

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

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

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

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

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

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

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

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

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

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

```

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

# Initialize OrdinalEncoder
ordinal_encoder = OrdinalEncoder()

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

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

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

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

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

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

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

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

import pandas as pd

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

```

```
def descriptive_statistics(X_train_resampled, y_train_resampled, X_test, y_test)
    # Merge features and target variable for descriptive statistics on the train
    train_data_resampled = pd.concat([X_train_resampled, y_train_resampled], 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("\nResampled Training Set:")
    print(train_data_resampled[numeric_vars].describe())
    print("\nTest Set:")
    print(test_data[numeric_vars].describe())

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

    # 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 test sets
descriptive_statistics(X_train_resampled, y_train_resampled, X_test, y_test)
```



CARS CAT Statistics:

Resampled Training Set:

Count: {0: 384, 1: 171, 2: 59}

Percentage: {0: 62.54071661237784, 1: 27.850162866449512, 2: 9.609120521172}

Test Set:

Count: {0: 101, 1: 33, 2: 14}

Percentage: {0: 68.24324324324324, 1: 22.2972972972973, 2: 9.45945945945946}

FDI ADAPT CAT Statistics:

Resampled Training Set:

Count: {2: 461, 1: 133, 3: 20}

Percentage: {2: 75.0814332247557, 1: 21.661237785016286, 3: 3.2573289902280}

Test Set:

Count: {2: 106, 1: 35, 3: 7}

Percentage: {2: 71.62162162162163, 1: 23.64864864864865, 3: 4.7297297297297}

FDI CAR CAT Statistics:

Resampled Training Set:

Count: {1: 366, 2: 218, 3: 30}

Percentage: {1: 59.60912052117264, 2: 35.50488599348534, 3: 4.8859934853420}

Test Set:

Count: {1: 98, 2: 46, 3: 4}

Percentage: {1: 66.21621621621621, 2: 31.08108108108108, 3: 2.7027027027027}

ANY FAILURE Statistics:

Resampled Training Set:

Count: {1: 307, 0: 307}

Percentage: {1: 50.0, 0: 50.0}

Test Set:

Count: {0: 132, 1: 16}

Percentage: {0: 89.1891891891892, 1: 10.81081081081081}

FINAL DECISION Statistics:

Resampled Training Set:

Count: {0: 515, 1: 66, 2: 33}

Percentage: {0: 83.87622149837134, 1: 10.749185667752444, 2: 5.374592833876}

Test Set:

Count: {0: 118, 1: 18, 2: 12}

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

N SURFACES Statistics:

Resampled Training Set:

Count: {1.0: 263, 2.0: 154, 3.0: 62, 4.0: 22, 5.0: 4, 1.8912918949771986: 1}

Percentage: {1.0: 42.83387622149837, 2.0: 25.0814332247557, 3.0: 10.0977198}

Test Set:

Count: {1.0: 95, 2.0: 37, 3.0: 9, 4.0: 4, 5.0: 3}

Percentage: {1.0: 64.1891891891892, 2.0: 25.0, 3.0: 6.081081081081082, 4.0:

```
# 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 shap

import pandas as pd
import numpy as np
import shap
import sys
import tensorflow as tf
import matplotlib.pyplot as plt
import random
import seaborn as sns
from 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, GridSearch
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, LearningRateScheduler
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: shap in /usr/local/lib/python3.11/dist-packa
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: scikit-learn in /usr/local/lib/python3.11/di
Requirement already satisfied: tqdm>=4.27.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: packaging>20.9 in /usr/local/lib/python3.11/
Requirement already satisfied: slicer==0.0.8 in /usr/local/lib/python3.11/d
Requirement already satisfied: numba in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.11/dis
Requirement already satisfied: 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: 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: llvmlite<0.45,>=0.44.0dev0 in /usr/local/lib
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.11
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.11/d
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/pytho
Downloading catboost-1.2.7-cp311-cp311-manylinux2014_x86_64.whl (98.7 MB)
98.7/98.7 MB 7.8 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"\nEvaluating {name} with seed {seed}...")

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

    # Grid search using inner CV and nested cross-validation
    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}")

    # Calibrate the best model

