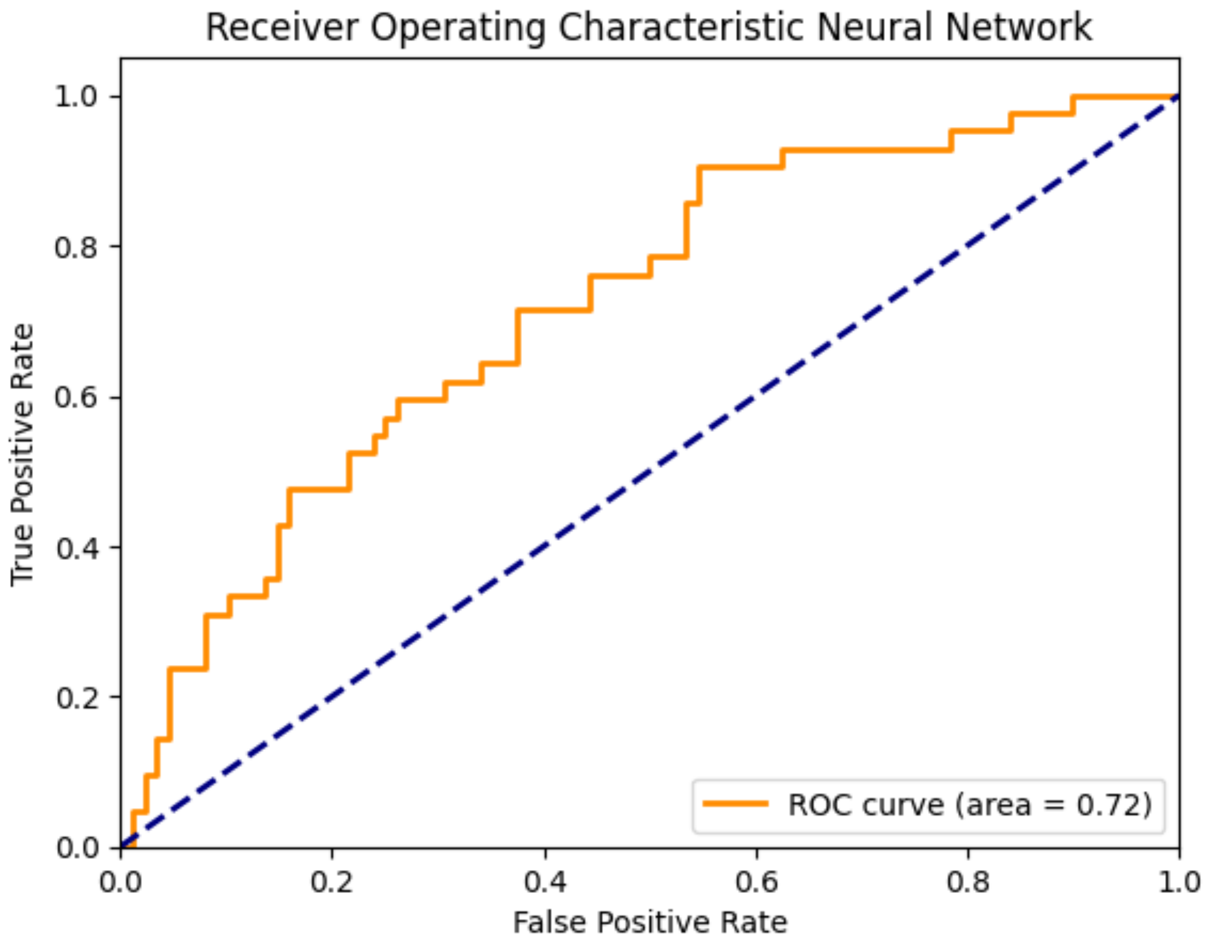
```
# Plotting functions for confusion matrix and ROC curve visualization.
def plot_confusion_matrix(y_true, y_pred):
    matrix = confusion_matrix(y_true, y_pred)
    sns.heatmap(matrix, annot=True, fmt='d', cmap='Blues',
                xticklabels=['Predicted Success', 'Predicted Failure'],
                yticklabels=['Actual Success', 'Actual Failure'])
    plt.title('Confusion Matrix Neural Network')
    plt.show()

def plot_roc_curve(y_true, y_probs):
    fpr, tpr, _ = roc_curve(y_true, y_probs)
    roc_auc_val = 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 Neural Network')
    plt.legend(loc="lower right")
    plt.show()

# Main function where the evaluation process is initiated.
def main(X_train, y_train, X_test, y_test):
    threshold_list = np.arange(0.1, 1.05, 0.05)
    aggregated_metrics = []

    # Loop over a range of seeds
    for seed_value in range(40, 50):
        # Set seed for reproducibility
        np.random.seed(seed_value)
        random.seed(seed_value)
        tf.random.set_seed(seed_value)

        # Evaluate the neural network model
        model, train_acc, train_sens, train_spec, train_f1, train_roc_auc, test

        # Choose a threshold for detailed evaluation (e.g., 0.35)
        chosen_threshold = 0.35
        y_test_probs = model.predict(X_test).ravel()
        y_test_pred = (y_test_probs >= chosen_threshold).astype(int)

        chosen_acc = accuracy_score(y_test, y_test_pred)
        chosen_sens = sensitivity(y_test, y_test_pred)
        chosen_spec = specificity(y_test, y_test_pred)
        chosen_f1 = f1_score(y_test, y_test_pred)
```

```
            print(f"\nMetrics for chosen threshold {chosen_threshold}:")
            print(f"Accuracy: {chosen_acc:.4f}, Sensitivity: {chosen_sens:.4f}, Spe

            # Store metrics from this seed for later aggregation
            test_metrics = {
                "accuracy": chosen_acc,
                "sensitivity": chosen_sens,
                "specificity": chosen_spec,
                "f1": chosen_f1,
                "roc_auc": test_roc_auc
            }
            aggregated_metrics.append(test_metrics)

            plot_confusion_matrix(y_test, y_test_pred)
            plot_roc_curve(y_test, y_test_probs)

        # 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 for
        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)
            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%

        print("\nSummary of Test Set Metrics (Mean, Standard Error, 95% Confidence
        for metric, summary in metrics_summary.items():
            print(f"{metric.capitalize()}: Mean = {summary['Mean']:.4f}, SE = {summ

    # Entry point of the script.
    if __name__ == '__main__':
        # Ensure that X_train, y_train, X_test, y_test are defined before calling m
        main(X_train, y_train, X_test, y_test)
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step

--- Dados ROC para copiar ---
FPR = [0.0, 0.011363636363636364, 0.011363636363636364, 0.022727272727272727
TPR = [0.0, 0.0, 0.047619047619047616, 0.047619047619047616, 0.095238095238
AUC = 0.7205086580086579
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7022, Sensitivity: 0.0000, Specificity: 1.0000, F1:
Threshold: 0.10, Accuracy: 0.3385, Sensitivity: 1.0000, Specificity: 0.0227
Threshold: 0.15, Accuracy: 0.4000, Sensitivity: 0.9762, Specificity: 0.1250
Threshold: 0.20, Accuracy: 0.6000, Sensitivity: 0.9048, Specificity: 0.4545
Threshold: 0.25, Accuracy: 0.6692, Sensitivity: 0.5952, Specificity: 0.7045
Threshold: 0.30, Accuracy: 0.7154, Sensitivity: 0.4286, Specificity: 0.8523
Threshold: 0.35, Accuracy: 0.7154, Sensitivity: 0.2143, Specificity: 0.9545
Threshold: 0.40, Accuracy: 0.6923, Sensitivity: 0.0952, Specificity: 0.9773
Threshold: 0.45, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.50, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.55, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.60, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.65, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.70, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.75, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.80, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.85, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.90, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.95, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 1.00, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step

Metrics for chosen threshold 0.35:
Accuracy: 0.7154, Sensitivity: 0.2143, Specificity: 0.9545, F1: 0.3273, ROC
```

