

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

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

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

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

# Load the data from an Excel file
data = pd.read_excel('2024_corrected_global_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 t
unique_n_part = data['IDpac'].unique()
train_n_part, test_n_part = train_test_split(unique_n_part, test_size=0.3, rand

# Filter the original dataset to create training data that includes only the 'I
train_data = data[data['IDpac'].isin(train_n_part)]
# Similarly, filter the original dataset to create testing data that includes c
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 targ
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 proc
X_test = test_data.drop(['Failure', 'IDrest', 'IDpac'], axis=1)
y_test = test_data['Failure'] # Isolate the target variable for the testing se

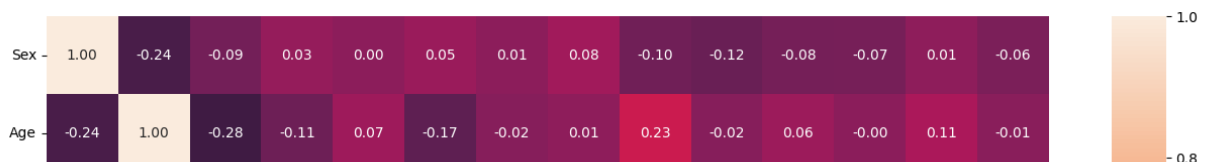
import seaborn as sns
import matplotlib.pyplot as plt

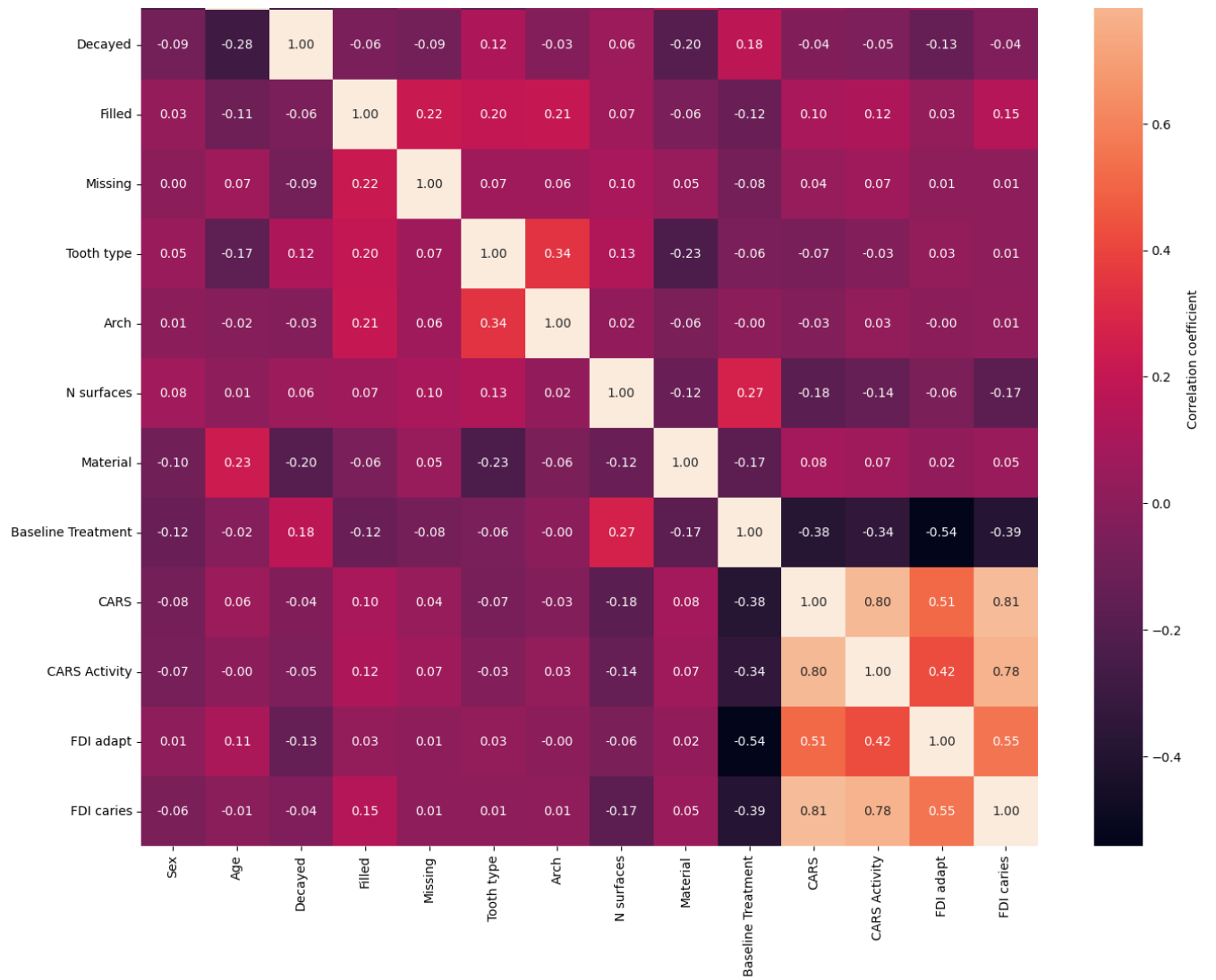
# Calculate the correlation matrix of the training data.
# The correlation matrix quantifies the linear relationships between the variabl
corr_matrix = X_train.corr()

# 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_kws

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





```

from sklearn.preprocessing import StandardScaler

# 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_test)

# 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))

import pandas as pd

# Define lists categorizing the types of variables in the dataset.
numeric_vars = ['Age', 'Decayed', 'Filled', 'Missing']
original_categorical_vars = ['Sex', 'Tooth type', 'Arch', 'Failure', 'CARS Activ
                             'Material', 'Baseline Treatment', 'CARS', 'FDI adap

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

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

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

    stats = {}
    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) * 100
            },
            "Test Set": {
                "Count": test_data[var].value_counts().to_dict(),
                "Percentage": (test_data[var].value_counts(normalize=True) * 100
            }
        }

    # Print Categorical Statistics

```

```

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

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

```

## ➡ N surfaces Statistics:

Training Set:

Count: {1: 180, 2: 130, 3: 61, 4: 50, 5: 25}

Percentage: {1: 40.35874439461883, 2: 29.14798206278027, 3: 13.677130044843

Test Set:

Count: {1: 83, 2: 36, 3: 31, 4: 24, 5: 17}

Percentage: {1: 43.455497382198956, 2: 18.848167539267017, 3: 16.2303664921

Material Statistics:

Training Set:

Count: {1: 263, 0: 169, 2: 14}

Percentage: {1: 58.96860986547085, 0: 37.89237668161435, 2: 3.1390134529147

Test Set:

Count: {1: 115, 0: 71, 2: 5}

Percentage: {1: 60.20942408376963, 0: 37.17277486910995, 2: 2.6178010471204

Baseline Treatment Statistics:

Training Set:

Count: {0: 261, 1: 147, 2: 38}

Percentage: {0: 58.52017937219731, 1: 32.95964125560538, 2: 8.5201793721973

Test Set:

Count: {0: 106, 1: 54, 2: 31}

Percentage: {0: 55.497382198952884, 1: 28.272251308900525, 2: 16.2303664921

CARS Statistics:

Training Set:

Count: {0: 350, 2: 53, 1: 43}

Percentage: {0: 78.47533632286996, 2: 11.883408071748878, 1: 9.641255605381

Test Set:

Count: {0: 148, 2: 29, 1: 14}

Percentage: {0: 77.4869109947644, 2: 15.18324607329843, 1: 7.32984293193717

FDI adapt Statistics:

Training Set:

Count: {0: 287, 1: 142, 2: 17}

Percentage: {0: 64.34977578475336, 1: 31.838565022421523, 2: 3.811659192825}

Test Set:

Count: {0: 128, 1: 56, 2: 7}

Percentage: {0: 67.01570680628272, 1: 29.31937172774869, 2: 3.6649214659685}

FDI caries Statistics:

Training Set:

Count: {0: 351, 1: 88, 2: 7}

Percentage: {0: 78.69955156950674, 1: 19.730941704035875, 2: 1.569506726457}

Test Set:

Count: {0: 147, 1: 40, 2: 4}

Percentage: {0: 76.96335078534031, 1: 20.94240837696335, 2: 2.0942408376963}

# Define custom metrics

def sensitivity(y\_true, y\_pred):

tn, fp, fn, tp = confusion\_matrix(y\_true, y\_pred).ravel()

return tp / (tp + fn)

def specificity(y\_true, y\_pred):

tn, fp, fn, tp = confusion\_matrix(y\_true, y\_pred).ravel()

return tn / (tn + fp)

```
!pip install catboost
```

```
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 catboost import CatBoostClassifier
from sklearn.model_selection import cross_validate, StratifiedKFold, GridSearchC
from sklearn.metrics import make_scorer, accuracy_score, roc_auc_score, f1_score
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, Learnin
from tensorflow.keras.regularizers import l2
from scipy import stats
```

🔄 Collecting catboost

```
Downloading catboost-1.2.7-cp311-cp311-manylinux2014_x86_64.whl.metadata
Requirement already satisfied: graphviz in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist
Requirement already satisfied: numpy<2.0, >=1.16.0 in /usr/local/lib/python3
Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.11/di
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: plotly in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: six in /usr/local/lib/python3.11/dist-packag
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: contourpy>=1.0.1 in /usr/local/lib/python3.1
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/di
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.1
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.11
Downloading catboost-1.2.7-cp311-cp311-manylinux2014_x86_64.whl (98.7 MB)
98.7/98.7 MB 6.0 MB/s eta 0:00:
Installing collected packages: catboost
Successfully installed catboost-1.2.7
```

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

```

print(f"Processing {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, scoring='roc_auc')

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

print(f"Best parameters for {name}: {best_params}") # Print the best parameters

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

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

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

# Imprimir os valores de FPR, TPR e AUC de forma fácil de copiar
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 training set metrics
print(f"Training - Accuracy: {train_acc}, Sensitivity: {train_sens}, Specificity: {train_spec}")

# 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

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'

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)

# --- Added for metrics 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, (CatBoostClassifier)):
        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 CatBoost')
    plt.show()

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

```

```

roc_auc = auc(tpr, 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 CatBoost')
plt.legend(loc="lower right")
plt.show()

def evaluate_catboost(X_train, y_train, X_test, y_test, cv, scoring, manual_thre
    print("Inside evaluate_catboost function")
    model = CatBoostClassifier(verbose=0, random_seed=seed)
    grid = {
        'depth': [6],
        'learning_rate': [0.01],
        'iterations': [300],
        'l2_leaf_reg': [5],
        'border_count': [128],
    }

    return evaluate_model(model, "CatBoost", grid, X_train, y_train, X_test, y_t

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 = 0.45
    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_clf
            X_train, y_train, X_test, y_test, cv, scoring, manual_threshold, thr
        )

        # Use calibrated_clf for prediction probabilities
        v_probs = calibrated_clf.predict_proba(X_test)[0: -1]

```

```

y_probs = calibrated_catboost_probs(y_test, 1)
y_pred_manual = (y_probs >= manual_threshold).astype(int)

# Plotting functions assuming they are imported or defined elsewhere
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 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)

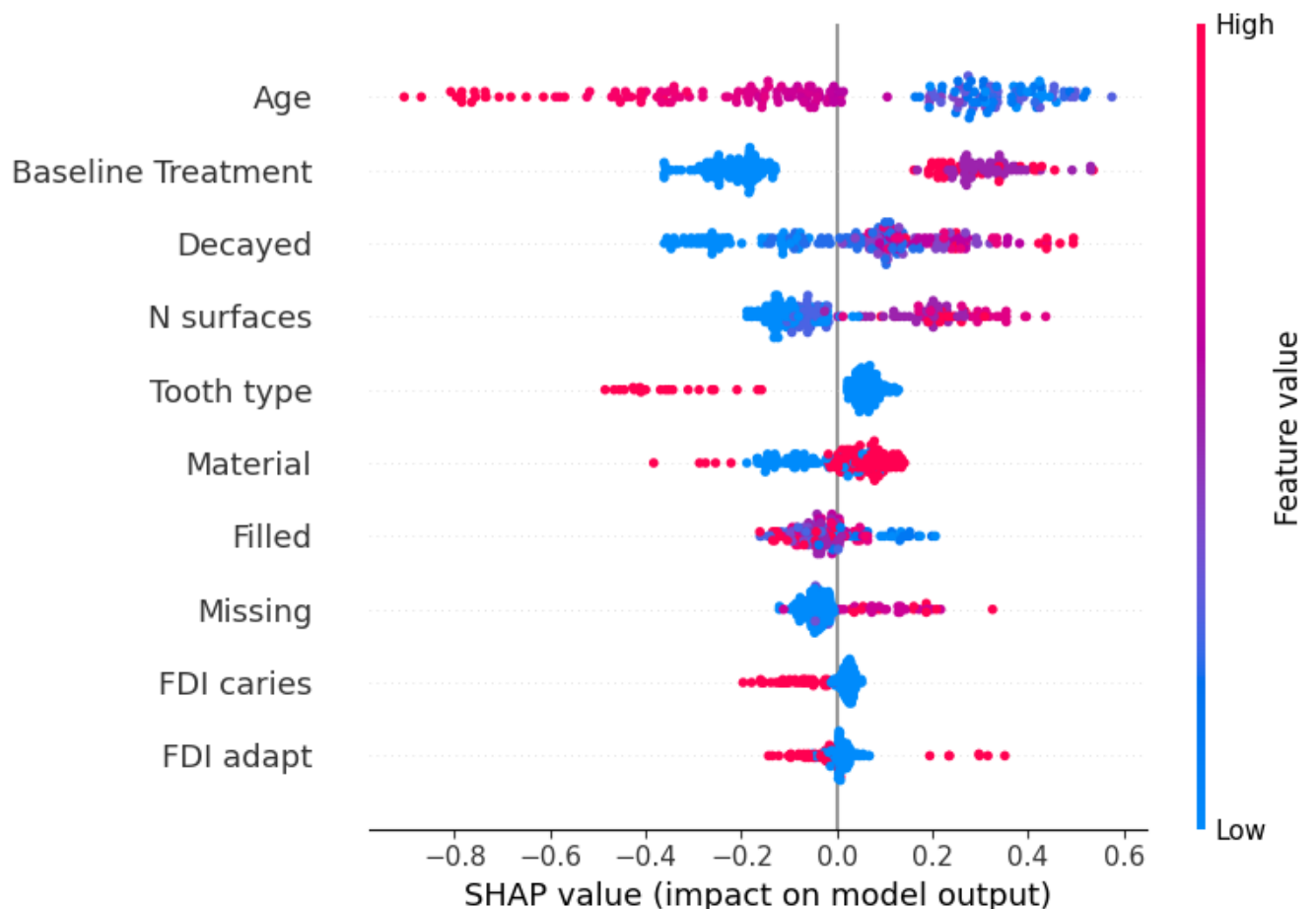
```