```

```

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

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

# Calculate ROC metrics
fpr, tpr, thresholds = roc_curve(y_test, y_probs)
roc_auc = auc(fpr, tpr)
print("\n--- ROC Data ---")
print("FPR =", fpr.tolist())
print("TPR =", tpr.tolist())
print("AUC =", roc_auc)
print("--- End 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_roc_auc:.3f}")

# --- 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: {manual_roc_auc:.3f}")

# --- Evaluate metrics across a range of thresholds ---
threshold_metrics = {}
for threshold in threshold_list:
    y_pred_threshold = (y_probs >= threshold).astype(int)
    threshold_acc = accuracy_score(y_test, y_pred_threshold)
    threshold_sens = sensitivity(y_test, y_pred_threshold)
    threshold_spec = specificity(y_test, y_pred_threshold)
    threshold_f1 = f1_score(y_test, y_pred_threshold)
    threshold_metrics[threshold] = {

```



```

        'Accuracy': threshold_acc,
        'Sensitivity': threshold_sens,
        'Specificity': threshold_spec,
        'F1': threshold_f1,
        'ROC AUC': manual_roc_auc # same ROC AUC regardless of threshold
    }
for threshold, metrics in threshold_metrics.items():
    print(f"Threshold: {threshold:.2f}, Metrics: {metrics}")

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

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

return best_model, manual_threshold, best_params, nested_scores, calibrated_

# --- SHAP Plotting Function ---
def calculate_and_plot_shap(model, X_train, X_test, model_name):
    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)

# --- Plotting Functions ---
def plot_confusion_matrix(y_true, y_pred):
    matrix = confusion_matrix(y_true, y_pred)
    sns.heatmap(matrix, annot=True, fmt='d', cmap='Blues',
                xticklabels=['Predicted Success', 'Predicted Failure'],
                yticklabels=['Actual Success', 'Actual Failure'])
    plt.title('Confusion Matrix CatBoost')
    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_auc:.2f})')
    plt.plot([0, 1], [0, 1], color='black', lw=1, label='Random Guess')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('ROC Curve')
    plt.show()

```

```

plt.plot(fpr, tpr, 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()

print("ROC Curve Metrics:")
print("FPR:", fpr)
print("TPR:", tpr)
print("ROC AUC: {:.3f}".format(roc_auc))
return fpr, tpr, roc_auc

# --- Evaluation Function for CatBoost ---
def evaluate_catboost(X_train_resampled, y_train_resampled, X_test, y_test, cv,
                      print("Inside evaluate_catboost function")
                      model = CatBoostClassifier(verbose=0, random_seed=seed)
                      grid = {
                          'depth': [6],
                          'learning_rate': [0.5],
                          'iterations': [300],
                          'l2_leaf_reg': [15],
                          'border_count': [128],
                      }
                      return evaluate_model(model, "CatBoost", grid, X_train_resampled, y_train_re

# --- MAIN FUNCTION: AGGREGATING METRICS ACROSS SEEDS ---
def main(X_train_resampled, y_train_resampled, X_test, y_test):
    cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=10, random_state=42)
    scoring = {
        'accuracy': make_scorer(accuracy_score),
        'sensitivity': make_scorer(sensitivity),
        'specificity': make_scorer(specificity),
        'f1': make_scorer(f1_score),
        'roc_auc': make_scorer(roc_auc_score)
    }
    manual_threshold = 0.5
    threshold_list = np.arange(0.1, 1.05, 0.05)

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

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

```

```

    X_train_resampled, y_train_resampled, X_test, y_test, cv, scoring, ...
)

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

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

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

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

# Function to compute mean, standard error, and 95% CI using t-distribution
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__':
    # It is assumed that X_train_resampled, y_train_resampled, X_test, and y_tes
    main(X_train_resampled, y_train_resampled, X_test, y_test)

```



Running evaluation with seed 40
 Inside evaluate_catboost function

Evaluating CatBoost with seed 40...

Best parameters for CatBoost: {'border_count': 128, 'depth': 6, 'iterations

--- ROC Data ---

FPR = [0.0, 0.007575757575757576, 0.015151515151515152, 0.02272727272727272

TPR = [0.0, 0.0, 0.0625, 0.1875, 0.25, 0.25, 0.25, 0.25, 0.3125, 0.3125, 0.