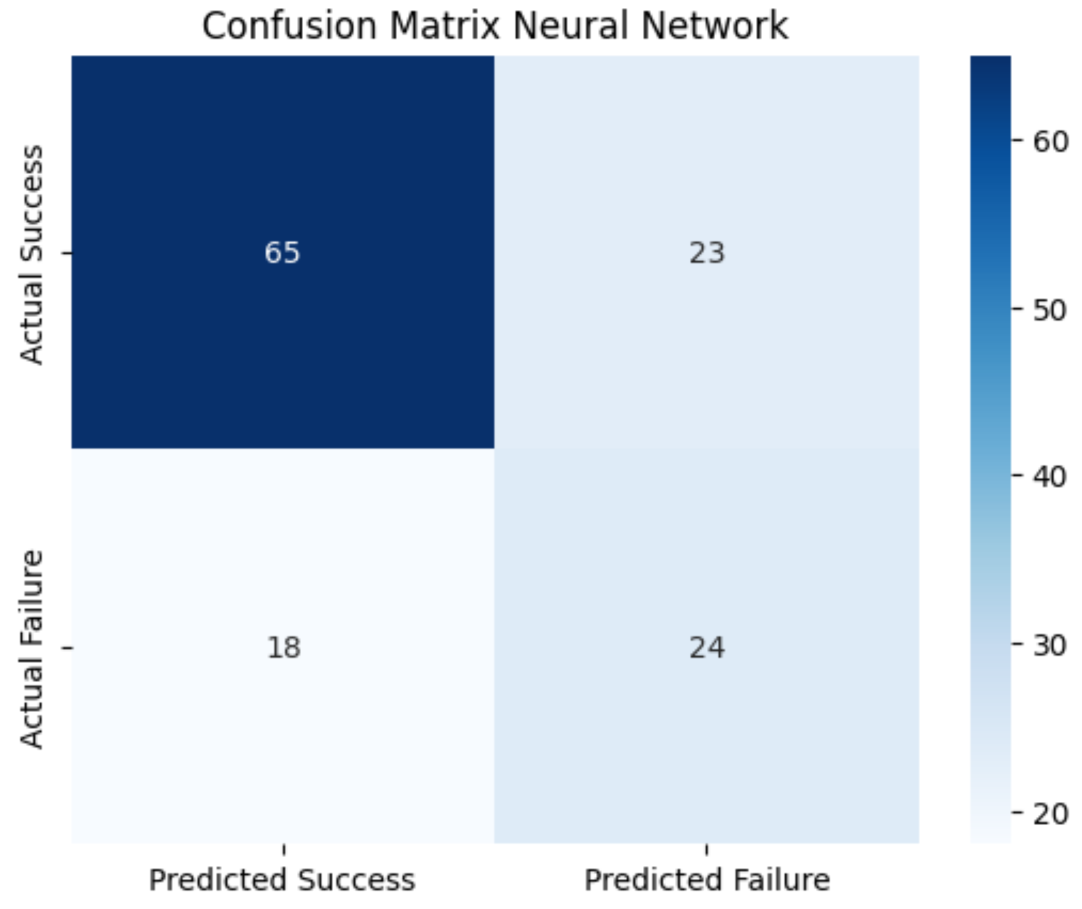


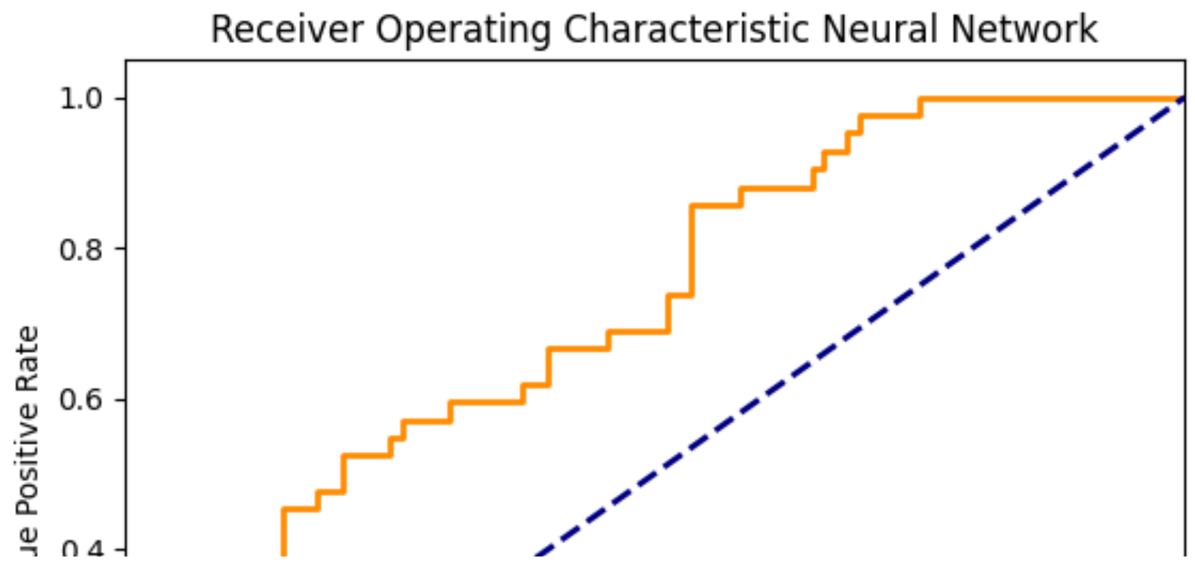


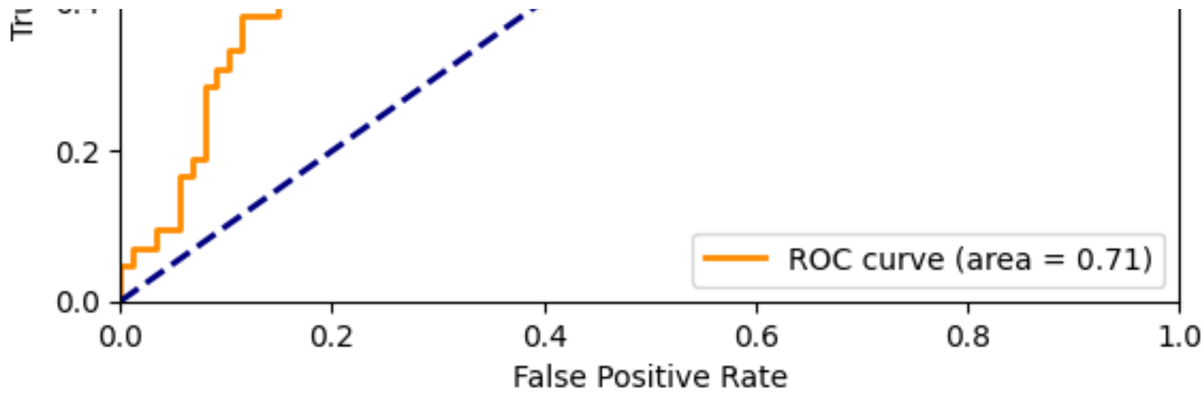


```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.0340909
TPR = [0.0, 0.023809523809523808, 0.047619047619047616, 0.0476190476190476l
AUC = 0.7094155844155844
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7022, Sensitivity: 0.0000, Specificity: 1.0000, F1:
Threshold: 0.10, Accuracy: 0.3231, Sensitivity: 1.0000, Specificity: 0.0000
Threshold: 0.15, Accuracy: 0.3462, Sensitivity: 1.0000, Specificity: 0.0341
Threshold: 0.20, Accuracy: 0.3846, Sensitivity: 1.0000, Specificity: 0.0909
Threshold: 0.25, Accuracy: 0.5308, Sensitivity: 0.8810, Specificity: 0.3636
Threshold: 0.30, Accuracy: 0.5846, Sensitivity: 0.6667, Specificity: 0.5455
Threshold: 0.35, Accuracy: 0.6846, Sensitivity: 0.5714, Specificity: 0.7386
Threshold: 0.40, Accuracy: 0.7077, Sensitivity: 0.4048, Specificity: 0.8523
Threshold: 0.45, Accuracy: 0.7154, Sensitivity: 0.3333, Specificity: 0.8977
Threshold: 0.50, Accuracy: 0.6846, Sensitivity: 0.1667, Specificity: 0.9318
Threshold: 0.55, Accuracy: 0.6923, Sensitivity: 0.0714, Specificity: 0.9886
```

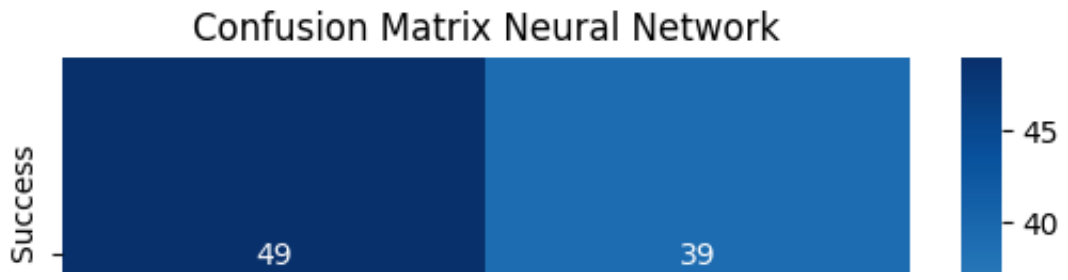
```
Threshold: 0.60, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.65, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.70, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.75, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.80, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.85, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.90, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.95, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 1.00, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step

Metrics for chosen threshold 0.35:
Accuracy: 0.6846, Sensitivity: 0.5714, Specificity: 0.7386, F1: 0.5393, ROC
```