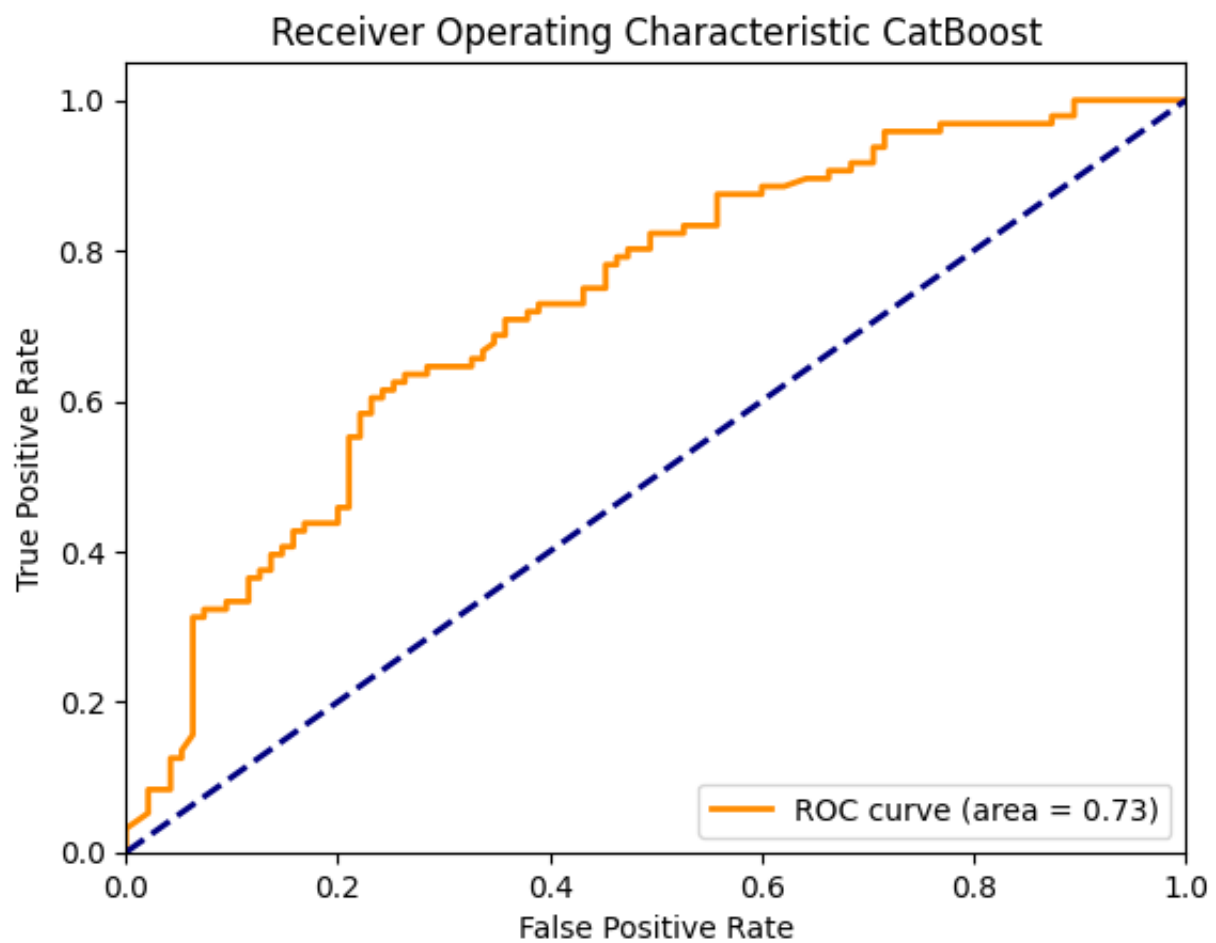
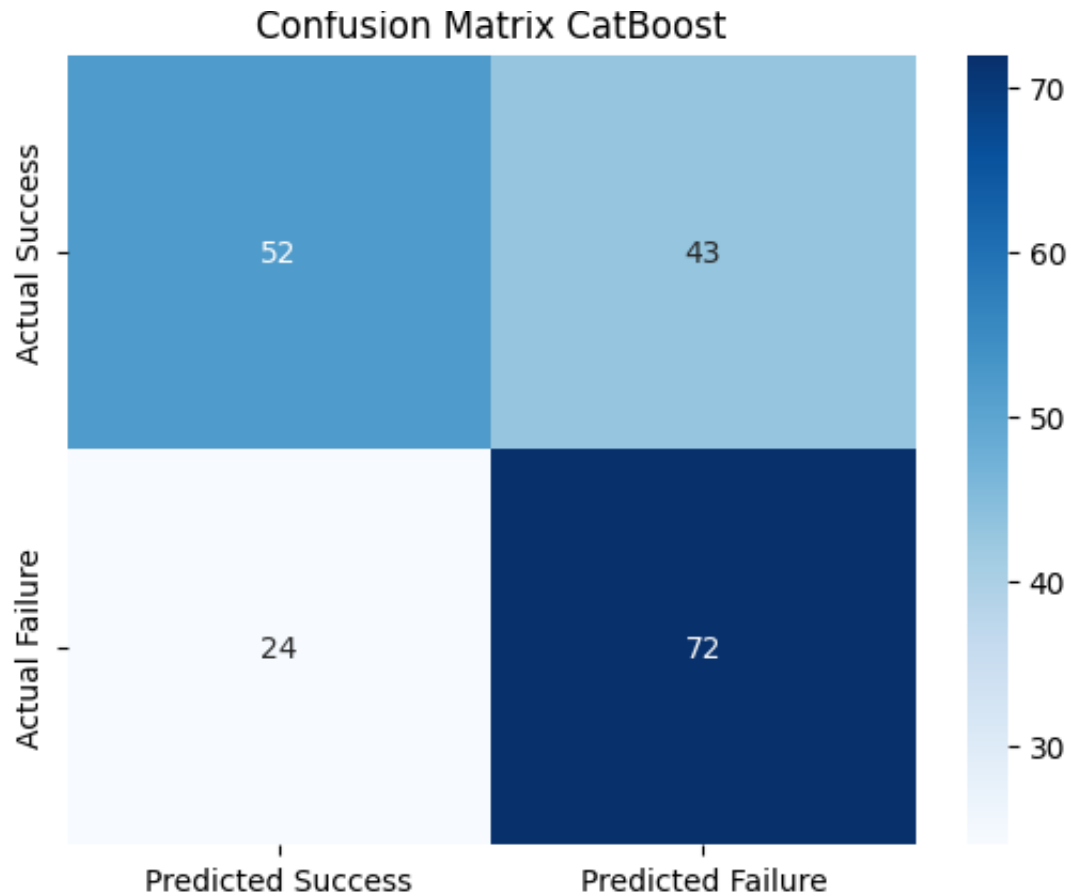
➡ Running evaluation with seed 40  
 Inside evaluate\_catboost function  
 Evaluating CatBoost...  
 Best parameters for CatBoost: {'border\_count': 128, 'depth': 6, 'iterations'  
 --- Dados ROC para copiar ---  
 FPR = 10.0 0.0 0.0 0.021052631578947368 0.021052631578947368 0.0421052

```

Training - Accuracy: 0.8340807174887892, Sensitivity: 0.8558139534883721, S
Metrics for manual threshold 0.45:
Accuracy: 0.6492146596858639, Sensitivity: 0.75, Specificity: 0.54736842105
Threshold: 0.10, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1
Threshold: 0.15, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1
Threshold: 0.20, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1
Threshold: 0.25, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1
Threshold: 0.30, Metrics: {'Accuracy': 0.5497382198952879, 'Sensitivity': 1
Threshold: 0.35, Metrics: {'Accuracy': 0.5863874345549738, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.6230366492146597, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.6492146596858639, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.680628272251309, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.6282722513089005, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.5968586387434555, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.518324607329843, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
SHAP Summary for CatBoost

```





Running evaluation with seed 41  
Inside evaluate\_catboost function

Evaluating CatBoost...

Best parameters for CatBoost: {'border\_count': 128, 'depth': 6, 'iterations

--- Dados ROC para copiar ---

FPR = [0.0, 0.0, 0.0, 0.010526315789473684, 0.010526315789473684, 0.0210526

TPR = [0.0, 0.010416666666666666, 0.041666666666666664, 0.05208333333333333

AUC = 0.7308662280701755

--- Fim dos Dados ROC ---

Training - Accuracy: 0.8295964125560538, Sensitivity: 0.8465116279069768, S  
Metrics for manual threshold 0.45:

Accuracy: 0.6596858638743456, Sensitivity: 0.78125, Specificity: 0.53684210

Threshold: 0.10, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.15, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.20, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.25, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.30, Metrics: {'Accuracy': 0.5497382198952879, 'Sensitivity': 1

Threshold: 0.35, Metrics: {'Accuracy': 0.6020942408376964, 'Sensitivity': 0

Threshold: 0.40, Metrics: {'Accuracy': 0.6387434554973822, 'Sensitivity': 0

Threshold: 0.45, Metrics: {'Accuracy': 0.6596858638743456, 'Sensitivity': 0

Threshold: 0.50, Metrics: {'Accuracy': 0.6858638743455497, 'Sensitivity': 0

Threshold: 0.55, Metrics: {'Accuracy': 0.6230366492146597, 'Sensitivity': 0

Threshold: 0.60, Metrics: {'Accuracy': 0.6178010471204188, 'Sensitivity': 0

Threshold: 0.65, Metrics: {'Accuracy': 0.5340314136125655, 'Sensitivity': 0

Threshold: 0.70, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 0

Threshold: 0.75, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.80, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

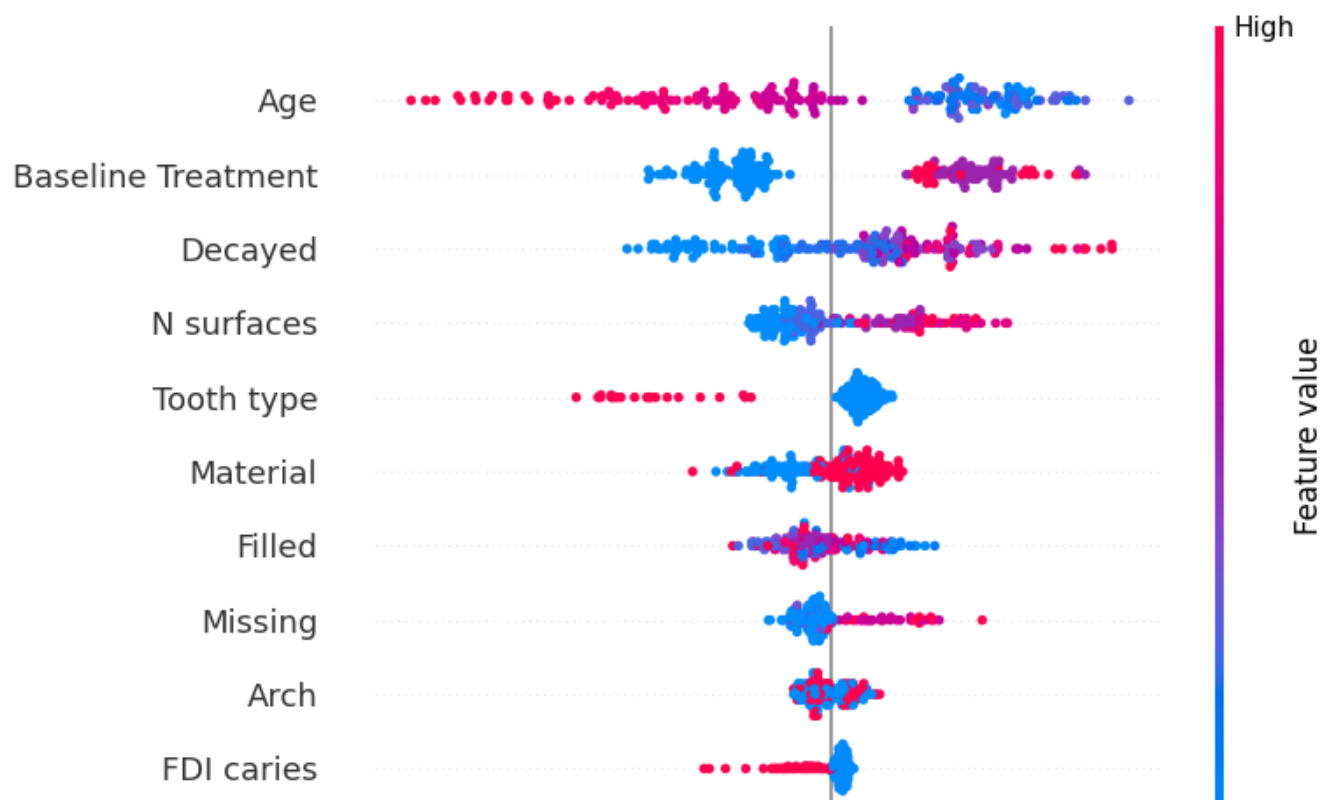
Threshold: 0.85, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

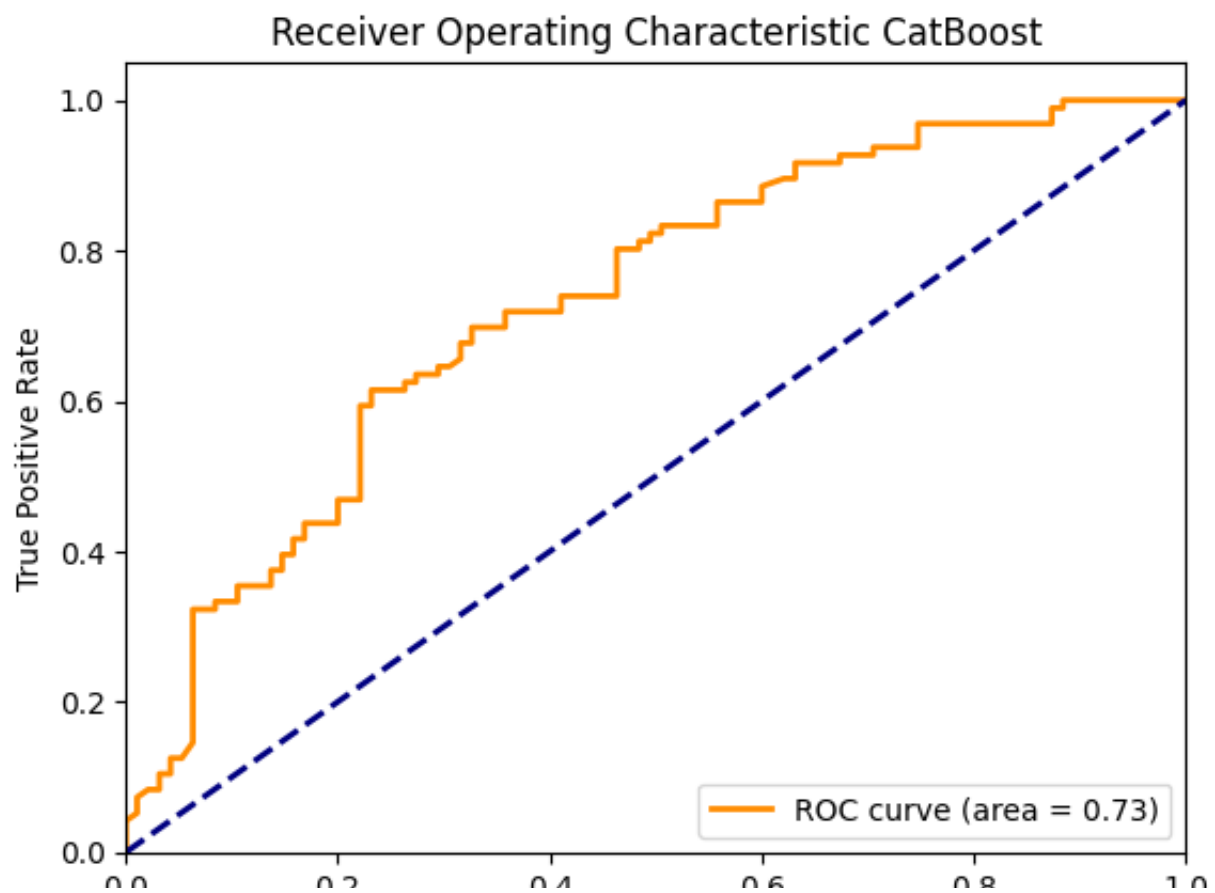
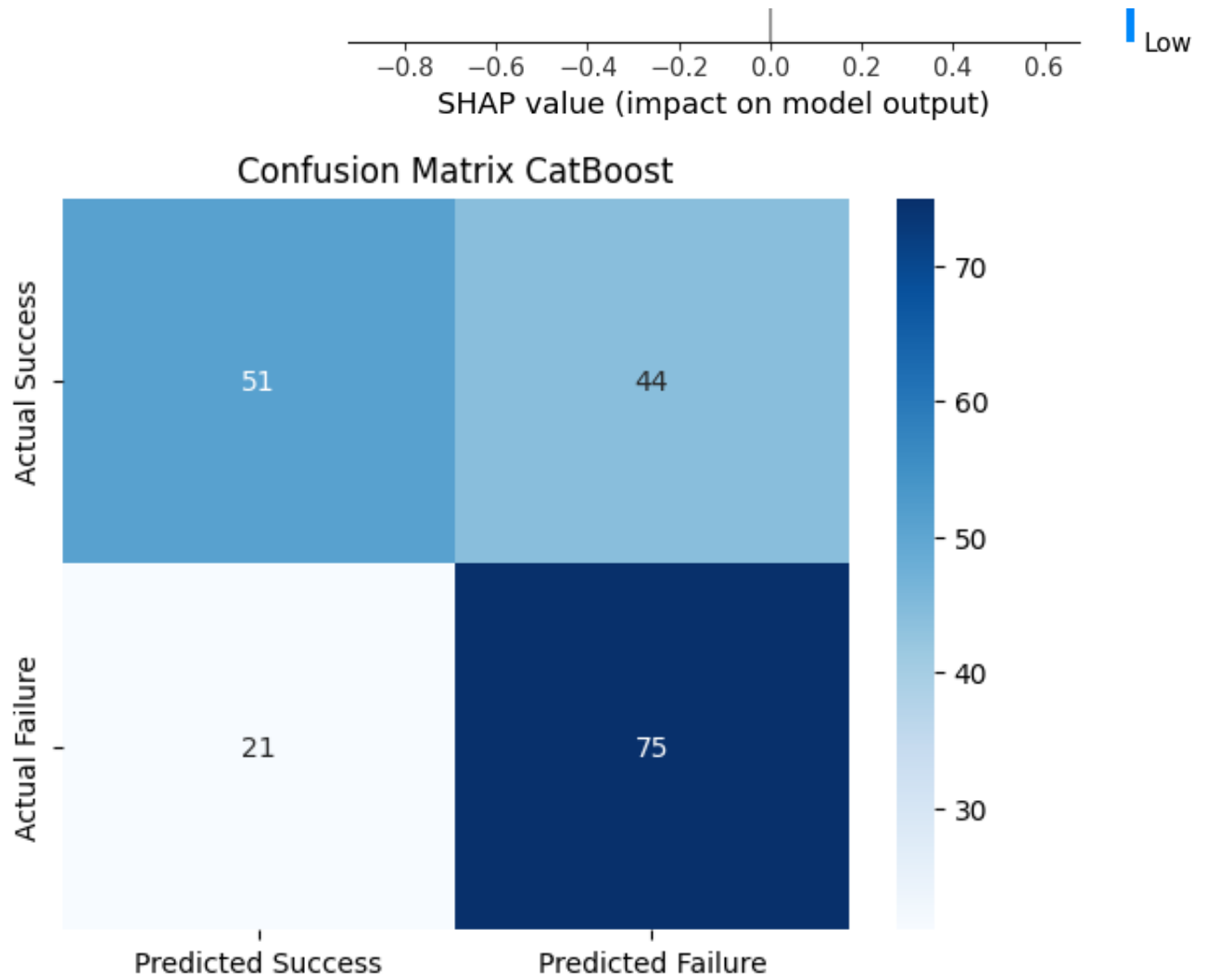
Threshold: 0.90, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.95, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 1.00, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