AUC = 0.6536458333333333

--- End ROC Data ---

Training - Accuracy: 0.995, Sensitivity: 0.997, Specificity: 0.993, F1: 0.9

Test Metrics for manual threshold 0.5:

Accuracy: 0.797, Sensitivity: 0.375, Specificity: 0.848, F1: 0.286, ROC AUC

Threshold: 0.10, Metrics: {'Accuracy': 0.5067567567567568, 'Sensitivity': 0

Threshold: 0.15, Metrics: {'Accuracy': 0.6081081081081081, 'Sensitivity': 0

Threshold: 0.20, Metrics: {'Accuracy': 0.6554054054054054, 'Sensitivity': 0

Threshold: 0.25, Metrics: {'Accuracy': 0.7162162162162162, 'Sensitivity': 0

Threshold: 0.30, Metrics: {'Accuracy': 0.722972972972973, 'Sensitivity': 0.

Threshold: 0.35, Metrics: {'Accuracy': 0.7432432432432432, 'Sensitivity': 0

Threshold: 0.40, Metrics: {'Accuracy': 0.75, 'Sensitivity': 0.5, 'Specifici

Threshold: 0.45, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0

Threshold: 0.50, Metrics: {'Accuracy': 0.7972972972972973, 'Sensitivity': 0

Threshold: 0.55, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0

Threshold: 0.60, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.