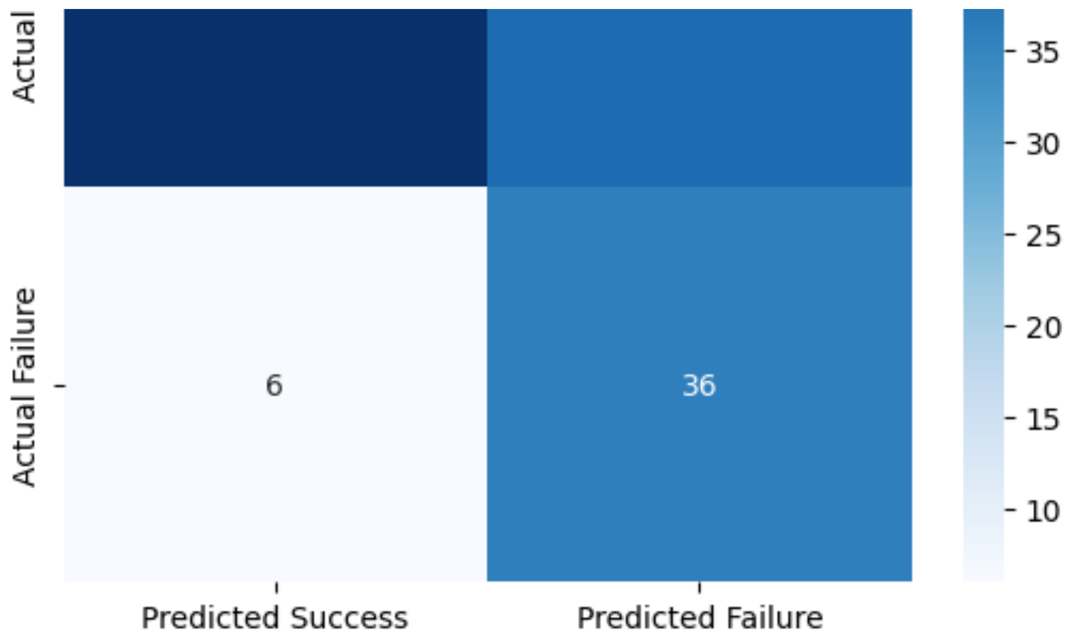


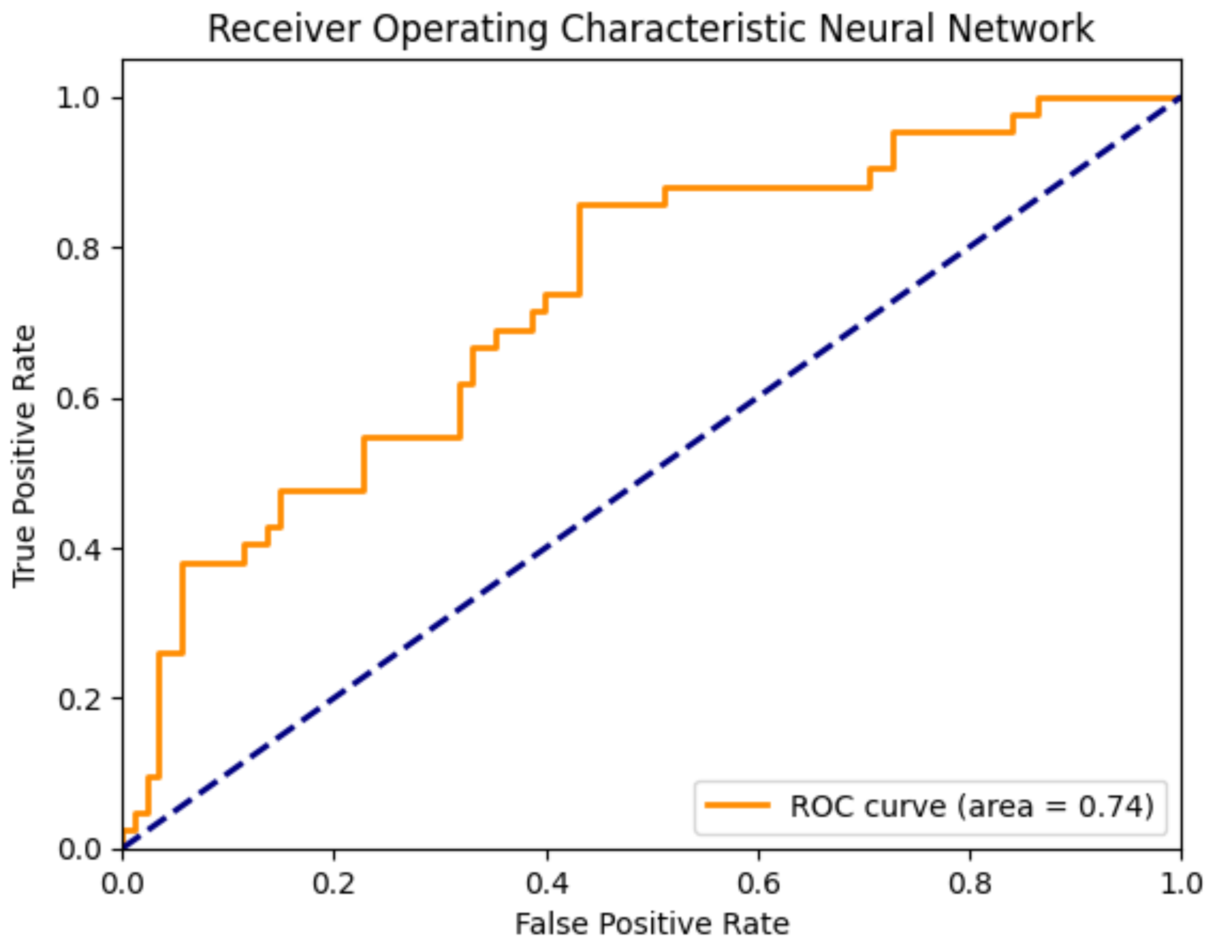




```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.022727272727
TPR = [0.0, 0.023809523809523808, 0.023809523809523808, 0.04761904761904761
AUC = 0.7359307359307359
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7022, Sensitivity: 0.0000, Specificity: 1.0000, F1:
Threshold: 0.10, Accuracy: 0.3231, Sensitivity: 1.0000, Specificity: 0.0000
Threshold: 0.15, Accuracy: 0.3385, Sensitivity: 1.0000, Specificity: 0.0227
Threshold: 0.20, Accuracy: 0.3538, Sensitivity: 1.0000, Specificity: 0.0455
Threshold: 0.25, Accuracy: 0.4154, Sensitivity: 0.9762, Specificity: 0.1477
Threshold: 0.30, Accuracy: 0.4846, Sensitivity: 0.8810, Specificity: 0.2955
Threshold: 0.35, Accuracy: 0.6538, Sensitivity: 0.8571, Specificity: 0.5568
Threshold: 0.40, Accuracy: 0.6615, Sensitivity: 0.5476, Specificity: 0.7159
Threshold: 0.45, Accuracy: 0.7308, Sensitivity: 0.3810, Specificity: 0.8977
Threshold: 0.50, Accuracy: 0.7385, Sensitivity: 0.2619, Specificity: 0.9659
Threshold: 0.55, Accuracy: 0.7077, Sensitivity: 0.1667, Specificity: 0.9659
Threshold: 0.60, Accuracy: 0.6769, Sensitivity: 0.0476, Specificity: 0.9773
Threshold: 0.65, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.70, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.75, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.80, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.85, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.90, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.95, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 1.00, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step

Metrics for chosen threshold 0.35:
Accuracy: 0.6538, Sensitivity: 0.8571, Specificity: 0.5568, F1: 0.6154, ROC
```

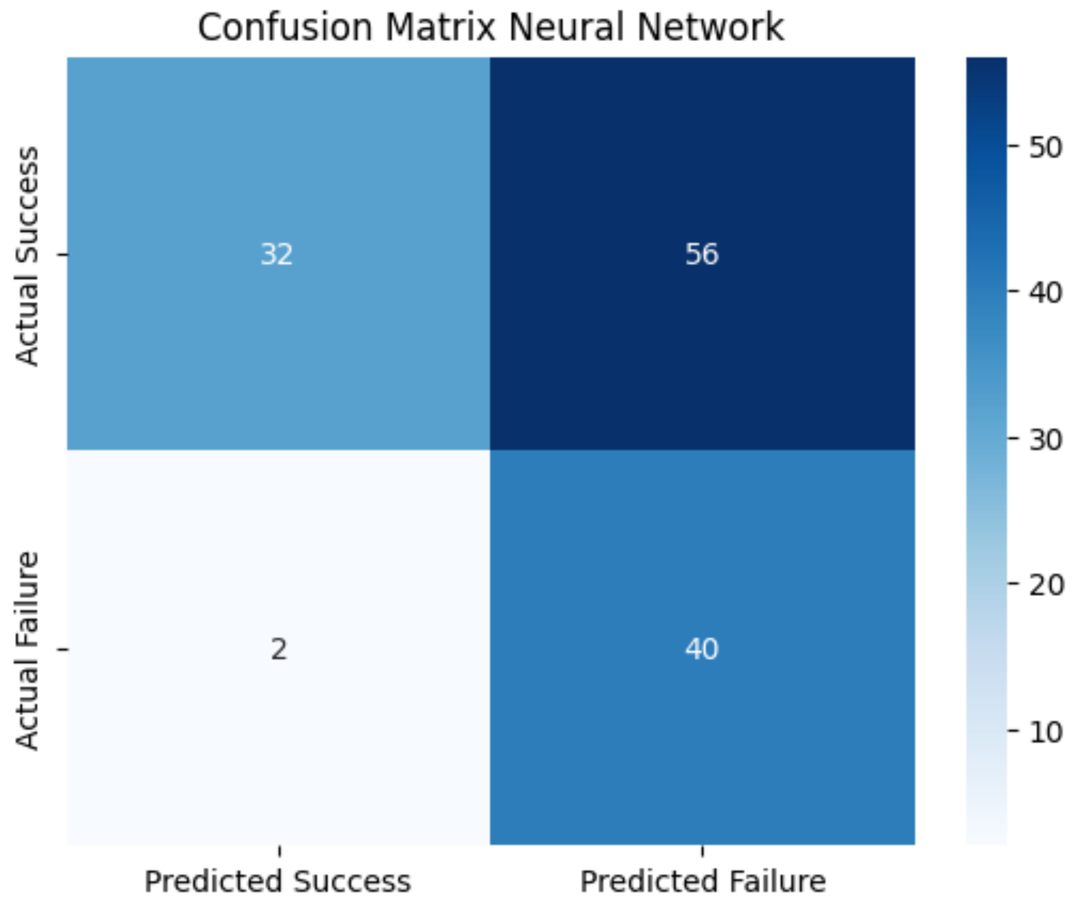


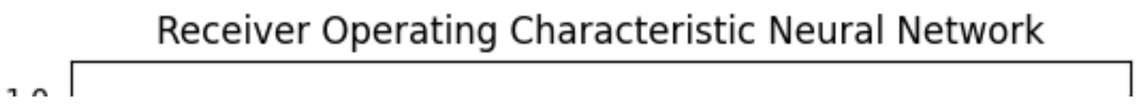


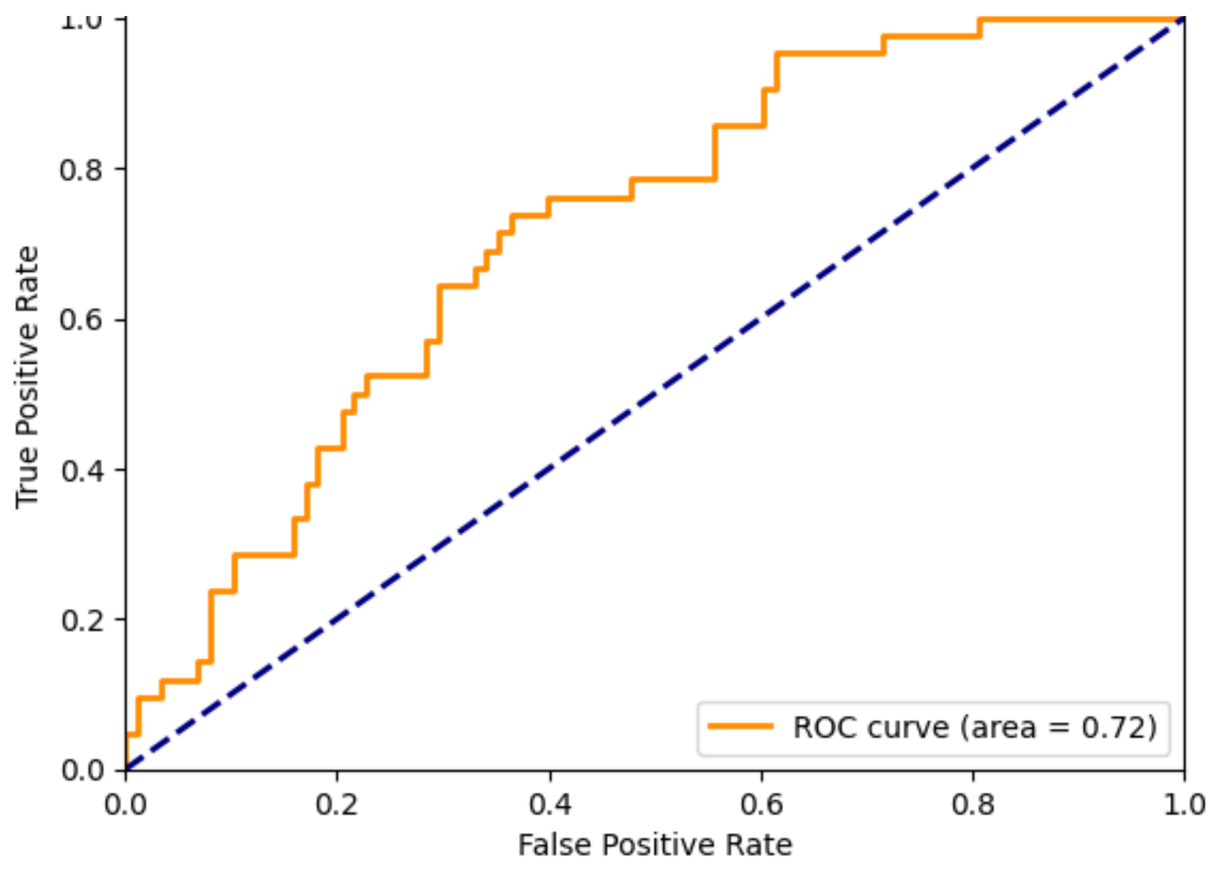


```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.0340909
TPR = [0.0, 0.023809523809523808, 0.047619047619047616, 0.04761904761904761
AUC = 0.7175324675324675
```

```
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7022, Sensitivity: 0.0000, Specificity: 1.0000, F1:
Threshold: 0.10, Accuracy: 0.3231, Sensitivity: 1.0000, Specificity: 0.0000
Threshold: 0.15, Accuracy: 0.3231, Sensitivity: 1.0000, Specificity: 0.0000
Threshold: 0.20, Accuracy: 0.3308, Sensitivity: 1.0000, Specificity: 0.0114
Threshold: 0.25, Accuracy: 0.3769, Sensitivity: 1.0000, Specificity: 0.0795
Threshold: 0.30, Accuracy: 0.4846, Sensitivity: 0.9762, Specificity: 0.2500
Threshold: 0.35, Accuracy: 0.5538, Sensitivity: 0.9524, Specificity: 0.3636
Threshold: 0.40, Accuracy: 0.5692, Sensitivity: 0.7857, Specificity: 0.4659
Threshold: 0.45, Accuracy: 0.6692, Sensitivity: 0.7381, Specificity: 0.6364
Threshold: 0.50, Accuracy: 0.6923, Sensitivity: 0.5238, Specificity: 0.7727
Threshold: 0.55, Accuracy: 0.6615, Sensitivity: 0.2857, Specificity: 0.8409
Threshold: 0.60, Accuracy: 0.6846, Sensitivity: 0.1905, Specificity: 0.9205
Threshold: 0.65, Accuracy: 0.7000, Sensitivity: 0.0952, Specificity: 0.9886
Threshold: 0.70, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.75, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.80, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.85, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.90, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.95, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 1.00, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step

Metrics for chosen threshold 0.35:
Accuracy: 0.5538, Sensitivity: 0.9524, Specificity: 0.3636, F1: 0.5797, ROC
```