SHAP Summary for CatBoost





0.0 0.2 0.4 0.6 0.8 1.0  
False Positive Rate

Running evaluation with seed 42

Inside evaluate\_catboost function

Evaluating CatBoost...

Best parameters for CatBoost: {'border\_count': 128, 'depth': 6, 'iterations

--- Dados ROC para copiar ---

FPR = [0.0, 0.0, 0.0, 0.010526315789473684, 0.010526315789473684, 0.0210526

TPR = [0.0, 0.010416666666666666, 0.041666666666666664, 0.05208333333333333

AUC = 0.7257127192982457

--- Fim dos Dados ROC ---

Training - Accuracy: 0.827354260089686, Sensitivity: 0.8465116279069768, Sp  
Metrics for manual threshold 0.45:

Accuracy: 0.6596858638743456, Sensitivity: 0.7604166666666666, Specificity:

Threshold: 0.10, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.15, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.20, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.25, Metrics: {'Accuracy': 0.5078534031413613, 'Sensitivity': 1

Threshold: 0.30, Metrics: {'Accuracy': 0.5497382198952879, 'Sensitivity': 1

Threshold: 0.35, Metrics: {'Accuracy': 0.6020942408376964, 'Sensitivity': 0

Threshold: 0.40, Metrics: {'Accuracy': 0.6492146596858639, 'Sensitivity': 0

Threshold: 0.45, Metrics: {'Accuracy': 0.6596858638743456, 'Sensitivity': 0

Threshold: 0.50, Metrics: {'Accuracy': 0.675392670157068, 'Sensitivity': 0.

Threshold: 0.55, Metrics: {'Accuracy': 0.6335078534031413, 'Sensitivity': 0

Threshold: 0.60, Metrics: {'Accuracy': 0.6073298429319371, 'Sensitivity': 0

Threshold: 0.65, Metrics: {'Accuracy': 0.5235602094240838, 'Sensitivity': 0

Threshold: 0.70, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 0

Threshold: 0.75, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.80, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.85, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.90, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

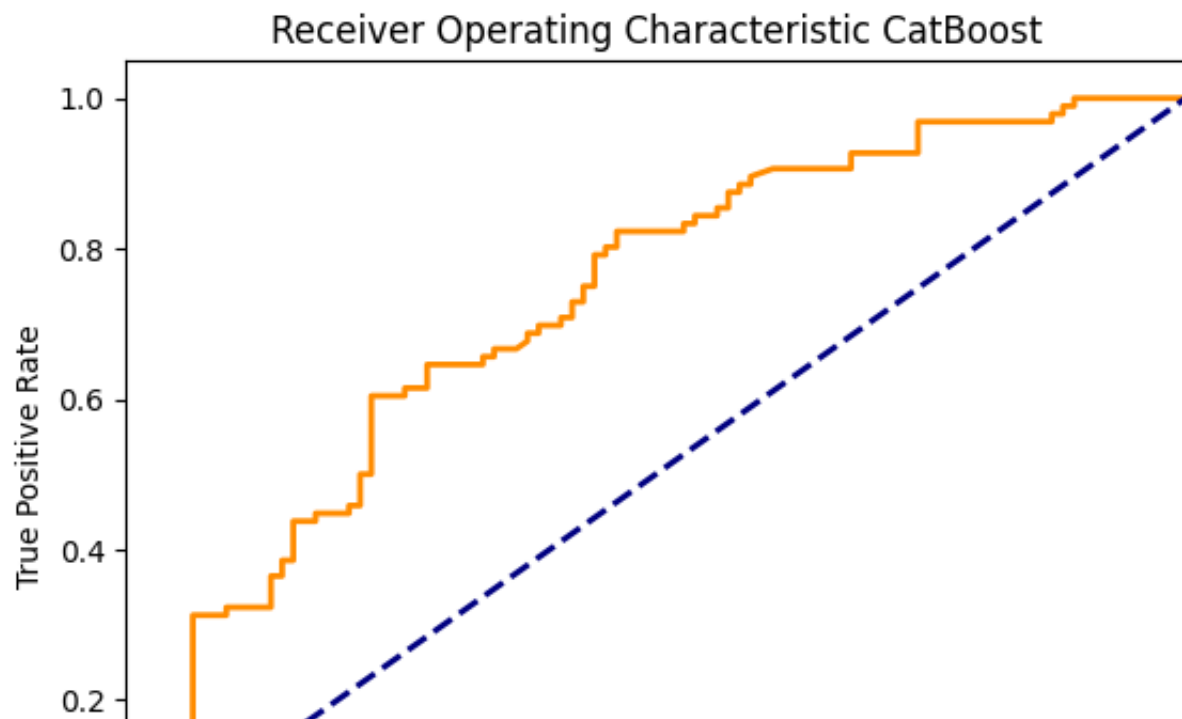
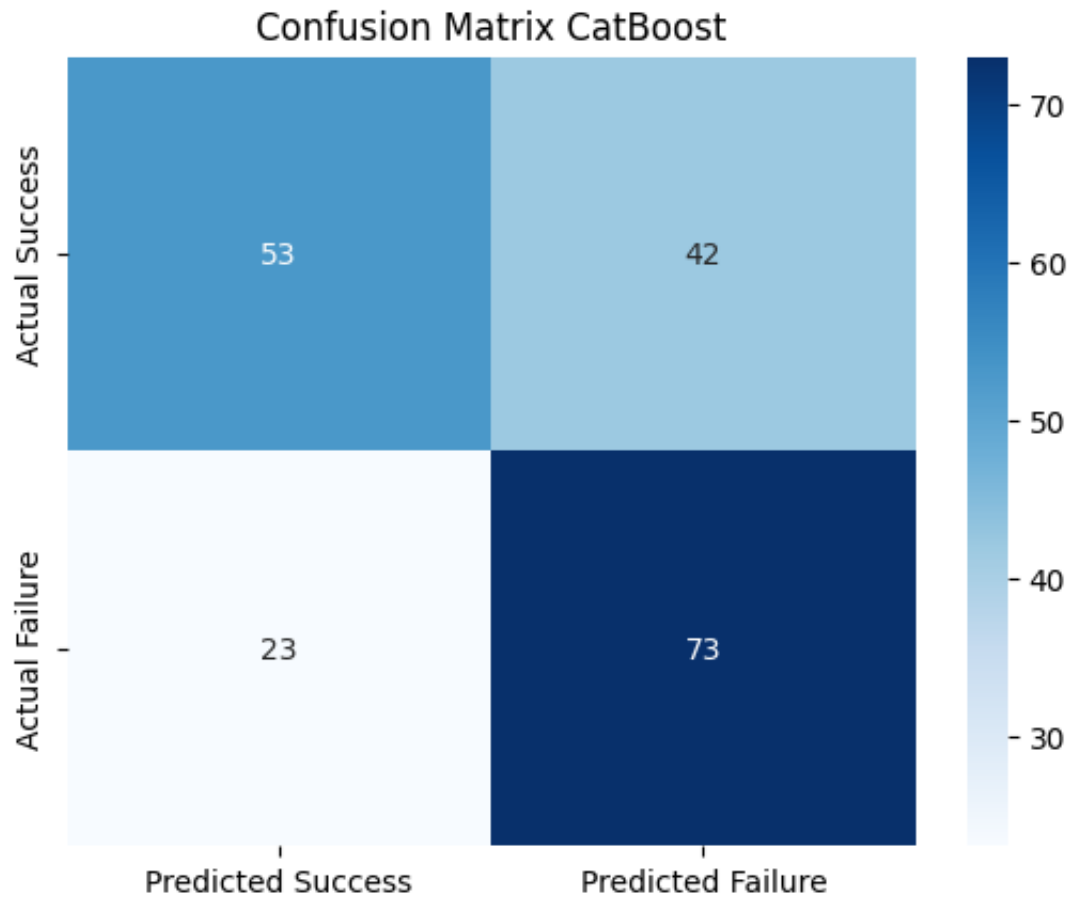
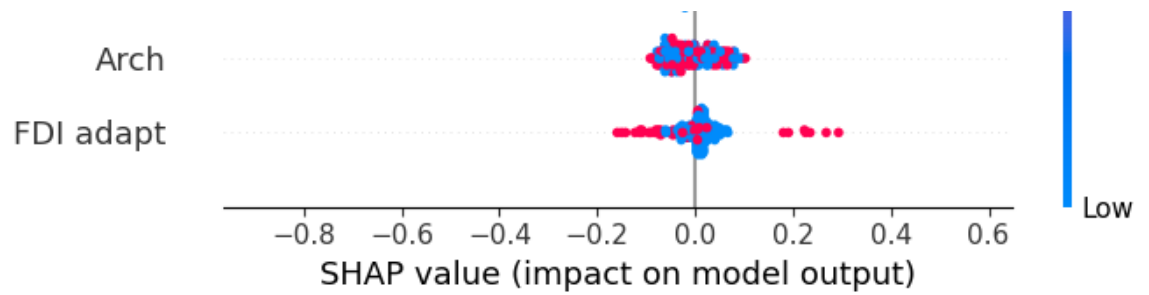
Threshold: 0.95, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

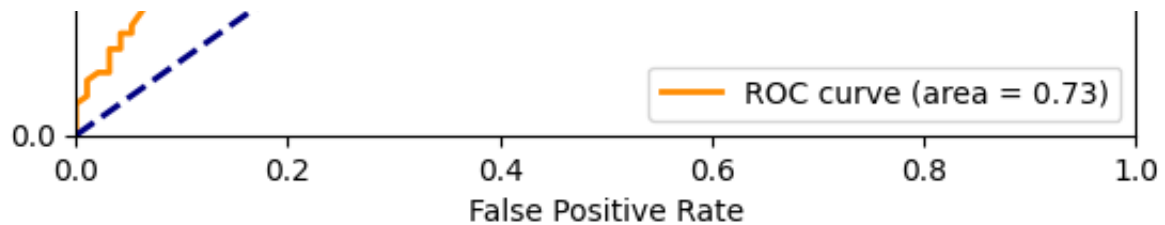
Threshold: 1.00, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

SHAP Summary for CatBoost









Running evaluation with seed 43

Inside evaluate\_catboost function

Evaluating CatBoost...

Best parameters for CatBoost: {'border\_count': 128, 'depth': 6, 'iterations

--- Dados ROC para copiar ---

FPR = [0.0, 0.0, 0.0, 0.010526315789473684, 0.010526315789473684, 0.0210526

TPR = [0.0, 0.010416666666666666, 0.03125, 0.041666666666666664, 0.05208333

AUC = 0.7354714912280702

--- Fim dos Dados ROC ---

Training - Accuracy: 0.8318385650224215, Sensitivity: 0.8465116279069768, S  
Metrics for manual threshold 0.45:

Accuracy: 0.6649214659685864, Sensitivity: 0.7708333333333334, Specificity:

Threshold: 0.10, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.15, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.20, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.25, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.30, Metrics: {'Accuracy': 0.5445026178010471, 'Sensitivity': 1

Threshold: 0.35, Metrics: {'Accuracy': 0.581151832460733, 'Sensitivity': 0.

Threshold: 0.40, Metrics: {'Accuracy': 0.6230366492146597, 'Sensitivity': 0

Threshold: 0.45, Metrics: {'Accuracy': 0.6649214659685864, 'Sensitivity': 0

Threshold: 0.50, Metrics: {'Accuracy': 0.680628272251309, 'Sensitivity': 0.

Threshold: 0.55, Metrics: {'Accuracy': 0.6282722513089005, 'Sensitivity': 0

Threshold: 0.60, Metrics: {'Accuracy': 0.5916230366492147, 'Sensitivity': 0

Threshold: 0.65, Metrics: {'Accuracy': 0.5287958115183246, 'Sensitivity': 0

Threshold: 0.70, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.75, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.80, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

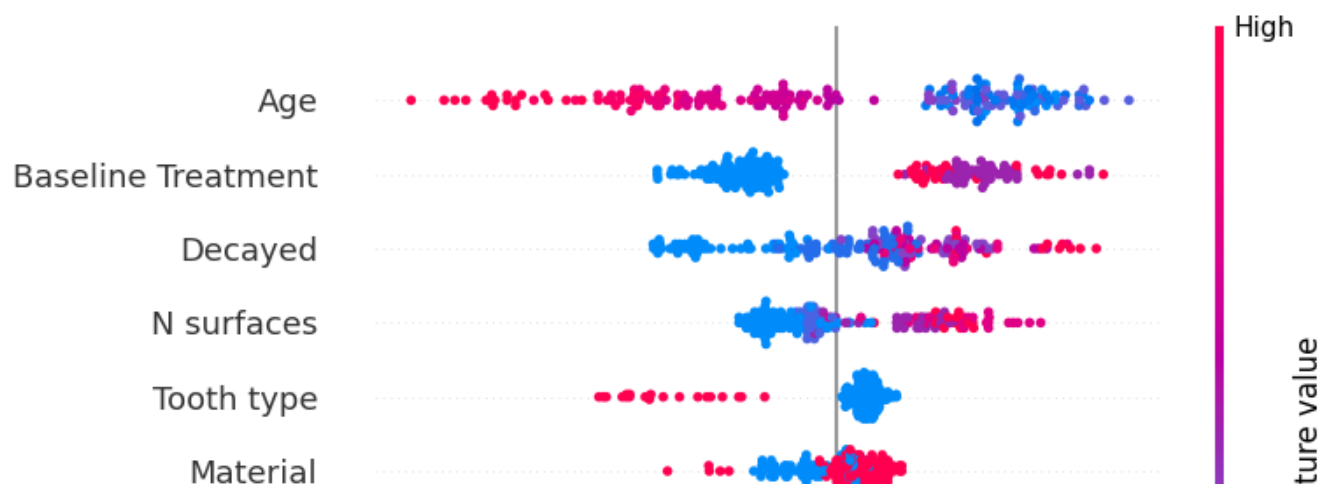
Threshold: 0.85, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