Threshold: 0.65, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0

Threshold: 0.70, Metrics: {'Accuracy': 0.8513513513513513, 'Sensitivity': 0

Threshold: 0.75, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0

Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0

Threshold: 0.85, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

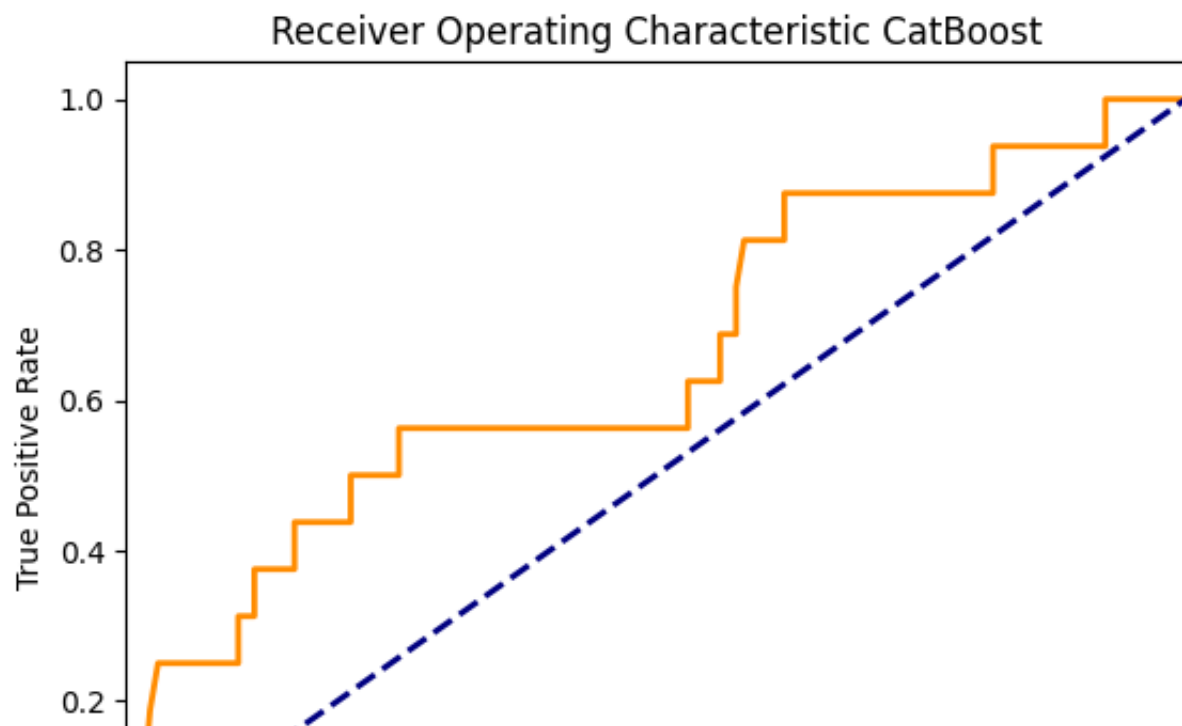
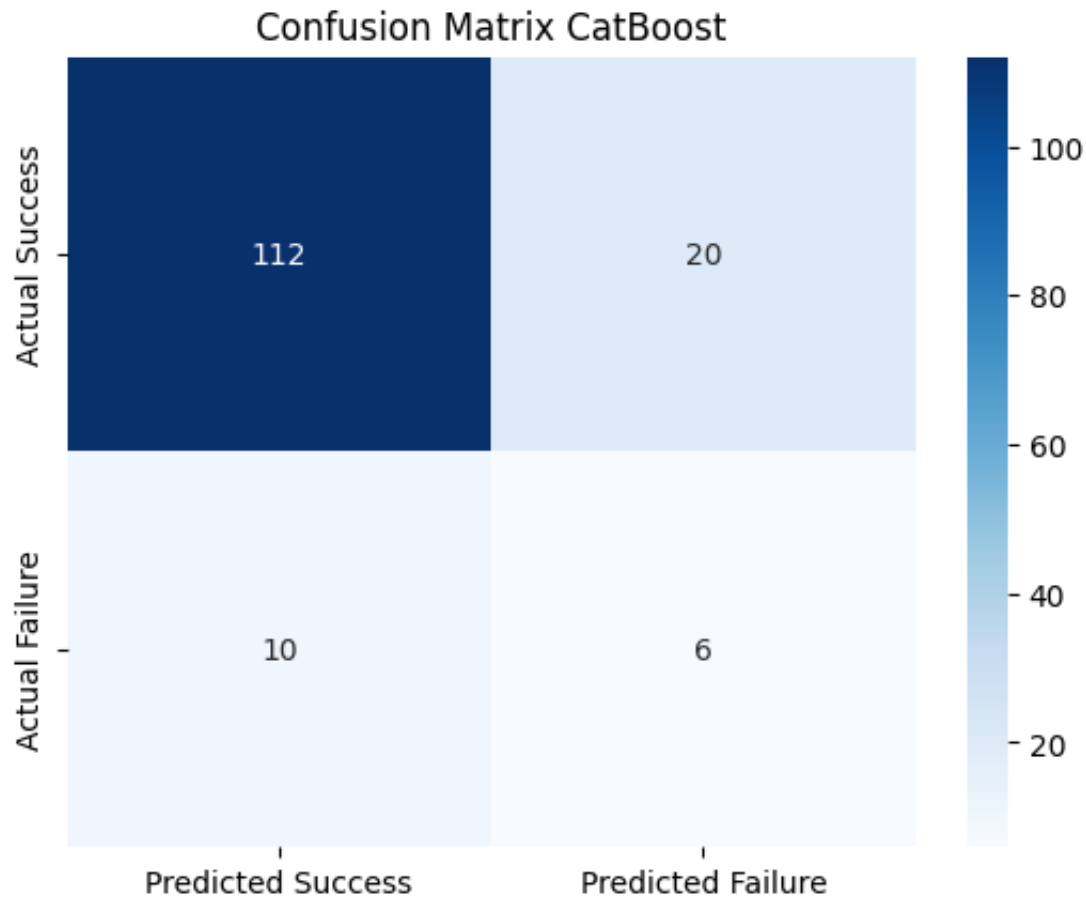
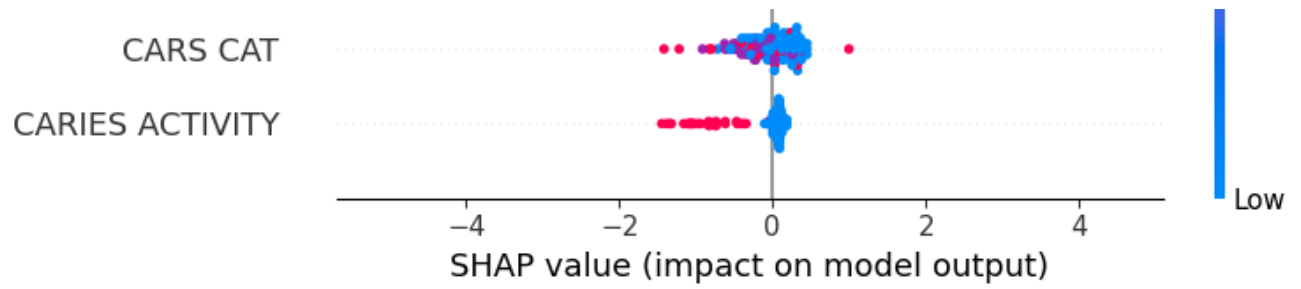
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