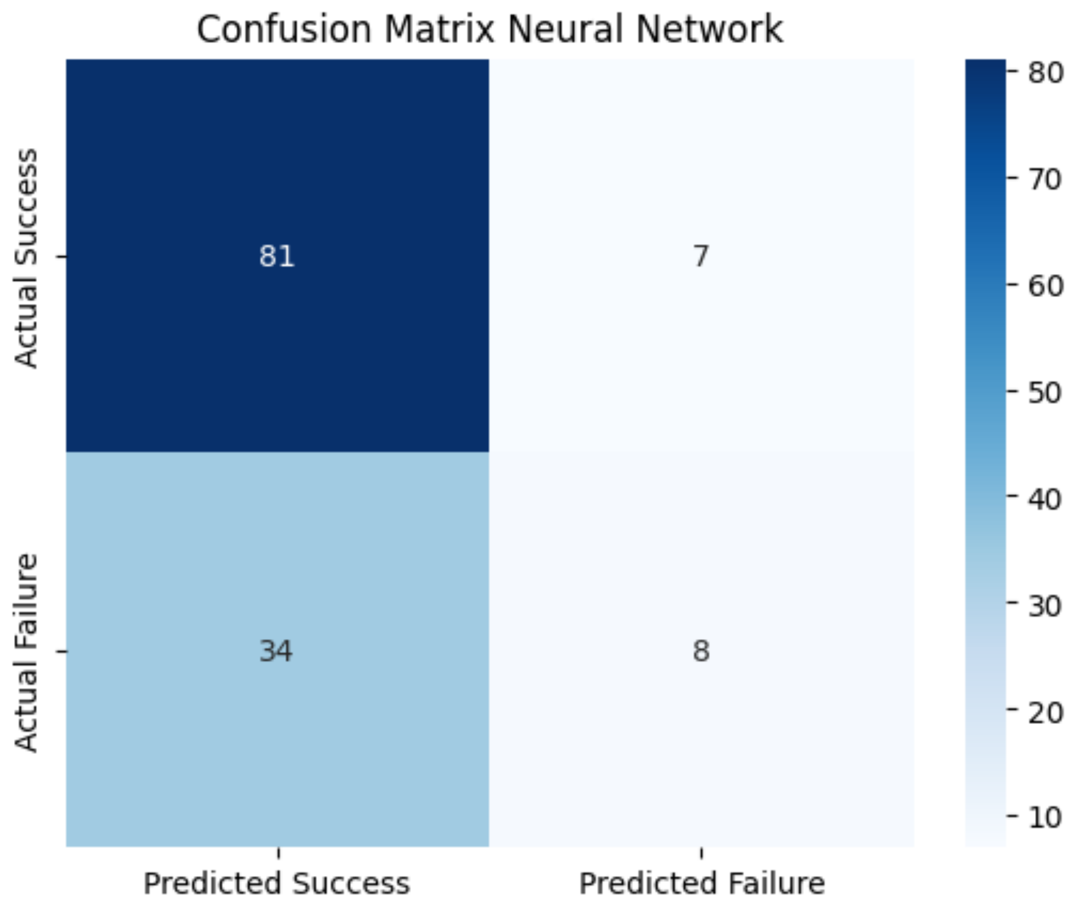


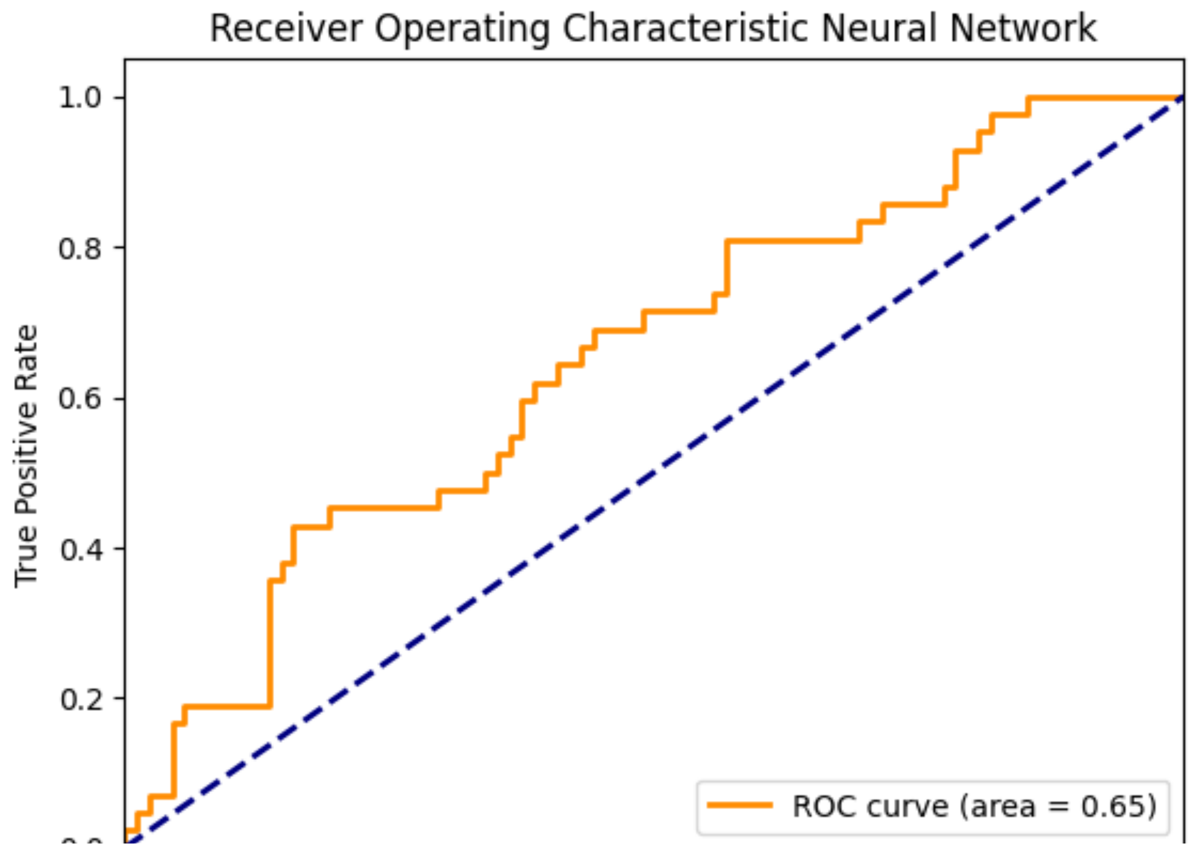


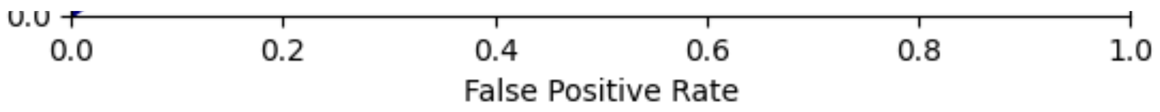


```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.022727272727
TPR = [0.0, 0.023809523809523808, 0.023809523809523808, 0.04761904761904761
AUC = 0.6515151515151515
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7022, Sensitivity: 0.0000, Specificity: 1.0000, F1:
Threshold: 0.10, Accuracy: 0.3846, Sensitivity: 1.0000, Specificity: 0.0909
Threshold: 0.15, Accuracy: 0.4385, Sensitivity: 0.9048, Specificity: 0.2159
Threshold: 0.20, Accuracy: 0.6000, Sensitivity: 0.6905, Specificity: 0.5568
Threshold: 0.25, Accuracy: 0.6538, Sensitivity: 0.4524, Specificity: 0.7500
Threshold: 0.30, Accuracy: 0.6462, Sensitivity: 0.1905, Specificity: 0.8636
Threshold: 0.35, Accuracy: 0.6846, Sensitivity: 0.1905, Specificity: 0.9205
Threshold: 0.40, Accuracy: 0.6846, Sensitivity: 0.1190, Specificity: 0.9545
Threshold: 0.45, Accuracy: 0.6846, Sensitivity: 0.0238, Specificity: 1.0000
Threshold: 0.50, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.55, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.60, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.65, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.70, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.75, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.80, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.85, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.90, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.95, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
```

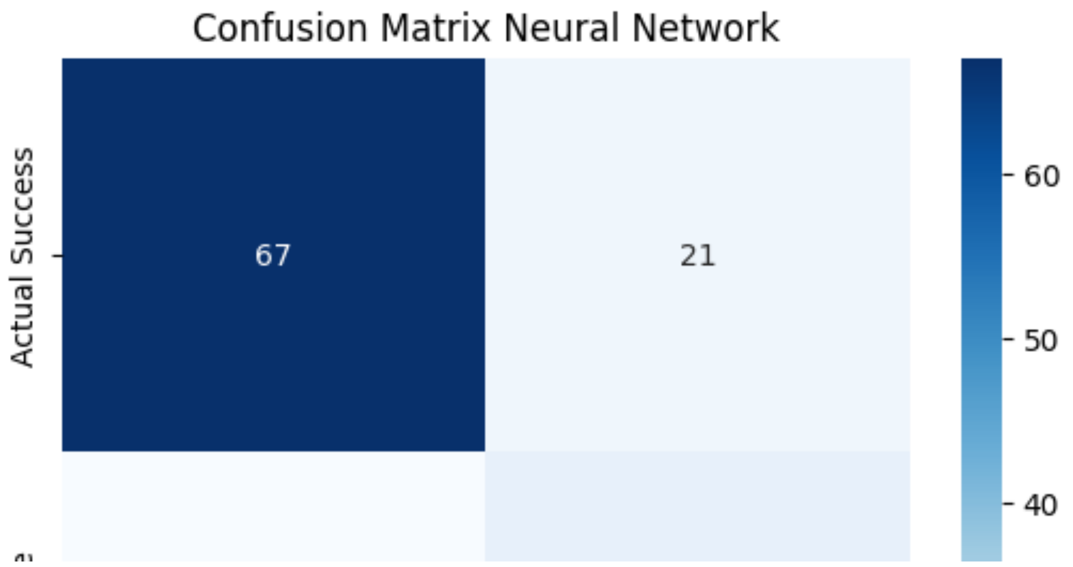
```
Threshold: 1.00, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step

Metrics for chosen threshold 0.35:
Accuracy: 0.6846, Sensitivity: 0.1905, Specificity: 0.9205, F1: 0.2807, ROC
```