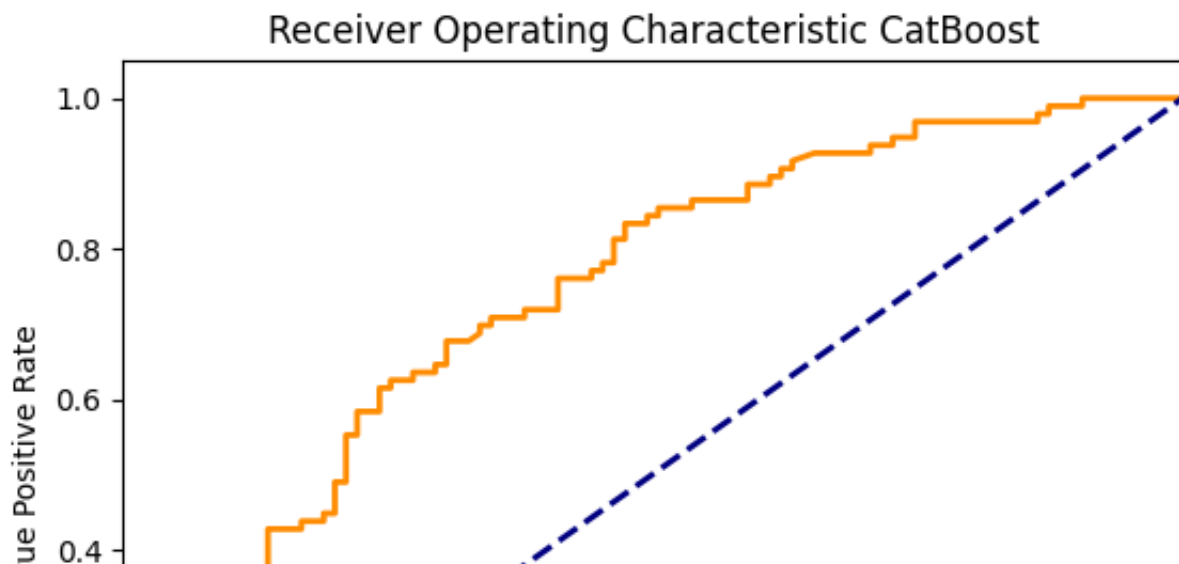
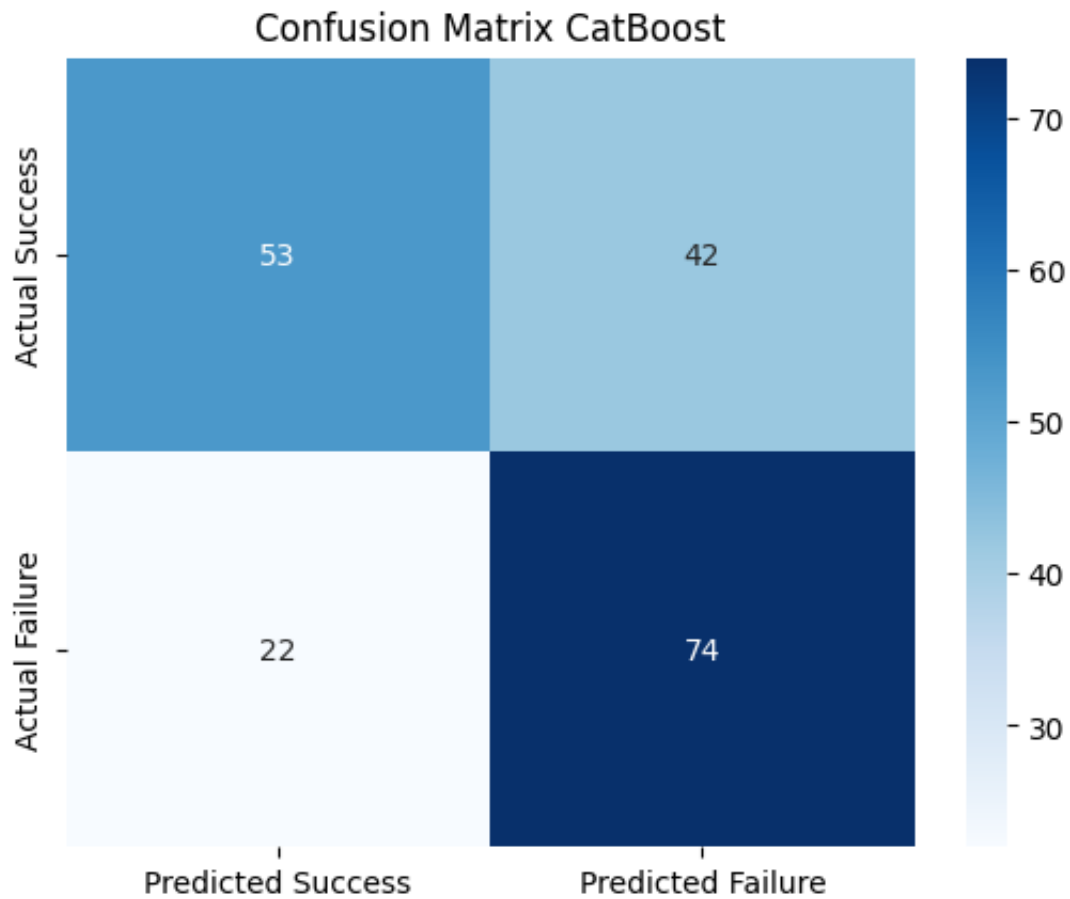
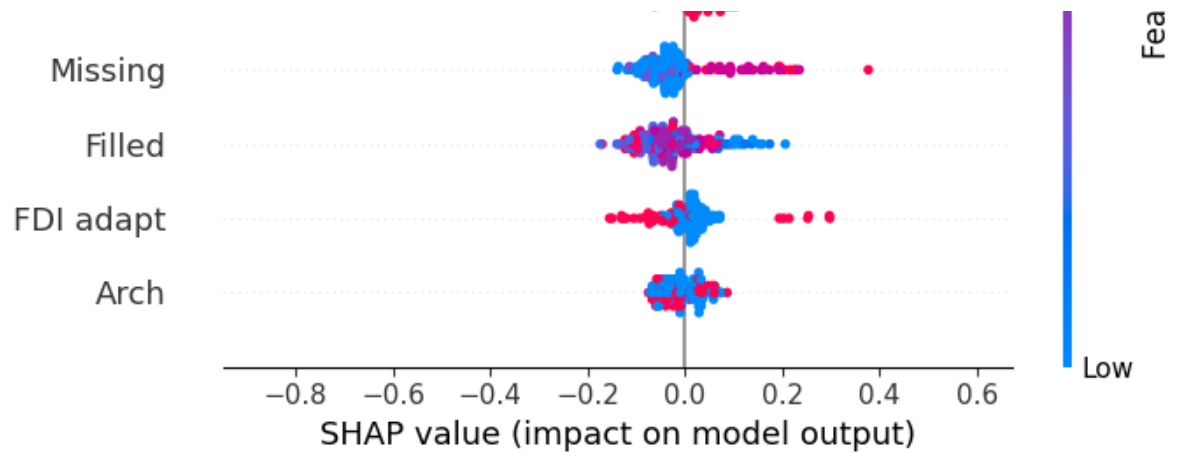
Threshold: 0.90, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

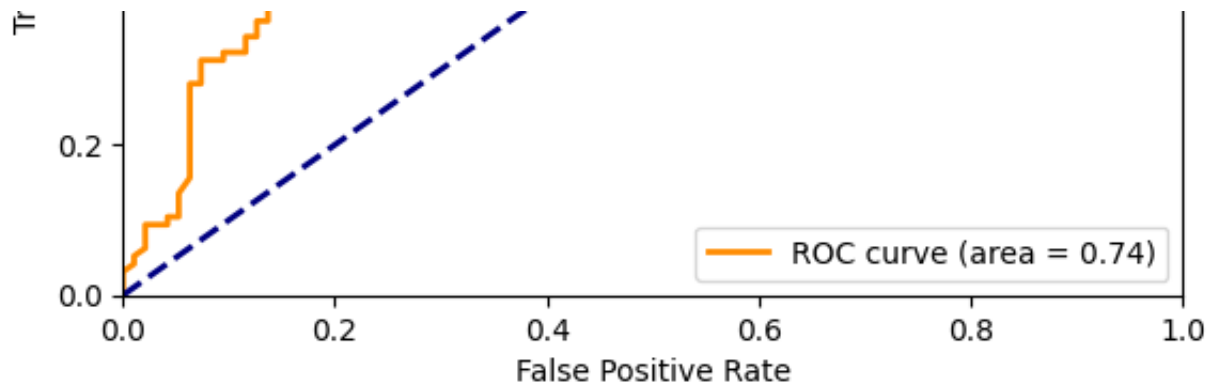
Threshold: 0.95, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 1.00, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

SHAP Summary for CatBoost







Running evaluation with seed 44

Inside evaluate\_catboost function

Evaluating CatBoost...

Best parameters for CatBoost: {'border\_count': 128, 'depth': 6, 'iterations

--- Dados ROC para copiar ---

FPR = [0.0, 0.0, 0.0, 0.010526315789473684, 0.010526315789473684, 0.0210526

TPR = [0.0, 0.010416666666666666, 0.041666666666666664, 0.05208333333333333

AUC = 0.7307565789473685

--- Fim dos Dados ROC ---

Training - Accuracy: 0.8295964125560538, Sensitivity: 0.8511627906976744, S  
Metrics for manual threshold 0.45:

Accuracy: 0.6596858638743456, Sensitivity: 0.78125, Specificity: 0.53684210

Threshold: 0.10, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.15, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.20, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.25, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.30, Metrics: {'Accuracy': 0.5497382198952879, 'Sensitivity': 1

Threshold: 0.35, Metrics: {'Accuracy': 0.5916230366492147, 'Sensitivity': 0

Threshold: 0.40, Metrics: {'Accuracy': 0.6282722513089005, 'Sensitivity': 0

Threshold: 0.45, Metrics: {'Accuracy': 0.6596858638743456, 'Sensitivity': 0

Threshold: 0.50, Metrics: {'Accuracy': 0.6858638743455497, 'Sensitivity': 0

Threshold: 0.55, Metrics: {'Accuracy': 0.6335078534031413, 'Sensitivity': 0

Threshold: 0.60, Metrics: {'Accuracy': 0.6020942408376964, 'Sensitivity': 0

Threshold: 0.65, Metrics: {'Accuracy': 0.5235602094240838, 'Sensitivity': 0

Threshold: 0.70, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.75, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.80, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

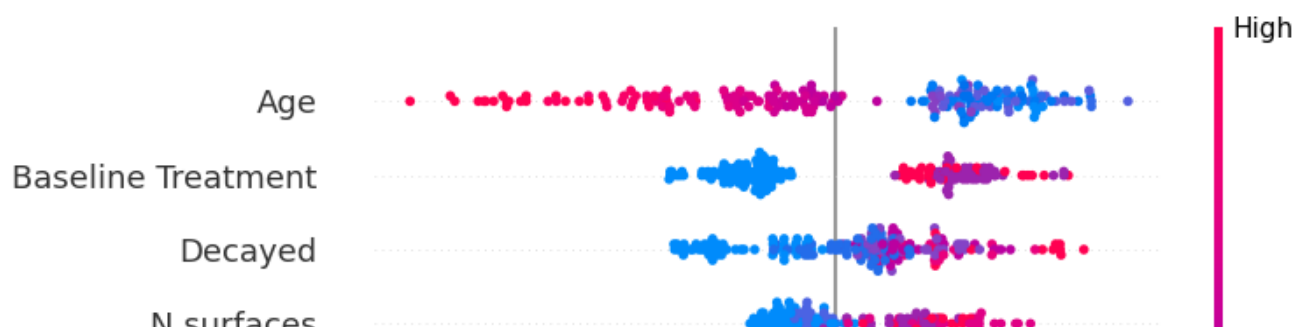
Threshold: 0.85, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

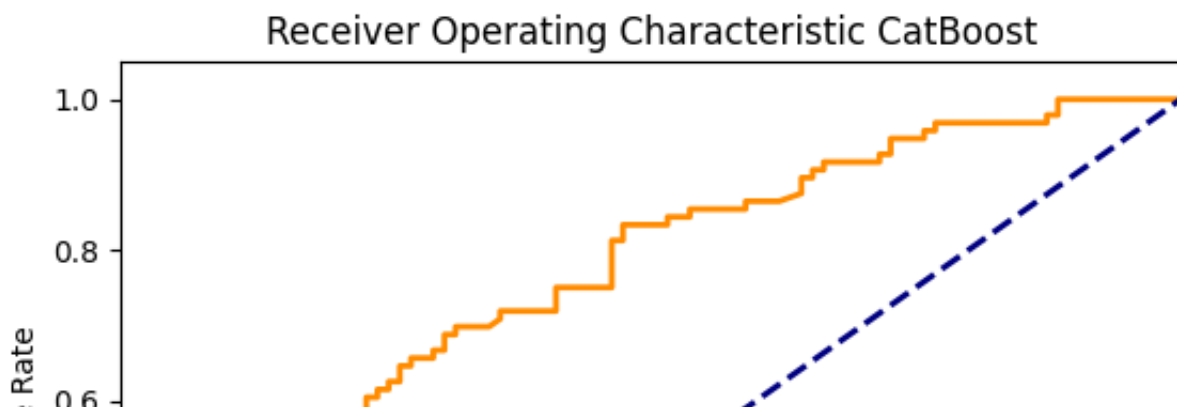
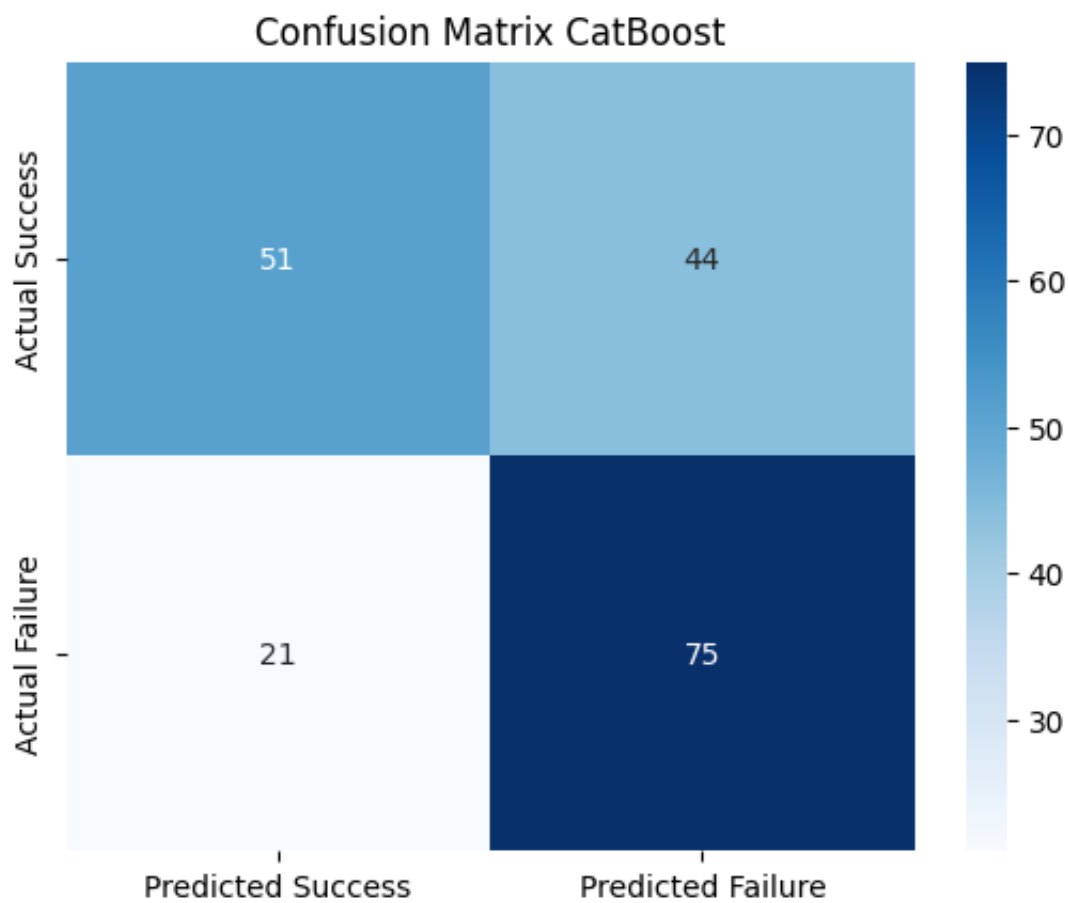
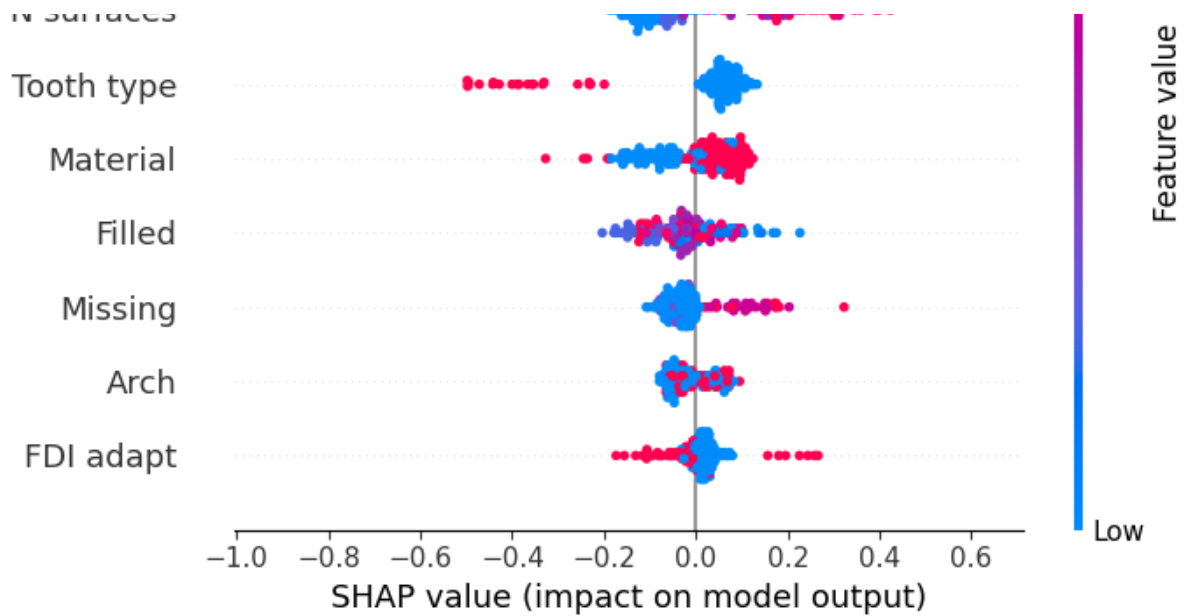
Threshold: 0.90, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