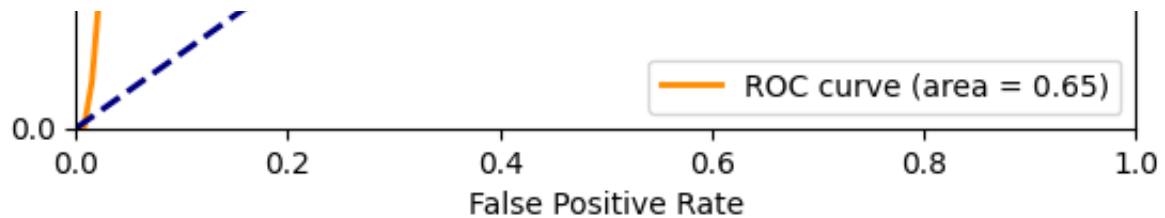
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

SHAP Summary for CatBoost







ROC Curve Metrics:

```
FPR: [0.          0.00757576 0.01515152 0.02272727 0.03030303 0.0530303
0.06818182 0.10606061 0.10606061 0.12121212 0.12121212 0.14393939
0.15909091 0.15909091 0.17424242 0.21212121 0.21212121 0.25757576
0.25757576 0.27272727 0.29545455 0.31060606 0.45454545 0.46969697
0.53030303 0.53030303 0.54545455 0.56060606 0.56060606 0.57575758
0.57575758 0.58333333 0.62121212 0.62121212 0.68181818 0.6969697
0.77272727 0.78787879 0.81818182 0.81818182 0.85606061 0.87121212
0.92424242 0.92424242 1.          ]
TPR: [0.          0.          0.0625 0.1875 0.25      0.25      0.25      0.25      0.3125 0.3125
0.375  0.375  0.375  0.4375 0.4375 0.4375 0.5       0.5       0.5625 0.5625
0.5625 0.5625 0.5625 0.5625 0.5625 0.625   0.625   0.625   0.6875 0.6875
0.75    0.8125 0.8125 0.875   0.875   0.875   0.875   0.875   0.875   0.9375
0.9375 0.9375 0.9375 1.        1.        ]
ROC AUC: 0.654
```

Running evaluation with seed 41

Inside evaluate_catboost function

Evaluating CatBoost with seed 41...

Best parameters for CatBoost: {'border_count': 128, 'depth': 6, 'iterations

--- ROC Data ---

```
FPR = [0.0, 0.007575757575757576, 0.015151515151515152, 0.02272727272727272
TPR = [0.0, 0.0, 0.0, 0.125, 0.25, 0.25, 0.25, 0.25, 0.25, 0.375, 0.375, 0.
AUC = 0.6337594696969696
```