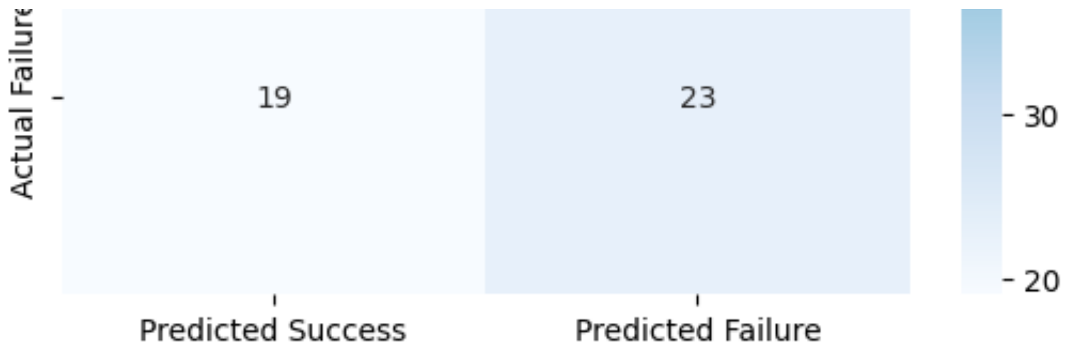


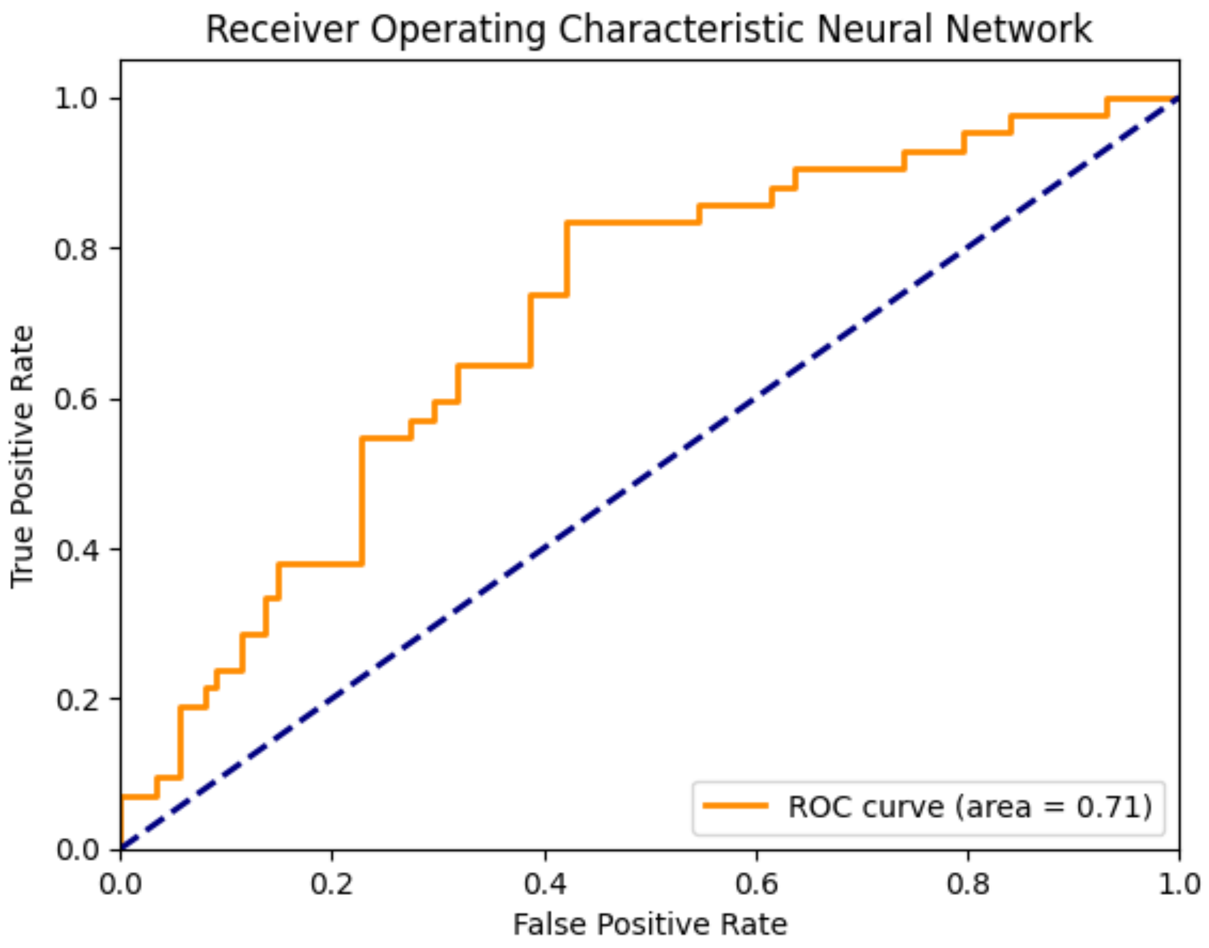




```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.03409090909090909, 0.03409090909090909, 0.056818181
TPR = [0.0, 0.023809523809523808, 0.07142857142857142, 0.07142857142857142,
AUC = 0.7058982683982684
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7022, Sensitivity: 0.0000, Specificity: 1.0000, F1:
Threshold: 0.10, Accuracy: 0.3231, Sensitivity: 1.0000, Specificity: 0.0000
Threshold: 0.15, Accuracy: 0.3385, Sensitivity: 1.0000, Specificity: 0.0227
Threshold: 0.20, Accuracy: 0.3846, Sensitivity: 0.9762, Specificity: 0.1023
Threshold: 0.25, Accuracy: 0.5462, Sensitivity: 0.8810, Specificity: 0.3864
Threshold: 0.30, Accuracy: 0.6462, Sensitivity: 0.7381, Specificity: 0.6023
Threshold: 0.35, Accuracy: 0.6923, Sensitivity: 0.5476, Specificity: 0.7614
Threshold: 0.40, Accuracy: 0.6923, Sensitivity: 0.3810, Specificity: 0.8409
Threshold: 0.45, Accuracy: 0.6846, Sensitivity: 0.2381, Specificity: 0.8977
Threshold: 0.50, Accuracy: 0.7000, Sensitivity: 0.1905, Specificity: 0.9432
Threshold: 0.55, Accuracy: 0.6769, Sensitivity: 0.0714, Specificity: 0.9659
Threshold: 0.60, Accuracy: 0.6923, Sensitivity: 0.0476, Specificity: 1.0000
Threshold: 0.65, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.70, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.75, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.80, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.85, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.90, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.95, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 1.00, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step

Metrics for chosen threshold 0.35:
Accuracy: 0.6923, Sensitivity: 0.5476, Specificity: 0.7614, F1: 0.5349, ROC
```