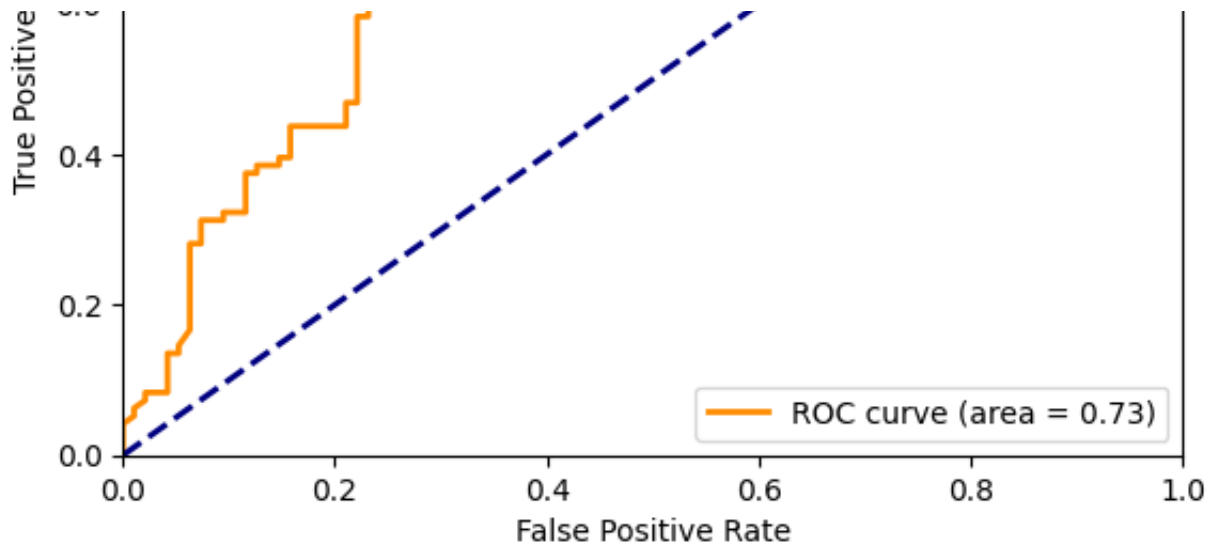
Threshold: 0.95, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 1.00, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

SHAP Summary for CatBoost







Running evaluation with seed 45

Inside evaluate\_catboost function

Evaluating CatBoost...

Best parameters for CatBoost: {'border\_count': 128, 'depth': 6, 'iterations

--- Dados ROC para copiar ---

FPR = [0.0, 0.0, 0.0, 0.010526315789473684, 0.010526315789473684, 0.0210526

TPR = [0.0, 0.010416666666666666, 0.03125, 0.041666666666666664, 0.05208333

AUC = 0.7292214912280702

--- Fim dos Dados ROC ---

Training - Accuracy: 0.8363228699551569, Sensitivity: 0.8558139534883721, S  
Metrics for manual threshold 0.45:

Accuracy: 0.6701570680628273, Sensitivity: 0.78125, Specificity: 0.55789473

Threshold: 0.10, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.15, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.20, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.25, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.30, Metrics: {'Accuracy': 0.5497382198952879, 'Sensitivity': 1

Threshold: 0.35, Metrics: {'Accuracy': 0.5968586387434555, 'Sensitivity': 0

Threshold: 0.40, Metrics: {'Accuracy': 0.6335078534031413, 'Sensitivity': 0

Threshold: 0.45, Metrics: {'Accuracy': 0.6701570680628273, 'Sensitivity': 0

Threshold: 0.50, Metrics: {'Accuracy': 0.675392670157068, 'Sensitivity': 0.

Threshold: 0.55, Metrics: {'Accuracy': 0.6230366492146597, 'Sensitivity': 0

Threshold: 0.60, Metrics: {'Accuracy': 0.5916230366492147, 'Sensitivity': 0

Threshold: 0.65, Metrics: {'Accuracy': 0.5235602094240838, 'Sensitivity': 0

Threshold: 0.70, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.75, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.80, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.85, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

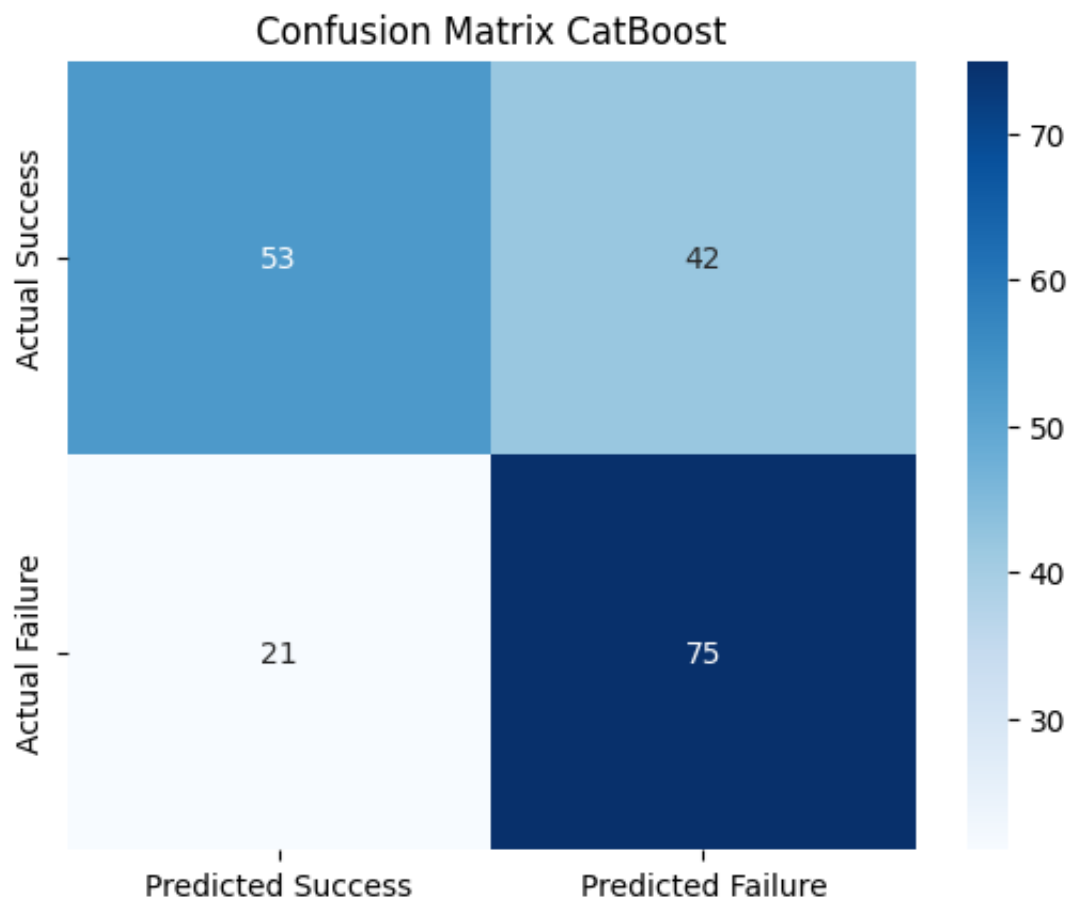
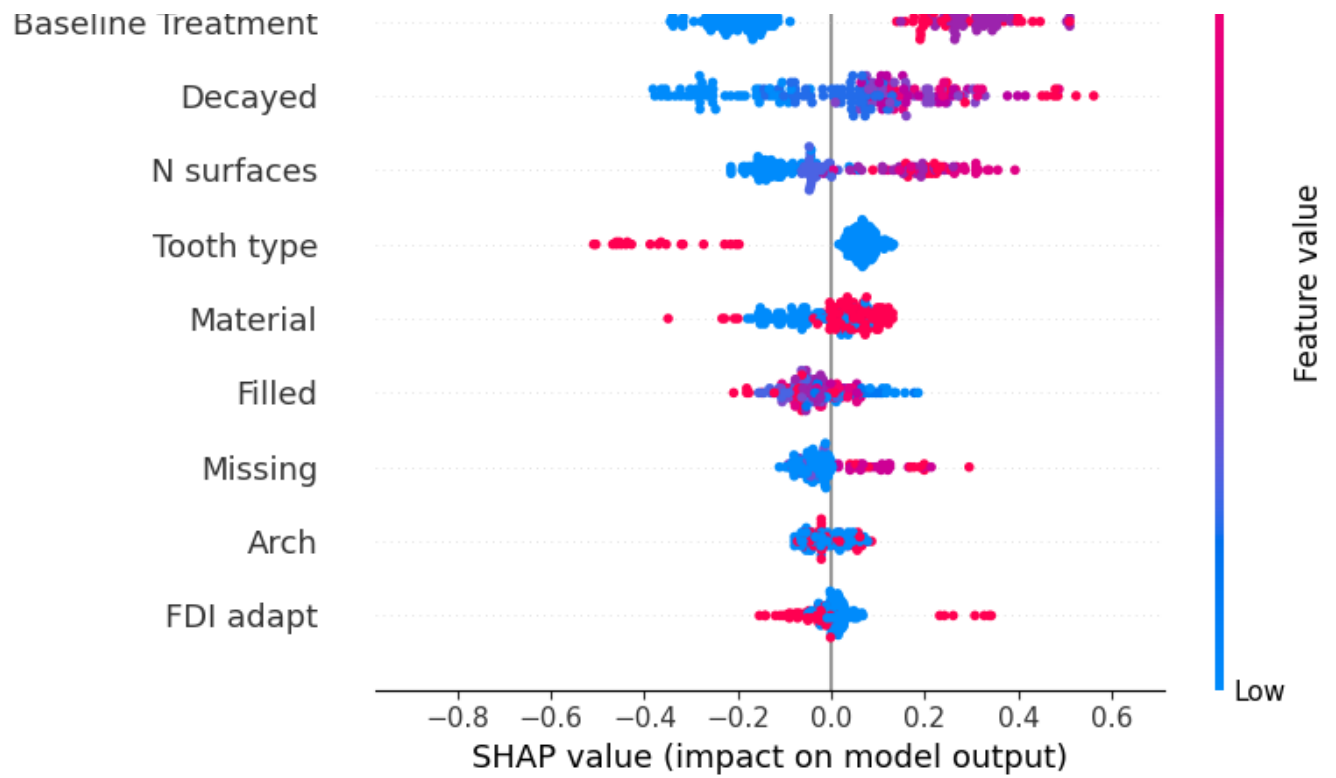
Threshold: 0.90, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.95, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

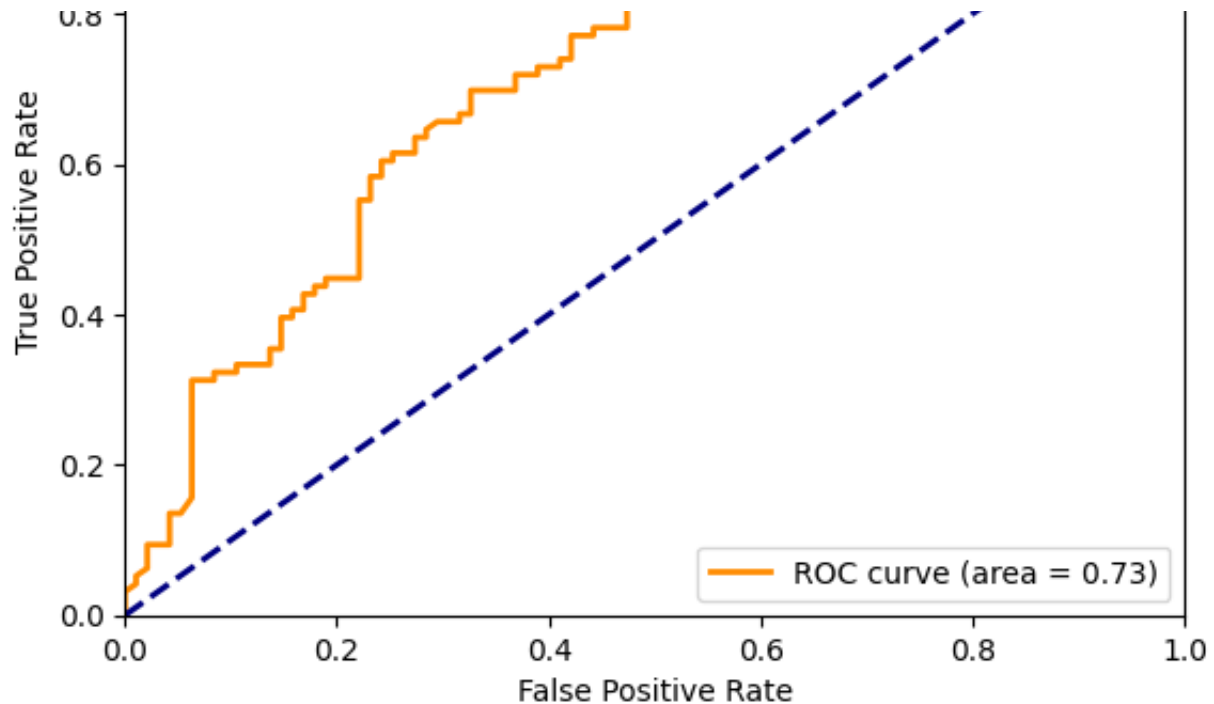
Threshold: 1.00, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

SHAP Summary for CatBoost









Running evaluation with seed 46

Inside evaluate\_catboost function

Evaluating CatBoost...