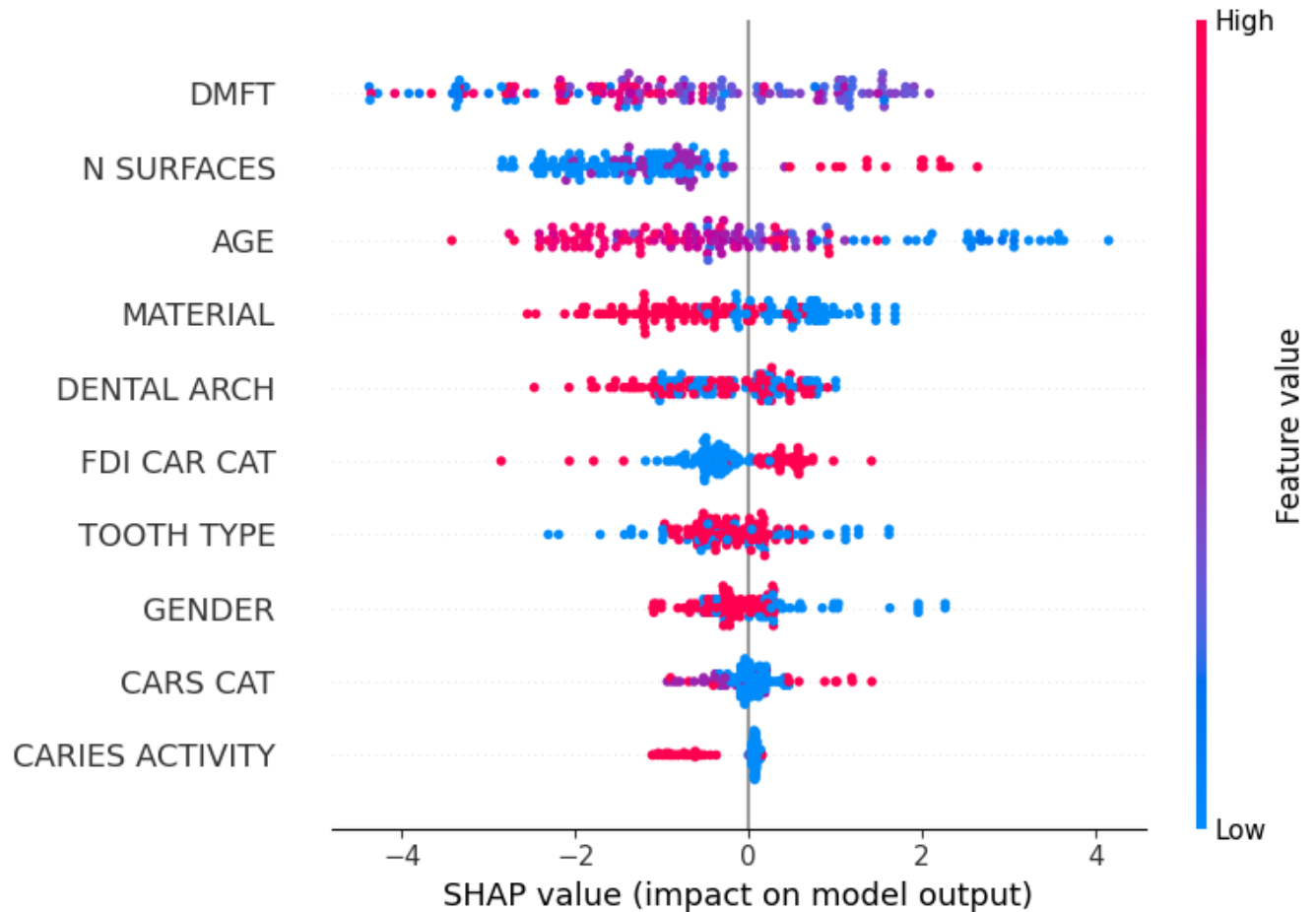
--- End ROC Data ---

Training - Accuracy: 0.995, Sensitivity: 0.997, Specificity: 0.993, F1: 0.9

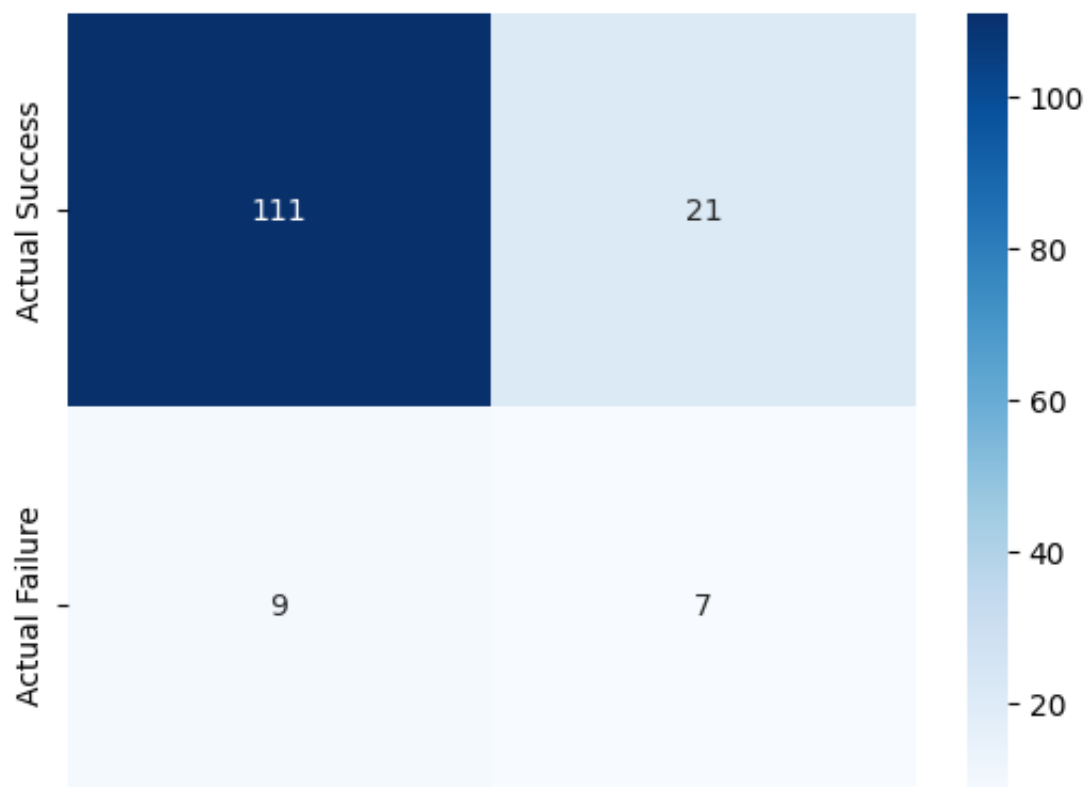
Test Metrics for manual threshold 0.5:

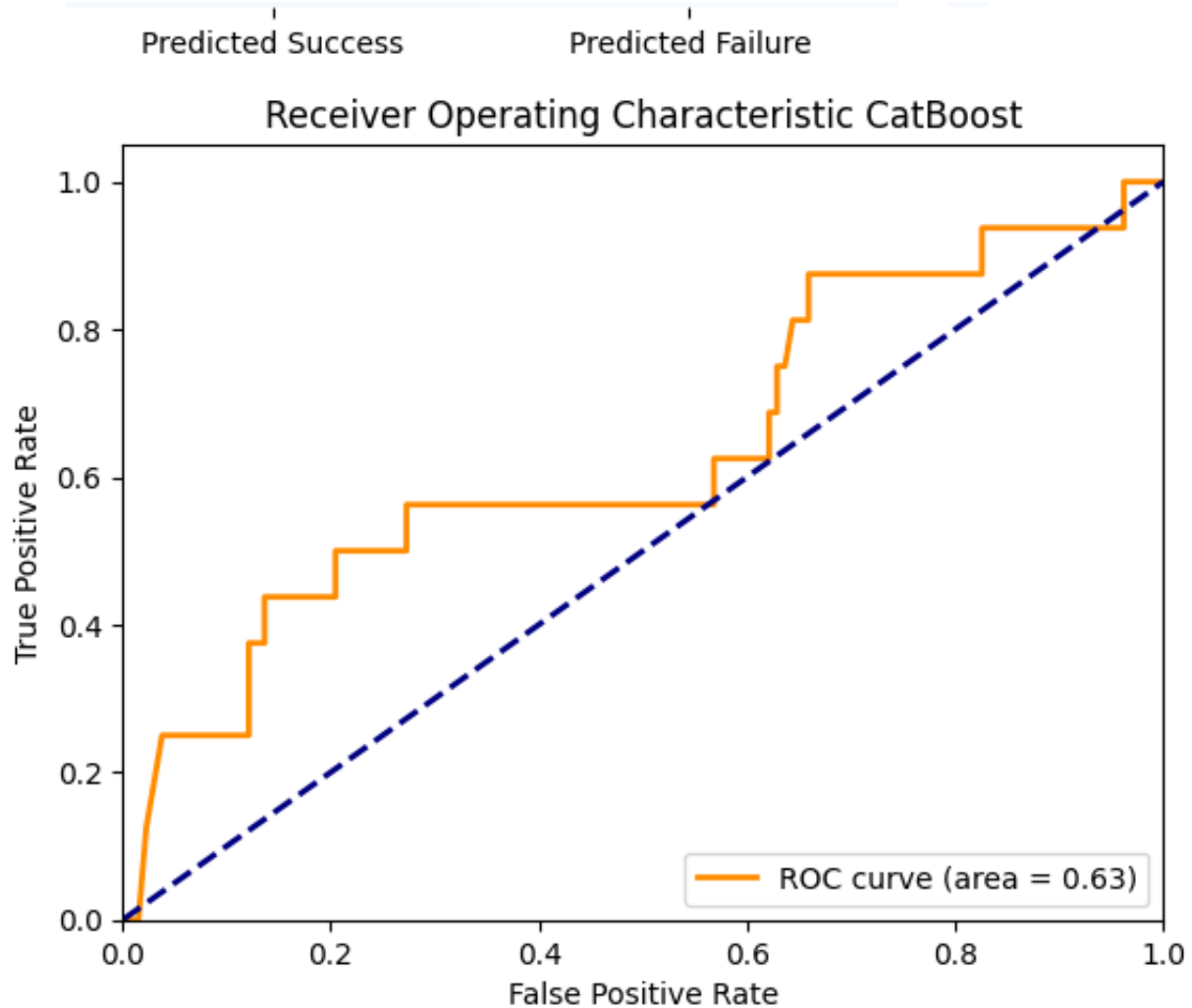
```
Accuracy: 0.797, Sensitivity: 0.438, Specificity: 0.841, F1: 0.318, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.5472972972972973, 'Sensitivity': 0
Threshold: 0.15, Metrics: {'Accuracy': 0.6351351351351351, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.6824324324324325, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.7094594594594594, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7162162162162162, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.7432432432432432, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7905405405405406, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7972972972972973, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.8513513513513513, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
```

```
Threshold: 0.89, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0.8918918918918919, 'Specificity': 0.8918918918918919}
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0.8918918918918919, 'Specificity': 0.8918918918918919}
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0.8918918918918919, 'Specificity': 0.8918918918918919}
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0.8918918918918919, 'Specificity': 0.8918918918918919}
SHAP Summary for CatBoost
```