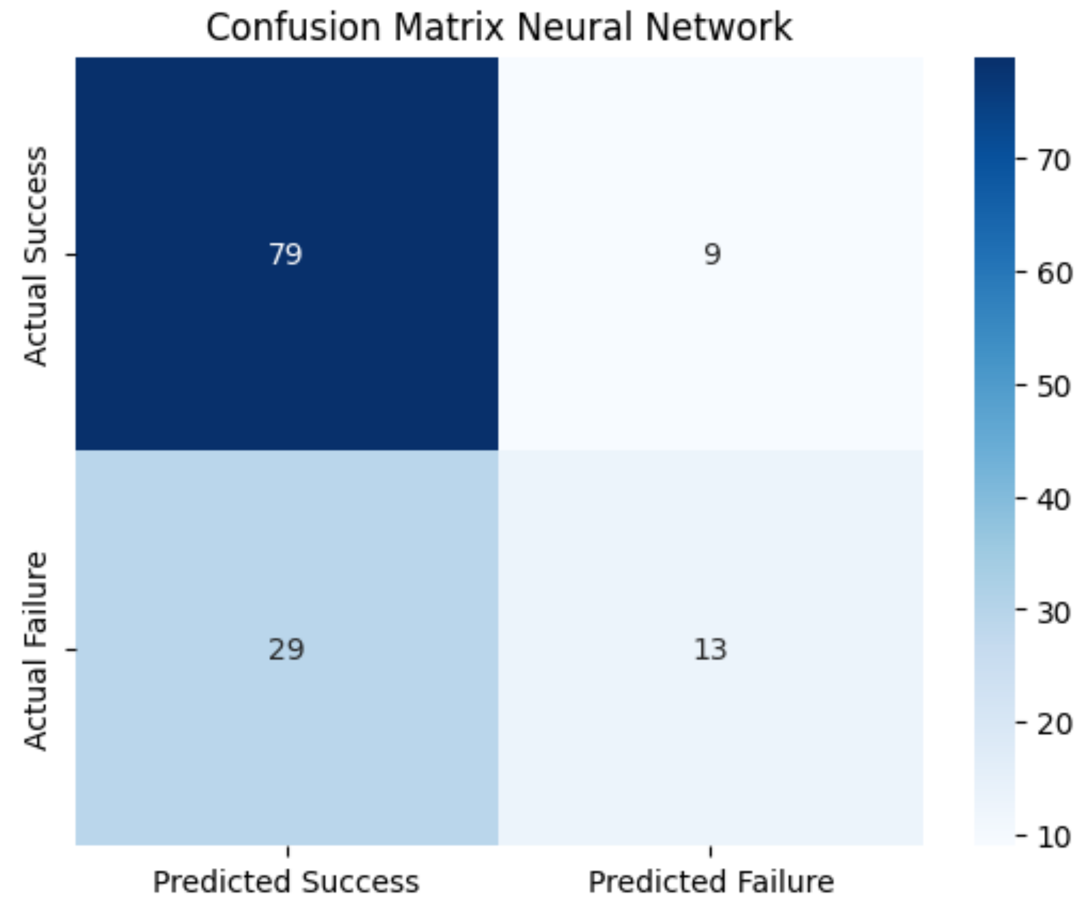


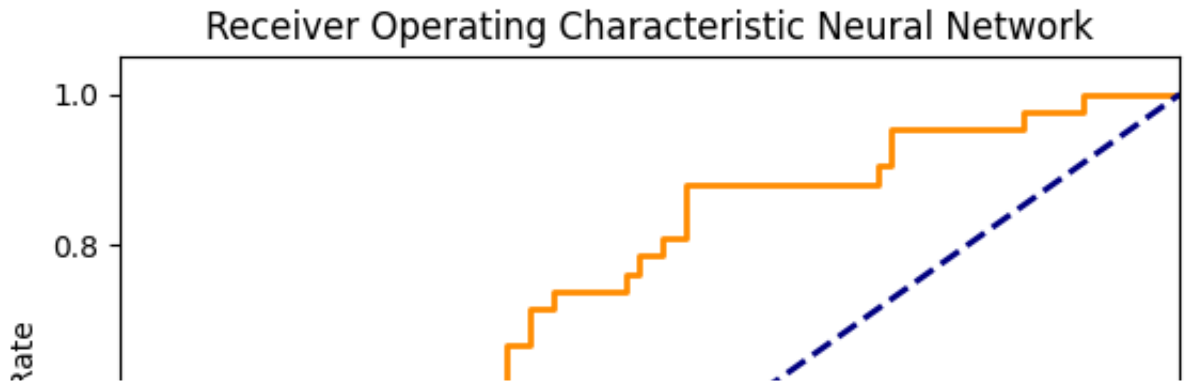


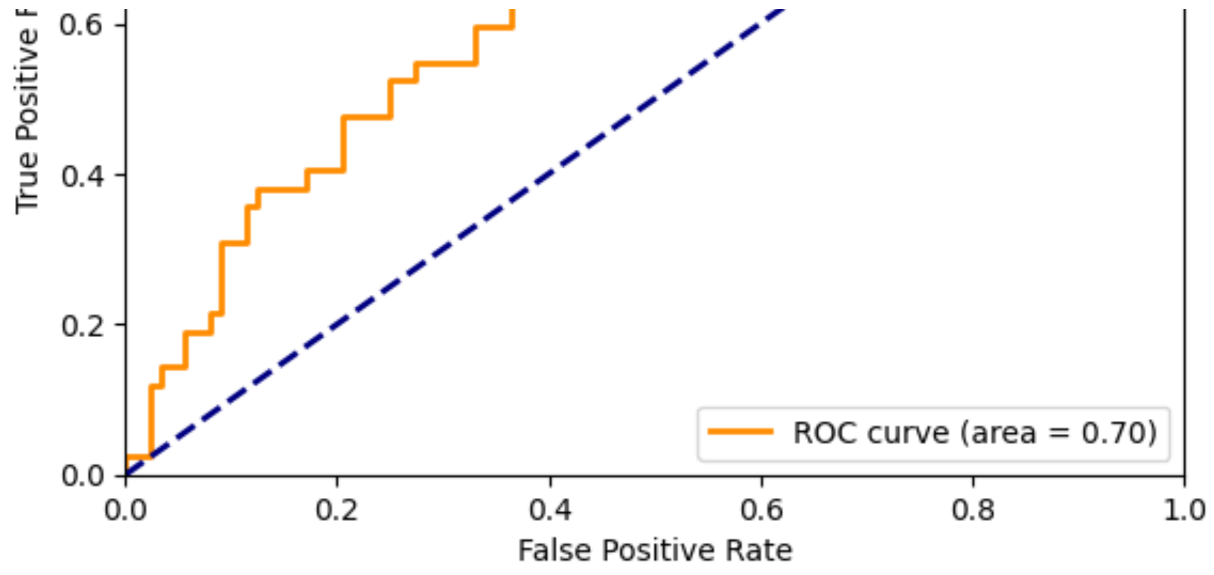


```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.034090909090
TPR = [0.0, 0.023809523809523808, 0.023809523809523808, 0.11904761904761904
AUC = 0.701569264069264
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7022, Sensitivity: 0.0000, Specificity: 1.0000, F1:
Threshold: 0.10, Accuracy: 0.3615, Sensitivity: 1.0000, Specificity: 0.0568
Threshold: 0.15, Accuracy: 0.4769, Sensitivity: 0.8810, Specificity: 0.2841
Threshold: 0.20, Accuracy: 0.6077, Sensitivity: 0.7381, Specificity: 0.5455
Threshold: 0.25, Accuracy: 0.6538, Sensitivity: 0.5476, Specificity: 0.7045
Threshold: 0.30, Accuracy: 0.6846, Sensitivity: 0.4048, Specificity: 0.8182
```

```
Threshold: 0.35, Accuracy: 0.7077, Sensitivity: 0.3095, Specificity: 0.8977
Threshold: 0.40, Accuracy: 0.6923, Sensitivity: 0.2143, Specificity: 0.9205
Threshold: 0.45, Accuracy: 0.6923, Sensitivity: 0.1190, Specificity: 0.9659
Threshold: 0.50, Accuracy: 0.6923, Sensitivity: 0.0952, Specificity: 0.9773
Threshold: 0.55, Accuracy: 0.6769, Sensitivity: 0.0238, Specificity: 0.9886
Threshold: 0.60, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.65, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.70, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.75, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.80, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.85, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.90, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.95, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 1.00, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step

Metrics for chosen threshold 0.35:
Accuracy: 0.7077, Sensitivity: 0.3095, Specificity: 0.8977, F1: 0.4063, ROC
```