Best parameters for CatBoost: {'border\_count': 128, 'depth': 6, 'iterations

--- Dados ROC para copiar ---

FPR = [0.0, 0.0, 0.0, 0.010526315789473684, 0.010526315789473684, 0.0210526

TPR = [0.0, 0.010416666666666666, 0.041666666666666664, 0.05208333333333333

AUC = 0.734265350877193

--- Fim dos Dados ROC ---

Training - Accuracy: 0.8251121076233184, Sensitivity: 0.8511627906976744, S  
Metrics for manual threshold 0.45:

Accuracy: 0.6596858638743456, Sensitivity: 0.7708333333333334, Specificity:

Threshold: 0.10, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.15, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.20, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.25, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.30, Metrics: {'Accuracy': 0.5445026178010471, 'Sensitivity': 1

Threshold: 0.35, Metrics: {'Accuracy': 0.5759162303664922, 'Sensitivity': 0

Threshold: 0.40, Metrics: {'Accuracy': 0.6387434554973822, 'Sensitivity': 0

Threshold: 0.45, Metrics: {'Accuracy': 0.6596858638743456, 'Sensitivity': 0

Threshold: 0.50, Metrics: {'Accuracy': 0.6858638743455497, 'Sensitivity': 0

Threshold: 0.55, Metrics: {'Accuracy': 0.6282722513089005, 'Sensitivity': 0

Threshold: 0.60, Metrics: {'Accuracy': 0.5706806282722513, 'Sensitivity': 0

Threshold: 0.65, Metrics: {'Accuracy': 0.518324607329843, 'Sensitivity': 0.

Threshold: 0.70, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.75, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.80, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.85, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

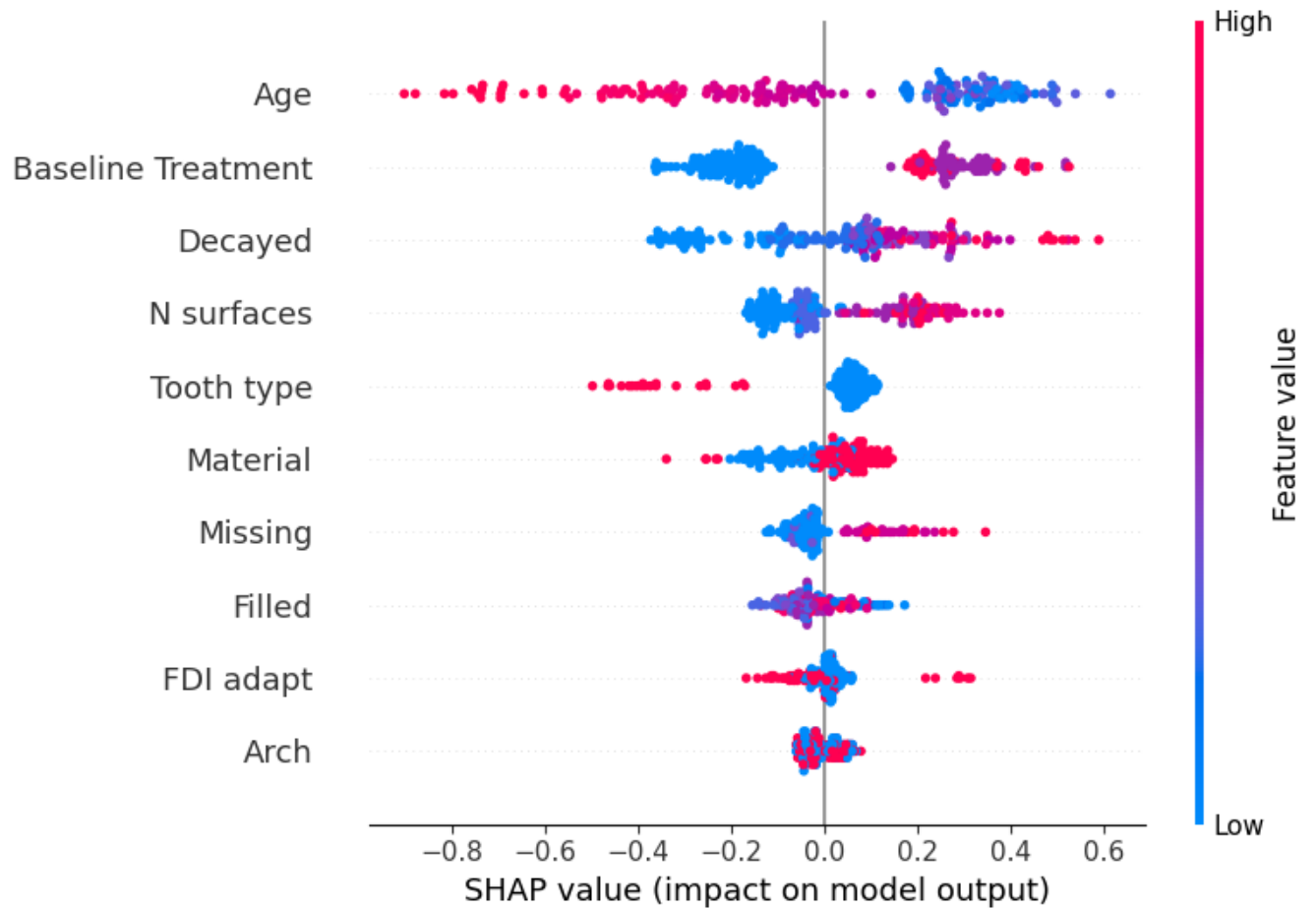
Threshold: 0.90, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.95, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

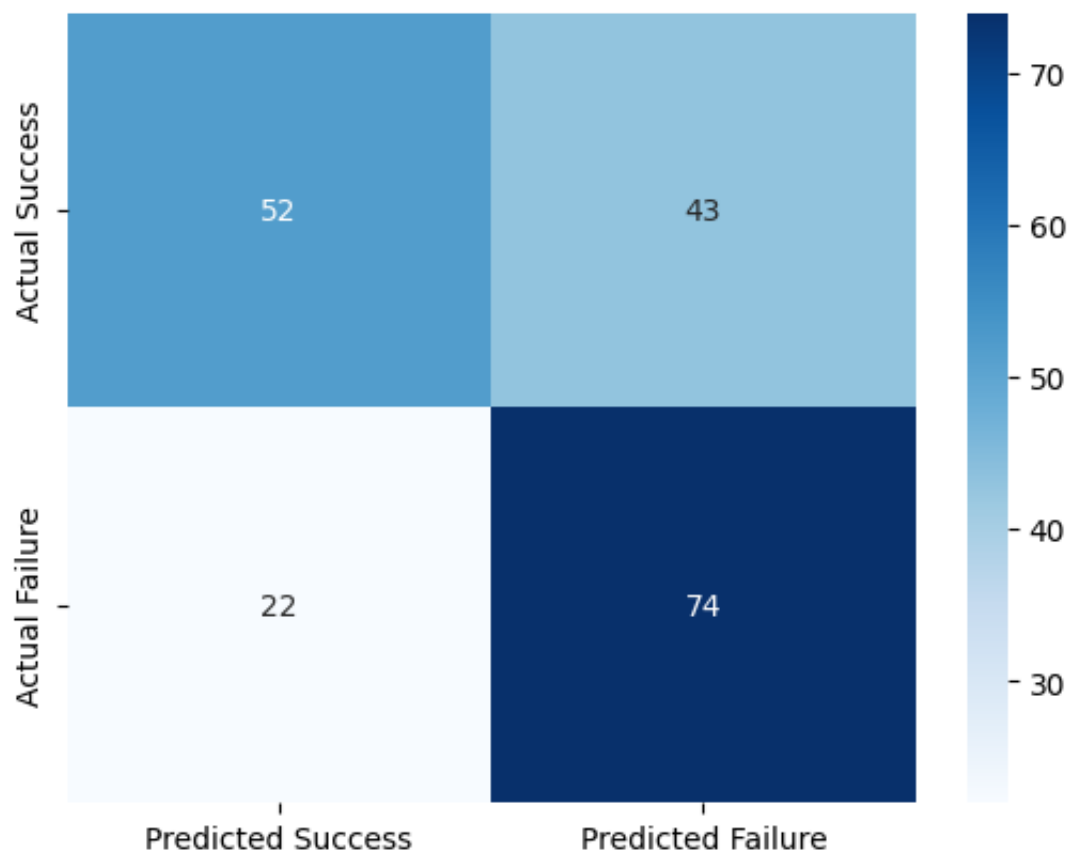
Threshold: 1.00, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

SHAP Summary for CatBoost

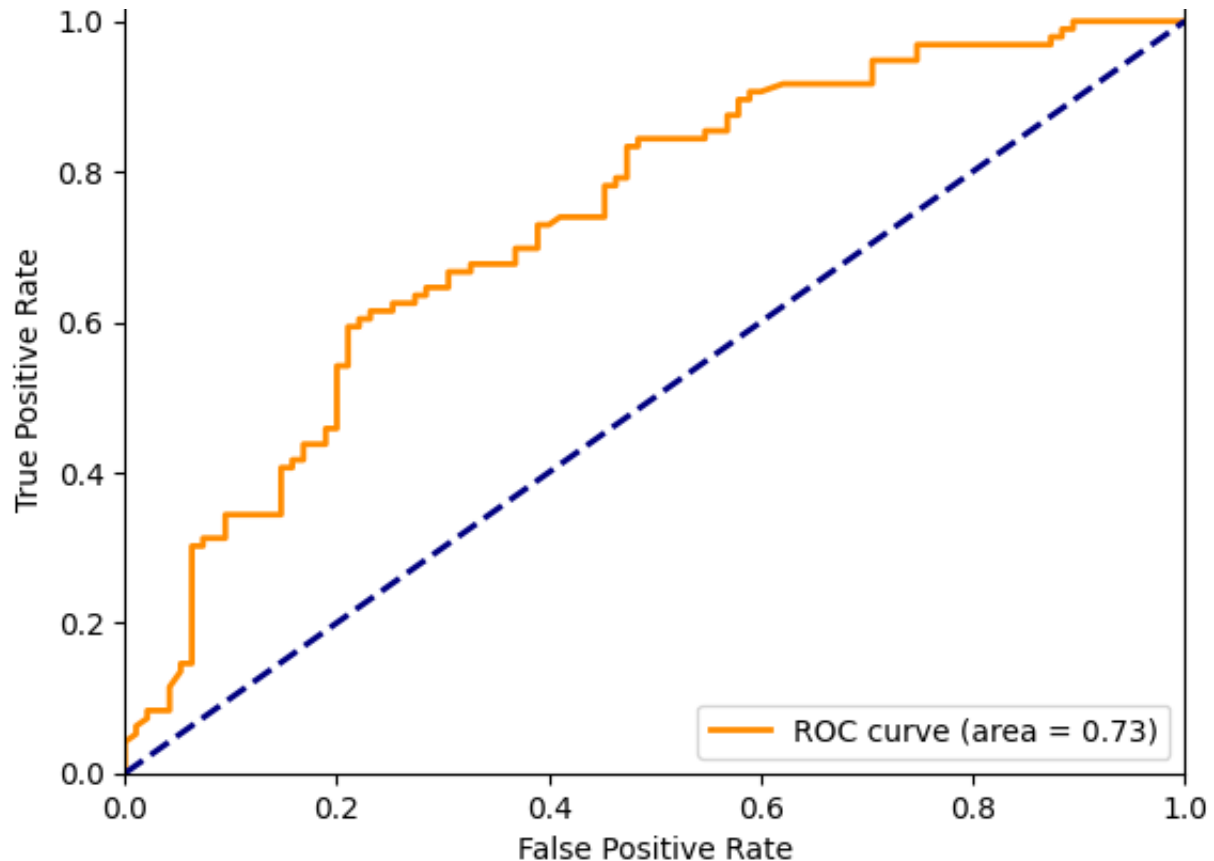




Confusion Matrix CatBoost



Receiver Operating Characteristic CatBoost



Running evaluation with seed 47

Inside evaluate\_catboost function

Evaluating CatBoost...

Best parameters for CatBoost: {'border\_count': 128, 'depth': 6, 'iterations

--- Dados ROC para copiar ---

FPR = [0.0, 0.0, 0.0, 0.010526315789473684, 0.010526315789473684, 0.0210526

TPR = [0.0, 0.010416666666666666, 0.041666666666666664, 0.05208333333333333

AUC = 0.7262609649122806

--- Fim dos Dados ROC ---

Training - Accuracy: 0.8183856502242153, Sensitivity: 0.827906976744186, Sp  
Metrics for manual threshold 0.45:

Accuracy: 0.6544502617801047, Sensitivity: 0.7604166666666666, Specificity:

Threshold: 0.10, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.15, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.20, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.25, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.30, Metrics: {'Accuracy': 0.5497382198952879, 'Sensitivity': 1

Threshold: 0.35, Metrics: {'Accuracy': 0.5968586387434555, 'Sensitivity': 0

Threshold: 0.40, Metrics: {'Accuracy': 0.6335078534031413, 'Sensitivity': 0

Threshold: 0.45, Metrics: {'Accuracy': 0.6544502617801047, 'Sensitivity': 0

Threshold: 0.50, Metrics: {'Accuracy': 0.680628272251309, 'Sensitivity': 0.

Threshold: 0.55, Metrics: {'Accuracy': 0.6282722513089005, 'Sensitivity': 0

Threshold: 0.60, Metrics: {'Accuracy': 0.5968586387434555, 'Sensitivity': 0

Threshold: 0.65, Metrics: {'Accuracy': 0.5235602094240838, 'Sensitivity': 0

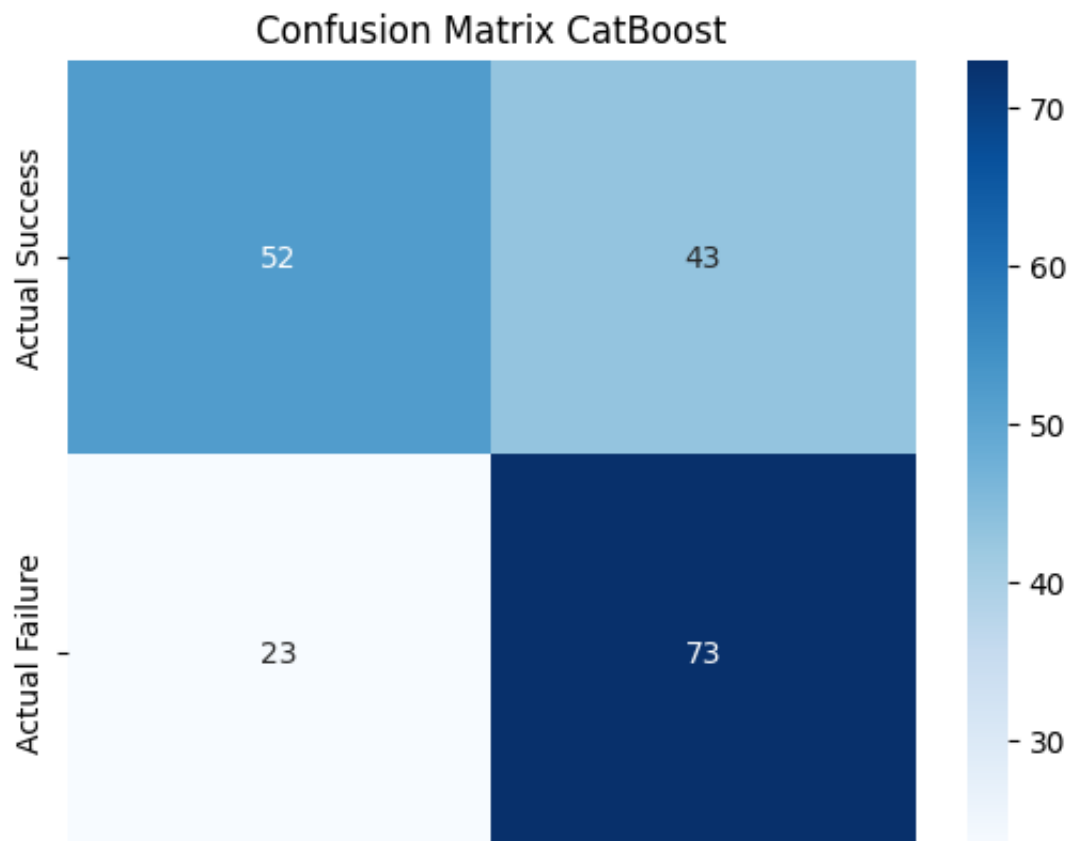
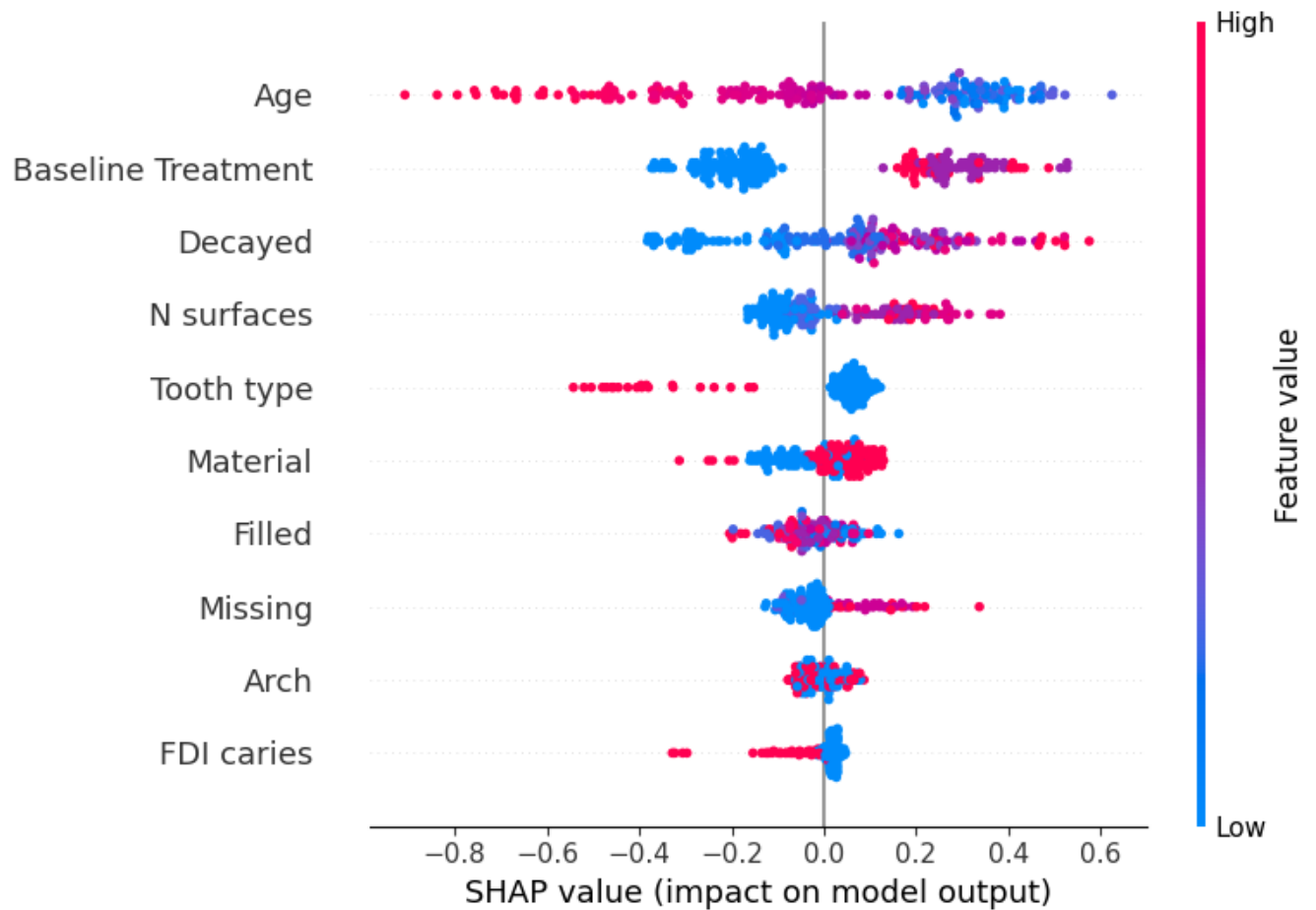
Threshold: 0.70, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