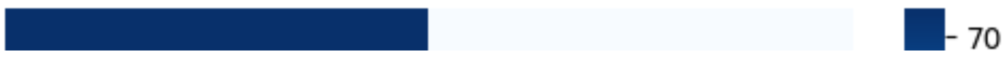


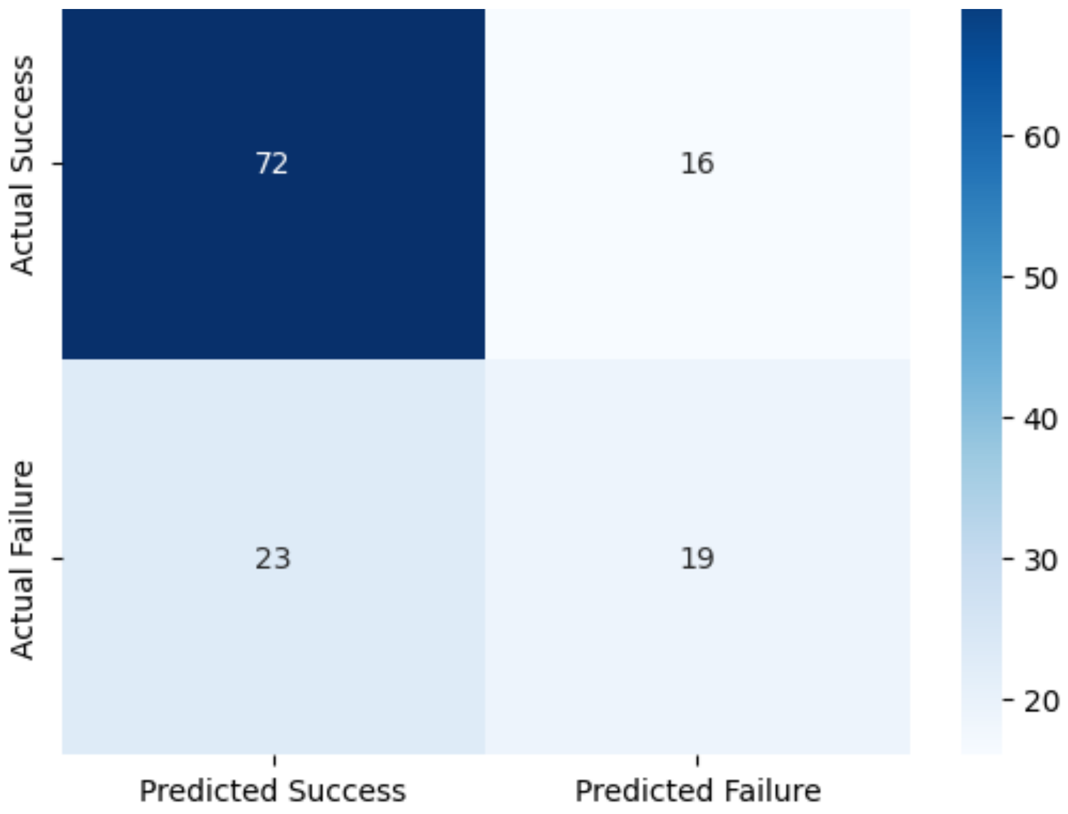


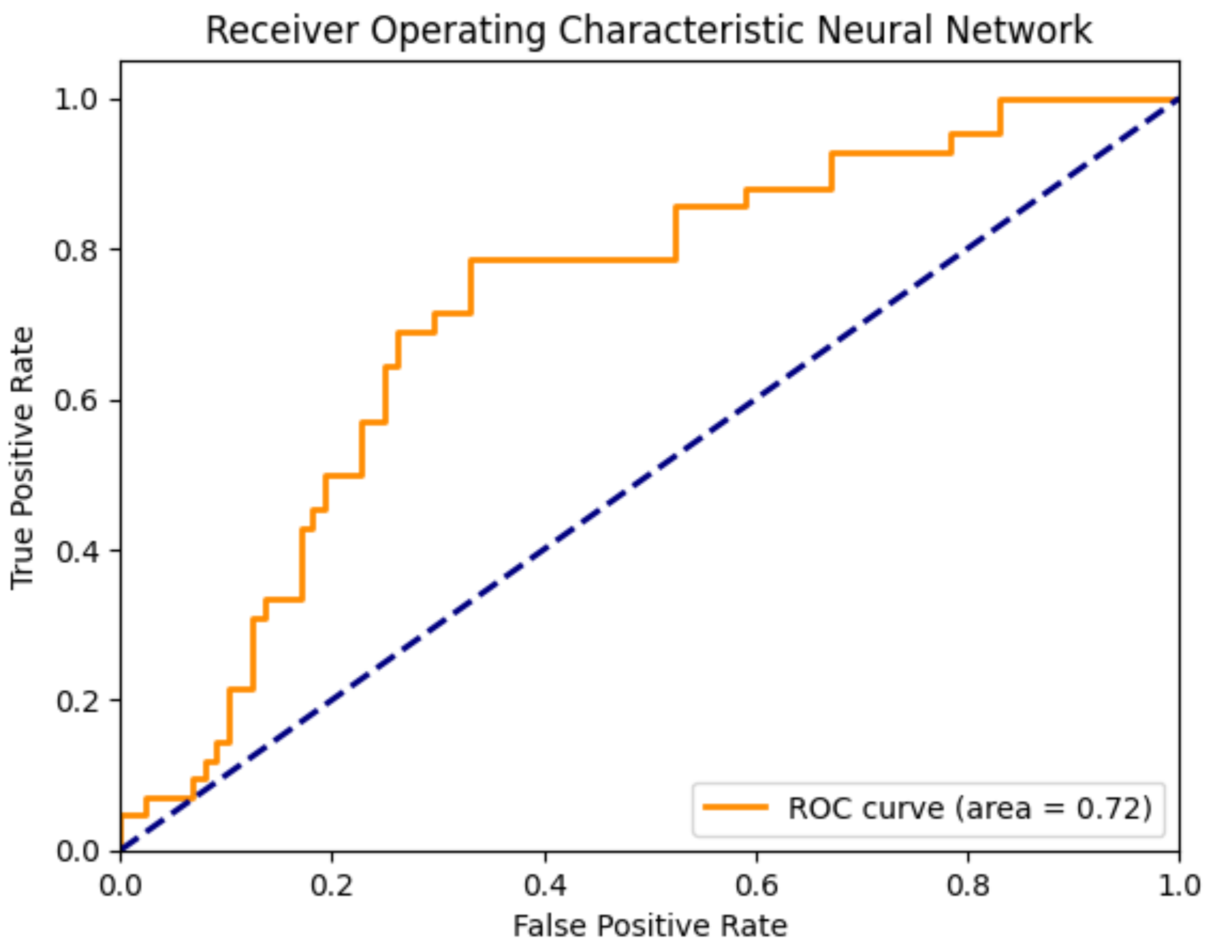


```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.022727272727272728, 0.022727272727272728, 0.0681818
TPR = [0.0, 0.023809523809523808, 0.047619047619047616, 0.04761904761904761
AUC = 0.722943722943723
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7022, Sensitivity: 0.0000, Specificity: 1.0000, F1:
Threshold: 0.10, Accuracy: 0.3231, Sensitivity: 1.0000, Specificity: 0.0000
Threshold: 0.15, Accuracy: 0.3385, Sensitivity: 1.0000, Specificity: 0.0227
Threshold: 0.20, Accuracy: 0.4231, Sensitivity: 0.9524, Specificity: 0.1705
Threshold: 0.25, Accuracy: 0.5231, Sensitivity: 0.8810, Specificity: 0.3523
Threshold: 0.30, Accuracy: 0.7000, Sensitivity: 0.7619, Specificity: 0.6705
Threshold: 0.35, Accuracy: 0.7000, Sensitivity: 0.4524, Specificity: 0.8182
Threshold: 0.40, Accuracy: 0.6769, Sensitivity: 0.2619, Specificity: 0.8750
Threshold: 0.45, Accuracy: 0.6692, Sensitivity: 0.0714, Specificity: 0.9545
Threshold: 0.50, Accuracy: 0.6923, Sensitivity: 0.0476, Specificity: 1.0000
Threshold: 0.55, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.60, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.65, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.70, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.75, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.80, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.85, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.90, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.95, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 1.00, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step

Metrics for chosen threshold 0.35:
Accuracy: 0.7000, Sensitivity: 0.4524, Specificity: 0.8182, F1: 0.4935, ROC
```

Confusion Matrix Neural Network

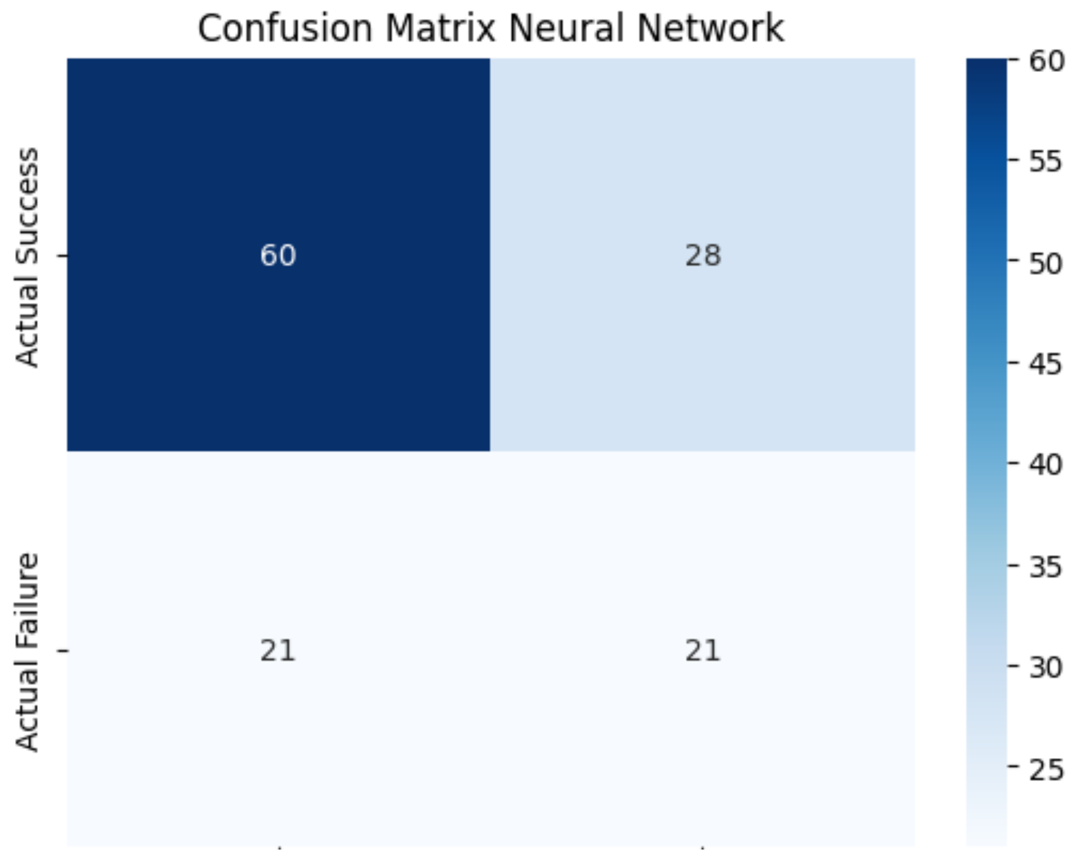


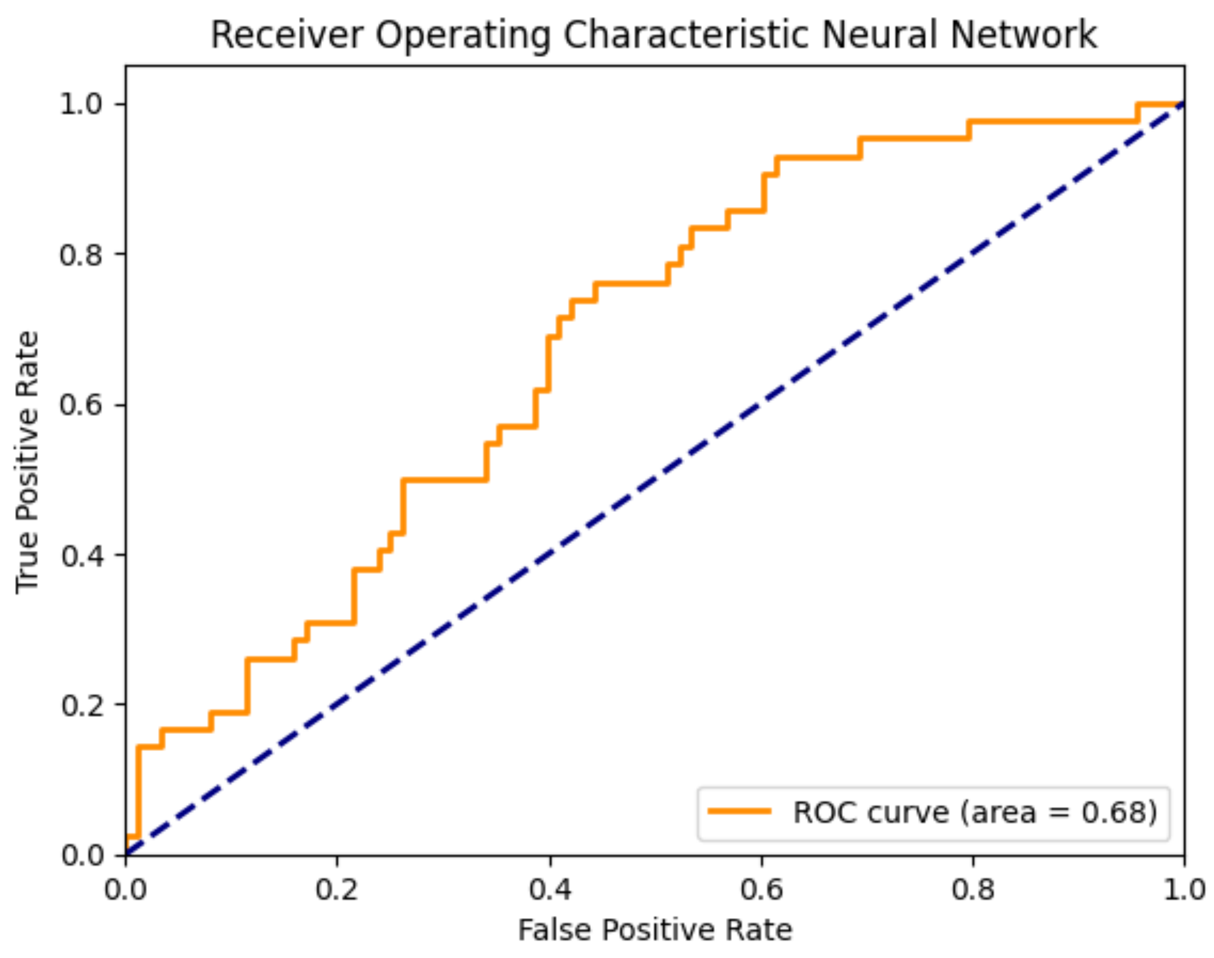




```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step
```

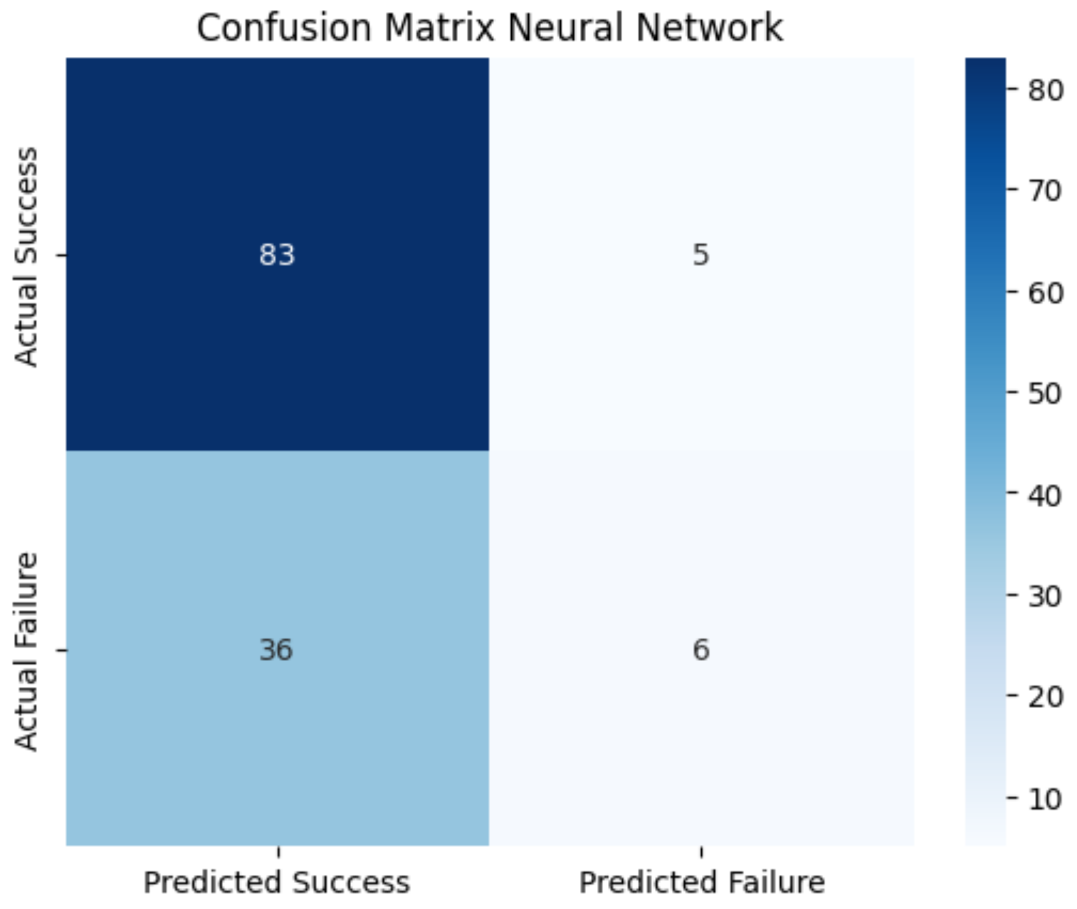
```
--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.011363636363636364, 0.011363636363636364, 0.011363636363
TPR = [0.0, 0.023809523809523808, 0.023809523809523808, 0.07142857142857142
AUC = 0.6801948051948052
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7022, Sensitivity: 0.0000, Specificity: 1.0000, F1:
Threshold: 0.10, Accuracy: 0.3231, Sensitivity: 1.0000, Specificity: 0.0000
Threshold: 0.15, Accuracy: 0.3538, Sensitivity: 1.0000, Specificity: 0.0455
Threshold: 0.20, Accuracy: 0.4077, Sensitivity: 0.9762, Specificity: 0.1364
Threshold: 0.25, Accuracy: 0.5231, Sensitivity: 0.9286, Specificity: 0.3295
Threshold: 0.30, Accuracy: 0.6077, Sensitivity: 0.7619, Specificity: 0.5341
Threshold: 0.35, Accuracy: 0.6231, Sensitivity: 0.5000, Specificity: 0.6818
Threshold: 0.40, Accuracy: 0.6308, Sensitivity: 0.3095, Specificity: 0.7841
Threshold: 0.45, Accuracy: 0.6769, Sensitivity: 0.1905, Specificity: 0.9091
Threshold: 0.50, Accuracy: 0.7077, Sensitivity: 0.1190, Specificity: 0.9886
Threshold: 0.55, Accuracy: 0.7000, Sensitivity: 0.0952, Specificity: 0.9886
Threshold: 0.60, Accuracy: 0.6769, Sensitivity: 0.0238, Specificity: 0.9886
Threshold: 0.65, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.70, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.75, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.80, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.85, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.90, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.95, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 1.00, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step

Metrics for chosen threshold 0.35:
Accuracy: 0.6231, Sensitivity: 0.5000, Specificity: 0.6818, F1: 0.4615, ROC
```

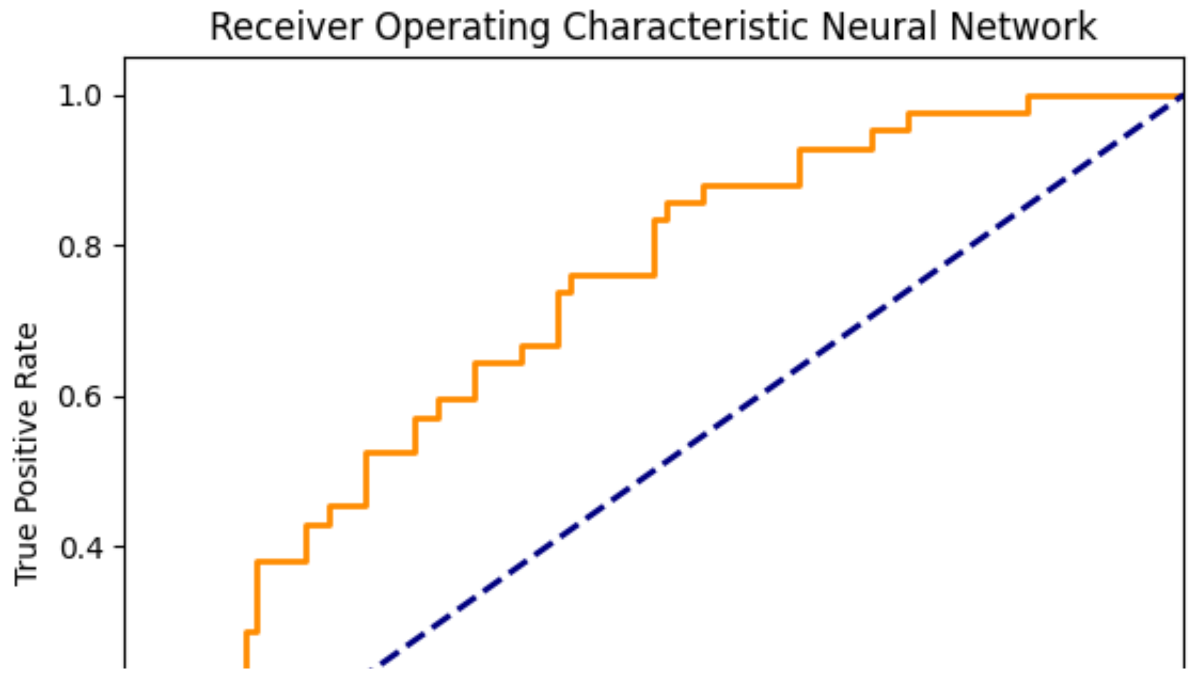


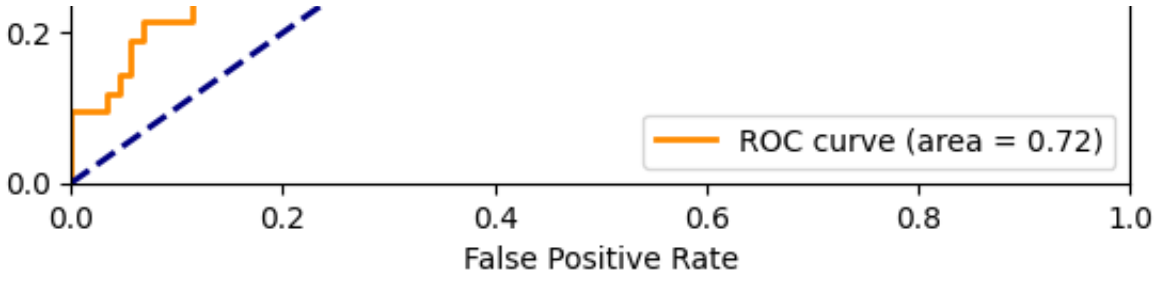


```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step

--- Dados ROC para copiar ---
FPR = [0.0, 0.0, 0.0, 0.03409090909090909, 0.03409090909090909, 0.045454545
TPR = [0.0, 0.023809523809523808, 0.09523809523809523, 0.09523809523809523,
AUC = 0.7150974025974025
--- Fim dos Dados ROC ---

Training - Accuracy: 0.7022, Sensitivity: 0.0000, Specificity: 1.0000, F1:
Threshold: 0.10, Accuracy: 0.3538, Sensitivity: 1.0000, Specificity: 0.0455
Threshold: 0.15, Accuracy: 0.5308, Sensitivity: 0.8810, Specificity: 0.3636
Threshold: 0.20, Accuracy: 0.6385, Sensitivity: 0.6429, Specificity: 0.6364
Threshold: 0.25, Accuracy: 0.6923, Sensitivity: 0.4286, Specificity: 0.8182
Threshold: 0.30, Accuracy: 0.6769, Sensitivity: 0.2143, Specificity: 0.8977
Threshold: 0.35, Accuracy: 0.6846, Sensitivity: 0.1429, Specificity: 0.9432
Threshold: 0.40, Accuracy: 0.7000, Sensitivity: 0.0952, Specificity: 0.9886
Threshold: 0.45, Accuracy: 0.6923, Sensitivity: 0.0476, Specificity: 1.0000
Threshold: 0.50, Accuracy: 0.6923, Sensitivity: 0.0476, Specificity: 1.0000
Threshold: 0.55, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.60, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.65, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.70, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.75, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
```

```
Threshold: 0.75, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.80, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.85, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.90, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 0.95, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
Threshold: 1.00, Accuracy: 0.6769, Sensitivity: 0.0000, Specificity: 1.0000
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step

Metrics for chosen threshold 0.35:
Accuracy: 0.6846, Sensitivity: 0.1429, Specificity: 0.9432, F1: 0.2264, ROC
```







```
Aggregated Test Set Metrics Across Seeds:
   accuracy  sensitivity  specificity        f1   roc_auc
0  0.715385     0.214286     0.954545  0.327273  0.720509
1  0.684615     0.571429     0.738636  0.539326  0.709416
2  0.653846     0.857143     0.556818  0.615385  0.735931
3  0.553846     0.952381     0.363636  0.579710  0.717532
4  0.684615     0.190476     0.920455  0.280702  0.651515
5  0.692308     0.547619     0.761364  0.534884  0.705898
6  0.707692     0.309524     0.897727  0.406250  0.701569
7  0.700000     0.452381     0.818182  0.493506  0.722944
8  0.623077     0.500000     0.681818  0.461538  0.680195
9  0.684615     0.142857     0.943182  0.226415  0.715097

Summary of Test Set Metrics (Mean, Standard Error, 95% Confidence Interval)
Accuracy: Mean = 0.6700, SE = 0.0154, 95% CI = [0.6351, 0.7049]
Sensitivity: Mean = 0.4738, SE = 0.0867, 95% CI = [0.2777, 0.6699]
Specificity: Mean = 0.7636, SE = 0.0600, 95% CI = [0.6278, 0.8994]
F1: Mean = 0.4465, SE = 0.0418, 95% CI = [0.3520, 0.5409]
Roc_auc: Mean = 0.7061, SE = 0.0077, 95% CI = [0.6887, 0.7234]
```