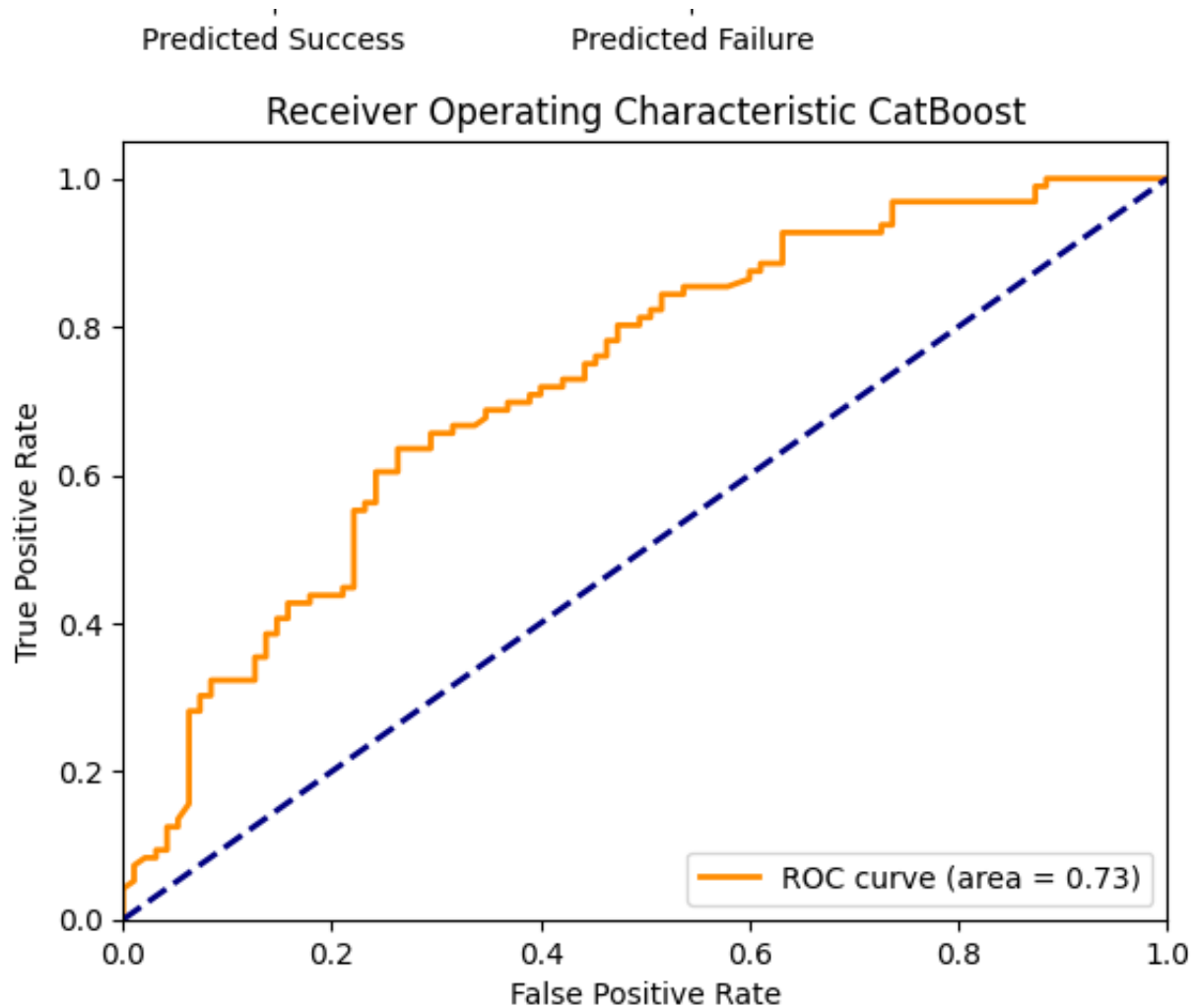
Threshold: 0.75, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.80, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.85, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

Threshold: 0.90, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0  
Threshold: 0.95, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0  
Threshold: 1.00, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0  
SHAP Summary for CatBoost





Running evaluation with seed 48

Inside evaluate\_catboost function

Evaluating CatBoost...

Best parameters for CatBoost: {'border\_count': 128, 'depth': 6, 'iterations

--- Dados ROC para copiar ---

FPR = [0.0, 0.0, 0.0, 0.010526315789473684, 0.010526315789473684, 0.0210526

TPR = [0.0, 0.010416666666666666, 0.041666666666666664, 0.05208333333333333

AUC = 0.7348135964912281

--- Fim dos Dados ROC ---

Training - Accuracy: 0.8228699551569507, Sensitivity: 0.8418604651162791, S

Metrics for manual threshold 0.45:

Accuracy: 0.6701570680628273, Sensitivity: 0.78125, Specificity: 0.55789473

Threshold: 0.10, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.15, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.20, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.25, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.30, Metrics: {'Accuracy': 0.5497382198952879, 'Sensitivity': 1

Threshold: 0.35, Metrics: {'Accuracy': 0.6073298429319371, 'Sensitivity': 0

Threshold: 0.40, Metrics: {'Accuracy': 0.6544502617801047, 'Sensitivity': 0

Threshold: 0.45, Metrics: {'Accuracy': 0.6701570680628273, 'Sensitivity': 0

Threshold: 0.50, Metrics: {'Accuracy': 0.6701570680628273, 'Sensitivity': 0

Threshold: 0.55, Metrics: {'Accuracy': 0.6335078534031413, 'Sensitivity': 0

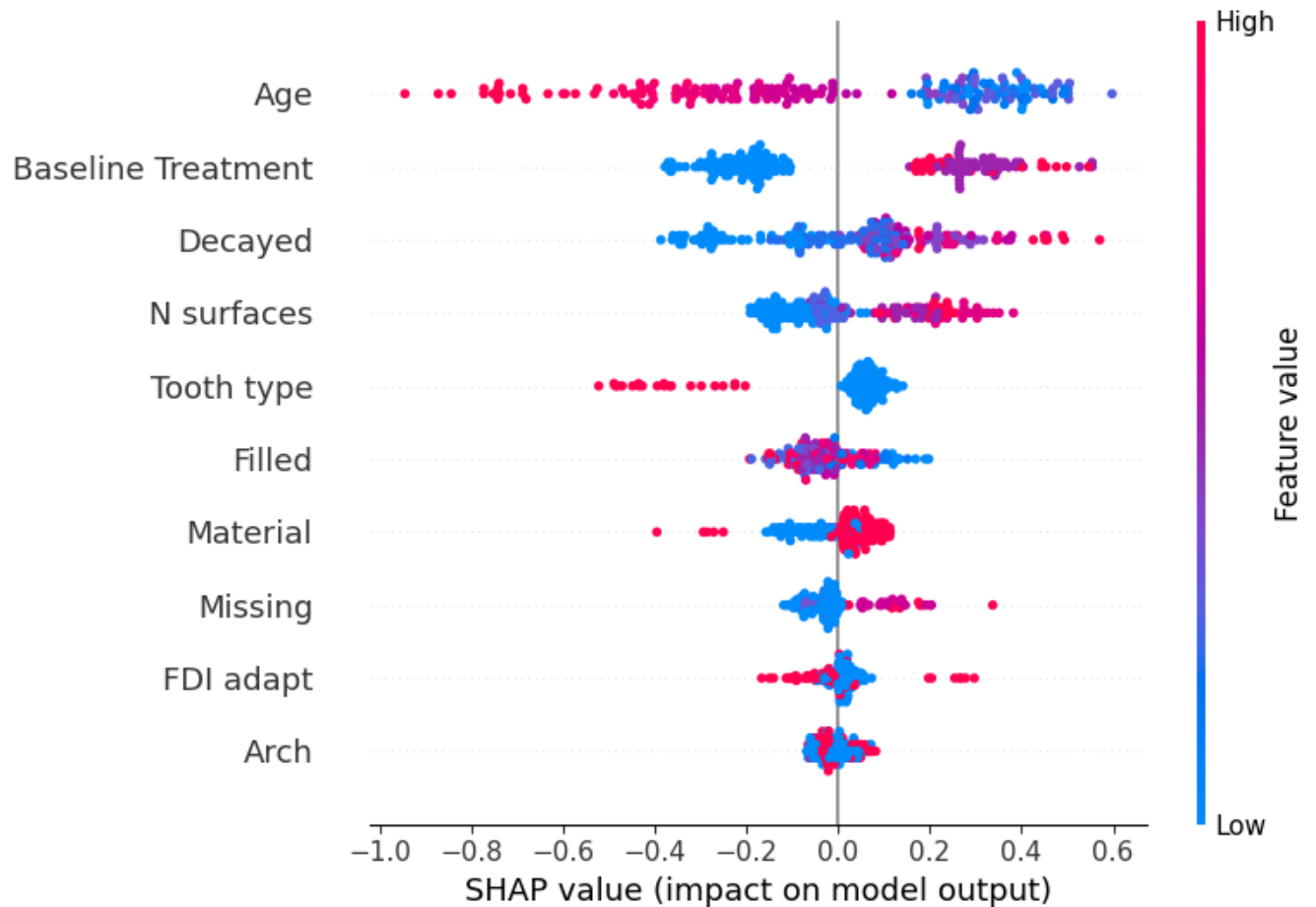
Threshold: 0.60, Metrics: {'Accuracy': 0.6073298429319371, 'Sensitivity': 0

```

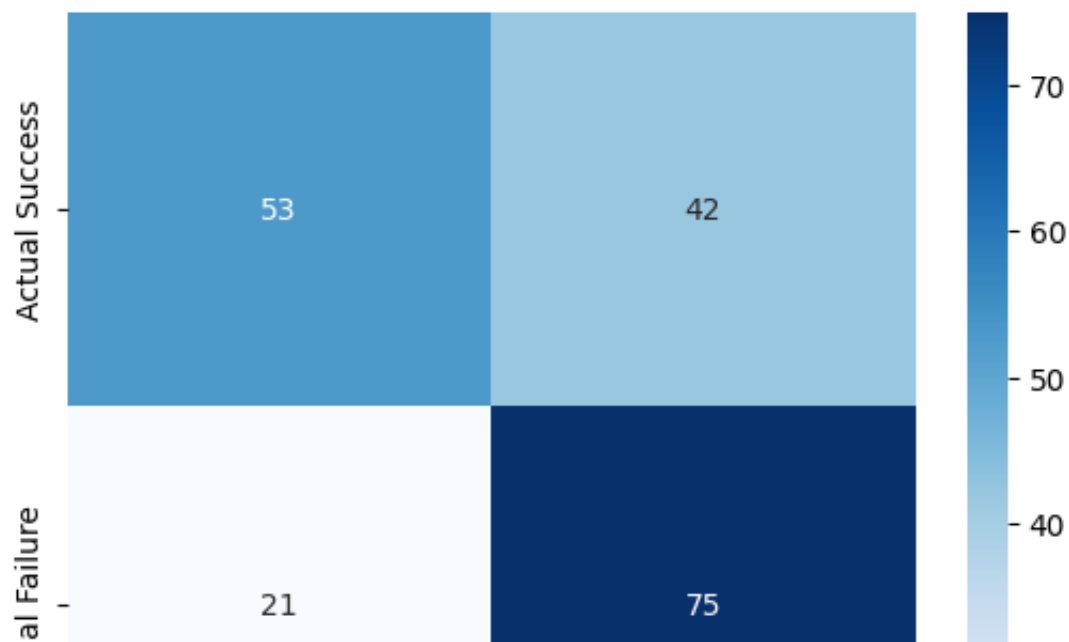
Threshold: 0.65, Metrics: {'Accuracy': 0.5235602094240838, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0

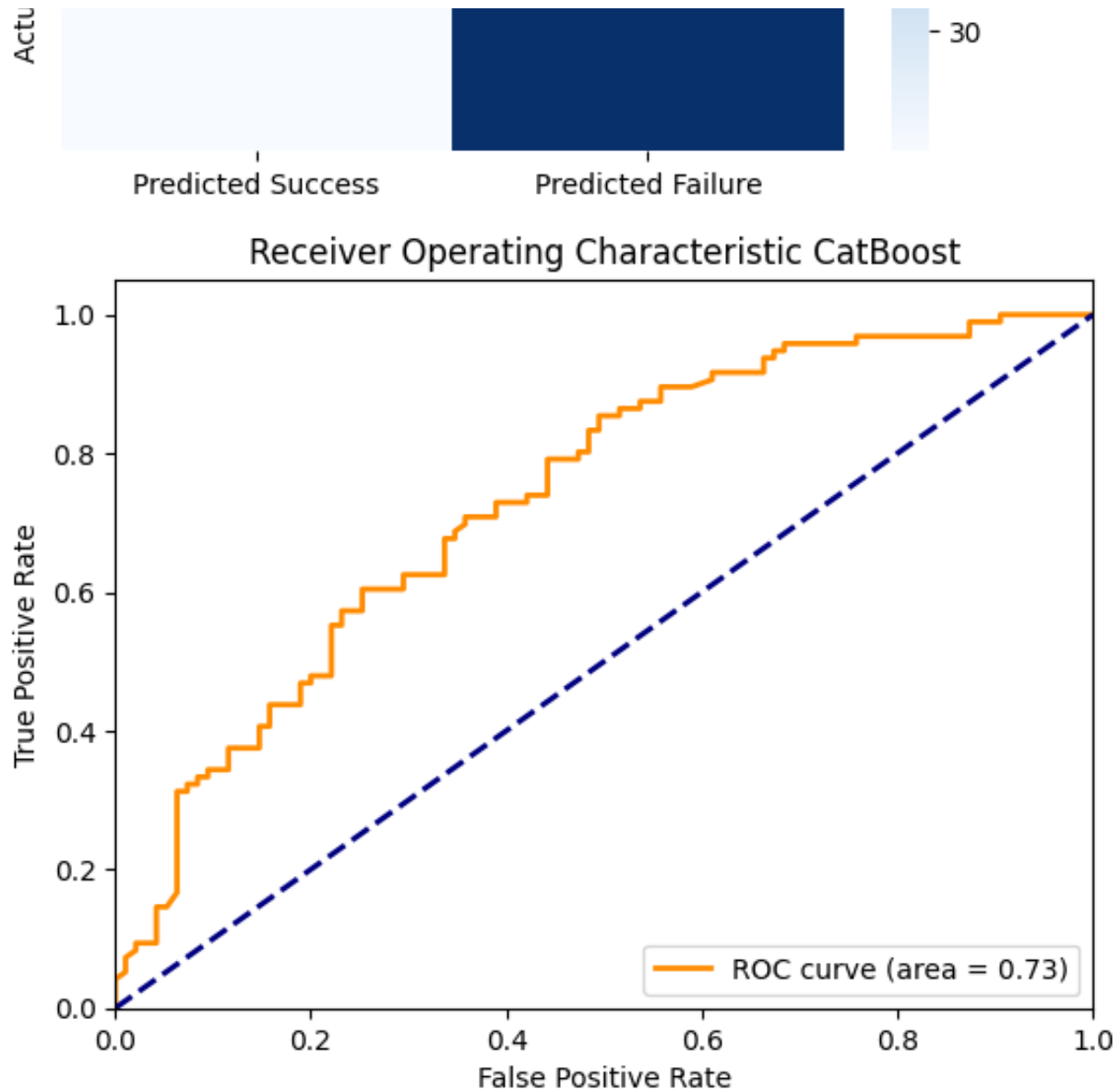
```

SHAP Summary for CatBoost



Confusion Matrix CatBoost





Running evaluation with seed 49

Inside evaluate\_catboost function

Evaluating CatBoost...

Best parameters for CatBoost: {'border\_count': 128, 'depth': 6, 'iterations

--- Dados ROC para copiar ---

FPR = [0.0, 0.0, 0.0, 0.010526315789473684, 0.010526315789473684, 0.0210526

TPR = [0.0, 0.010416666666666666, 0.03125, 0.041666666666666664, 0.0625, 0.

AUC = 0.7283442982456141

--- Fim dos Dados ROC ---

Training - Accuracy: 0.8363228699551569, Sensitivity: 0.8511627906976744, S  
Metrics for manual threshold 0.45:

Accuracy: 0.6701570680628273, Sensitivity: 0.78125, Specificity: 0.55789473

Threshold: 0.10, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.15, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.20, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.25, Metrics: {'Accuracy': 0.5026178010471204, 'Sensitivity': 1

Threshold: 0.30, Metrics: {'Accuracy': 0.5497382198952879, 'Sensitivity': 1

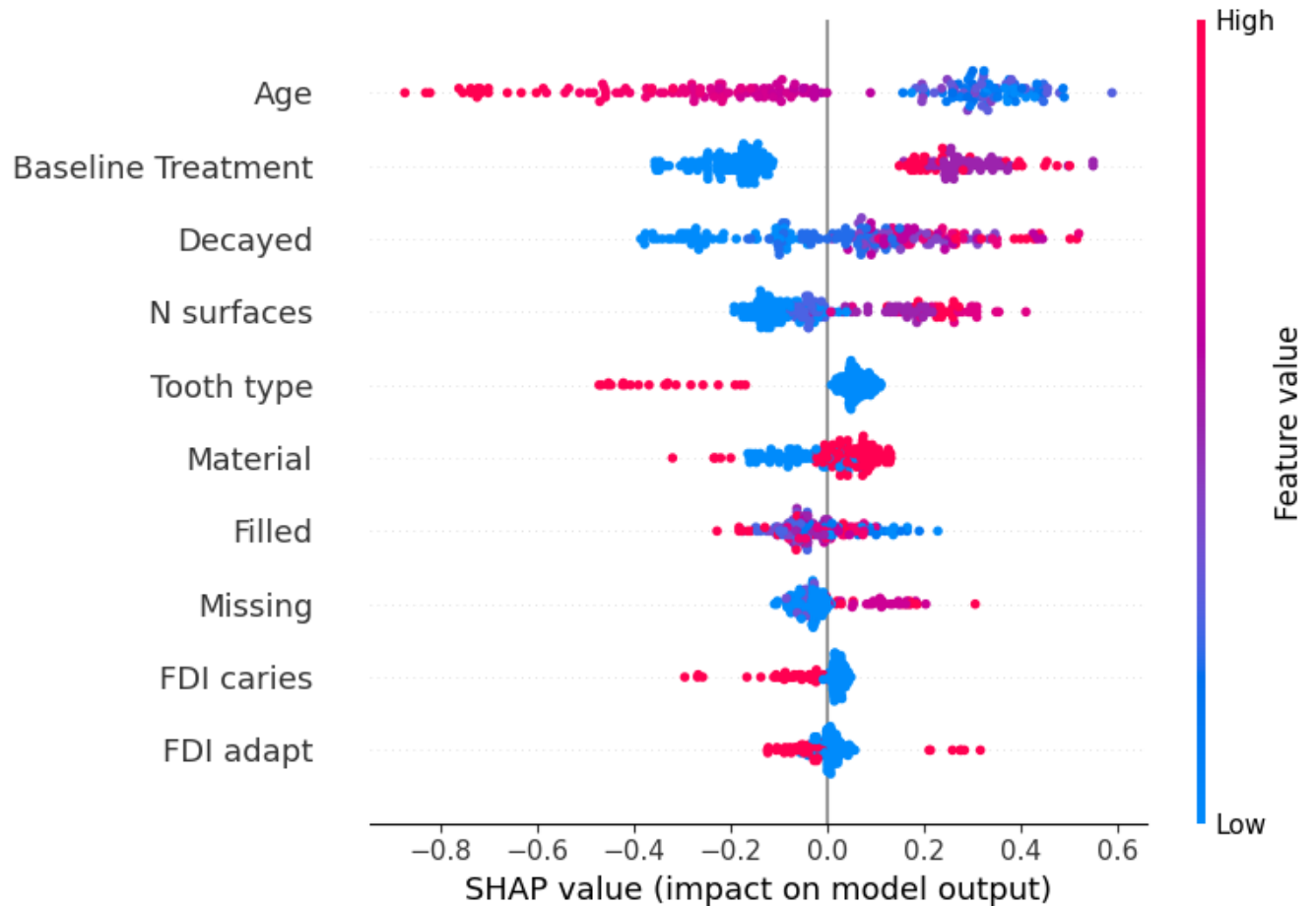
Threshold: 0.35, Metrics: {'Accuracy': 0.6073298429319371, 'Sensitivity': 0

Threshold: 0.40, Metrics: {'Accuracy': 0.6230366492146597, 'Sensitivity': 0

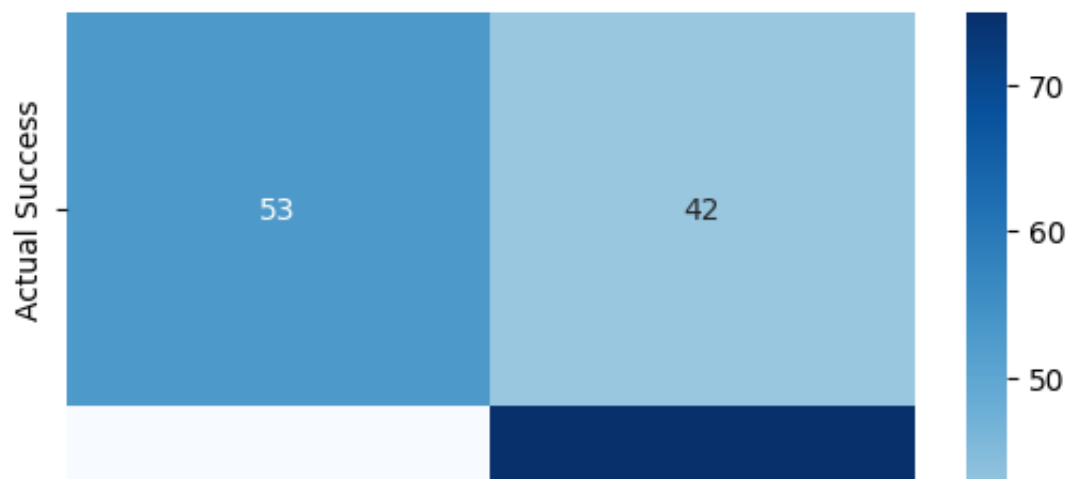
```

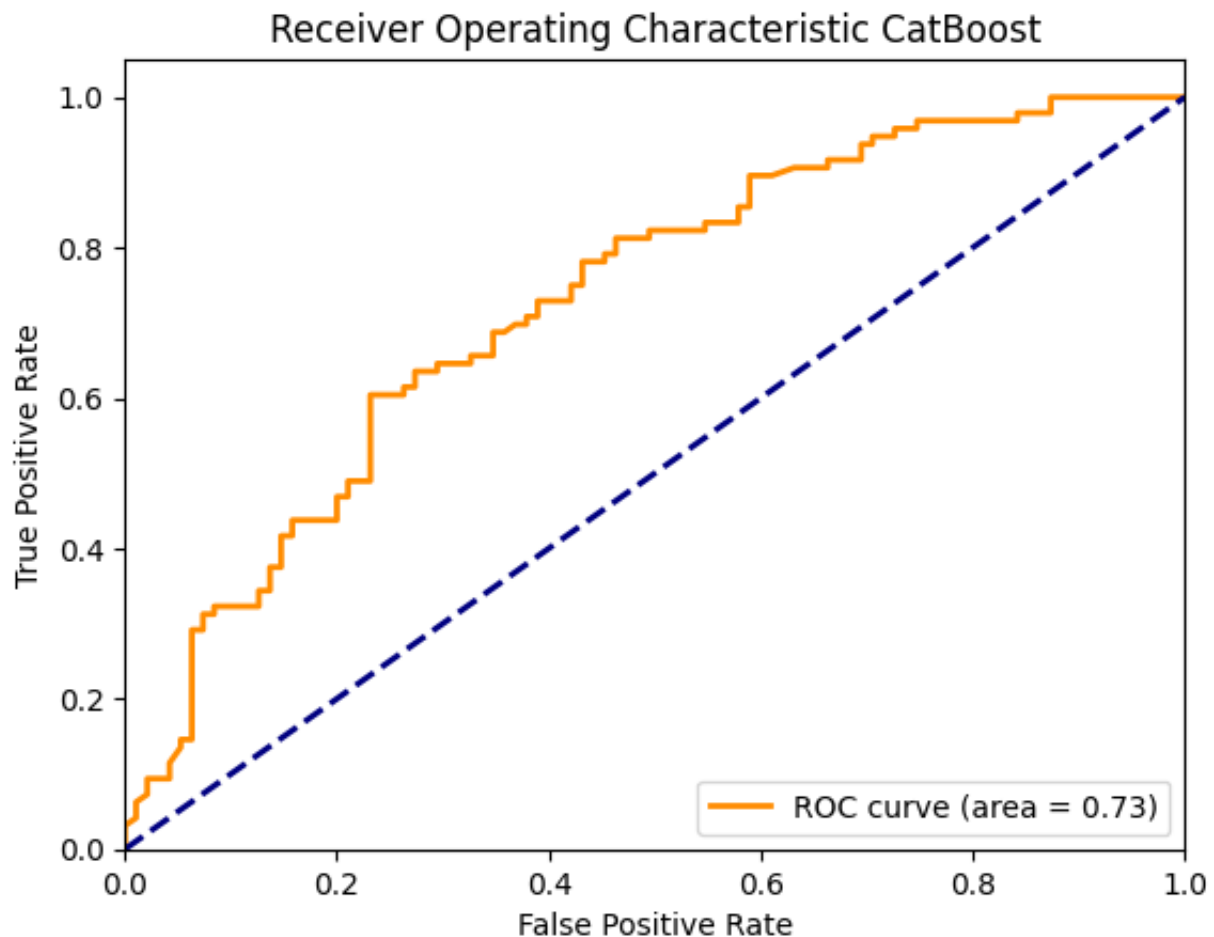
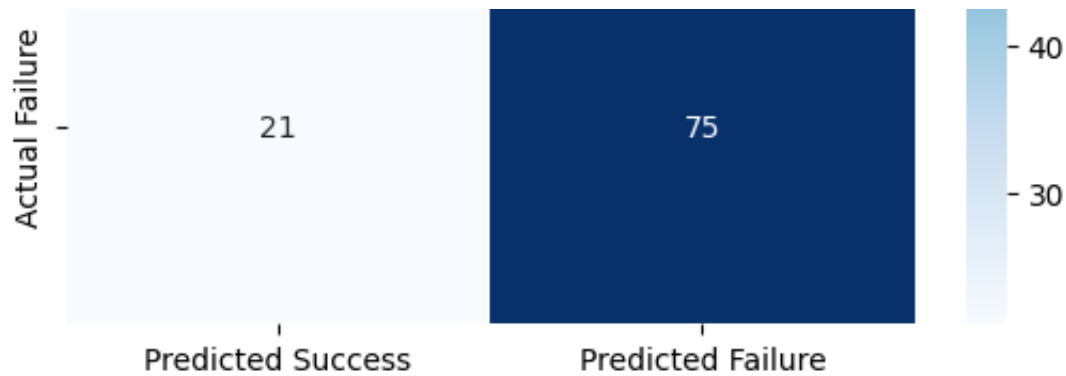
Threshold: 0.45, Metrics: {'Accuracy': 0.6701570680628273, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.680628272251309, 'Sensitivity': 0.
Threshold: 0.55, Metrics: {'Accuracy': 0.6387434554973822, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.5916230366492147, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.5340314136125655, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.4973821989528796, 'Sensitivity': 0
SHAP Summary for CatBoost

```



Confusion Matrix CatBoost





Aggregated Test Set Metrics Across Seeds:

|   | accuracy | sensitivity | specificity | f1       | roc_auc  |
|---|----------|-------------|-------------|----------|----------|
| 0 | 0.649215 | 0.750000    | 0.547368    | 0.682464 | 0.729441 |
| 1 | 0.659686 | 0.781250    | 0.536842    | 0.697674 | 0.730866 |
| 2 | 0.659686 | 0.760417    | 0.557895    | 0.691943 | 0.725713 |
| 3 | 0.664921 | 0.770833    | 0.557895    | 0.698113 | 0.735471 |
| 4 | 0.659686 | 0.781250    | 0.536842    | 0.697674 | 0.730757 |
| 5 | 0.670157 | 0.781250    | 0.557895    | 0.704225 | 0.729221 |
| 6 | 0.659686 | 0.770833    | 0.547368    | 0.694836 | 0.734265 |
| 7 | 0.654450 | 0.760417    | 0.547368    | 0.688679 | 0.726261 |
| 8 | 0.670157 | 0.781250    | 0.557895    | 0.704225 | 0.734814 |
| 9 | 0.670157 | 0.781250    | 0.557895    | 0.704225 | 0.728344 |

Summary of Test Set Metrics (Mean, Standard Error, 95% Confidence Interval)

Accuracy: Mean = 0.662, SE = 0.002, 95% CI = [0.657, 0.667]

Sensitivity: Mean = 0.772, SE = 0.004, 95% CI = [0.764, 0.780]



```
Sensitivity: Mean = 0.772, SE = 0.004, 95% CI = [0.764, 0.780]  
Specificity: Mean = 0.551, SE = 0.003, 95% CI = [0.544, 0.557]  
F1: Mean = 0.696, SE = 0.002, 95% CI = [0.691, 0.702]  
Roc_auc: Mean = 0.731, SE = 0.001, 95% CI = [0.728, 0.733]
```