Confusion Matrix CatBoost





ROC Curve Metrics:

```
FPR: [0.          0.00757576 0.01515152 0.02272727 0.03787879 0.06060606
0.07575758 0.09848485 0.12121212 0.12121212 0.13636364 0.13636364
0.17424242 0.18939394 0.20454545 0.20454545 0.27272727 0.27272727
0.28030303 0.3030303  0.36363636 0.37878788 0.48484848 0.5
0.50757576 0.52272727 0.56818182 0.56818182 0.62121212 0.62121212
0.62878788 0.62878788 0.63636364 0.64393939 0.65909091 0.65909091
0.68181818 0.6969697  0.74242424 0.75757576 0.78787879 0.8030303
0.82575758 0.82575758 0.96212121 0.96212121 1.          ]
```

```
TPR: [0.          0.          0.          0.125  0.25  0.25  0.25  0.25  0.25  0.375
0.375 0.4375 0.4375 0.4375 0.4375 0.5    0.5    0.5625 0.5625 0.5625
0.5625 0.5625 0.5625 0.5625 0.5625 0.5625 0.5625 0.625  0.625  0.6875
0.6875 0.75   0.75   0.8125 0.8125 0.875  0.875  0.875  0.875  0.875
0.875 0.875  0.875  0.9375 0.9375 1.      1.      ]
```

ROC AUC: 0.634

Running evaluation with seed 42

Inside evaluate_catboost function

Evaluating CatBoost with seed 42...

Best parameters for CatBoost: {'border_count': 128, 'depth': 6, 'iterations

--- ROC Data ---

FPR = [0.0, 0.007575757575757576, 0.015151515151515152, 0.02272727272727272

TPR = [0.0, 0.0, 0.125, 0.1875, 0.1875, 0.1875, 0.25, 0.25, 0.3125, 0.3125,

AUC = 0.6508049242424242

--- End ROC Data ---

Training - Accuracy: 0.995, Sensitivity: 0.997, Specificity: 0.993, F1: 0.9

Test Metrics for manual threshold 0.5:

Accuracy: 0.818, Sensitivity: 0.438, Specificity: 0.864, F1: 0.341, ROC AUC

Threshold: 0.10, Metrics: {'Accuracy': 0.5675675675675675, 'Sensitivity': 0

Threshold: 0.15, Metrics: {'Accuracy': 0.6351351351351351, 'Sensitivity': 0

Threshold: 0.20, Metrics: {'Accuracy': 0.6959459459459459, 'Sensitivity': 0

Threshold: 0.25, Metrics: {'Accuracy': 0.7162162162162162, 'Sensitivity': 0

Threshold: 0.30, Metrics: {'Accuracy': 0.7432432432432432, 'Sensitivity': 0

Threshold: 0.35, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0

Threshold: 0.40, Metrics: {'Accuracy': 0.7837837837837838, 'Sensitivity': 0

Threshold: 0.45, Metrics: {'Accuracy': 0.8040540540540541, 'Sensitivity': 0

Threshold: 0.50, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0

Threshold: 0.55, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0

Threshold: 0.60, Metrics: {'Accuracy': 0.8513513513513513, 'Sensitivity': 0

Threshold: 0.65, Metrics: {'Accuracy': 0.8513513513513513, 'Sensitivity': 0

Threshold: 0.70, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0

Threshold: 0.75, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0

Threshold: 0.80, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0

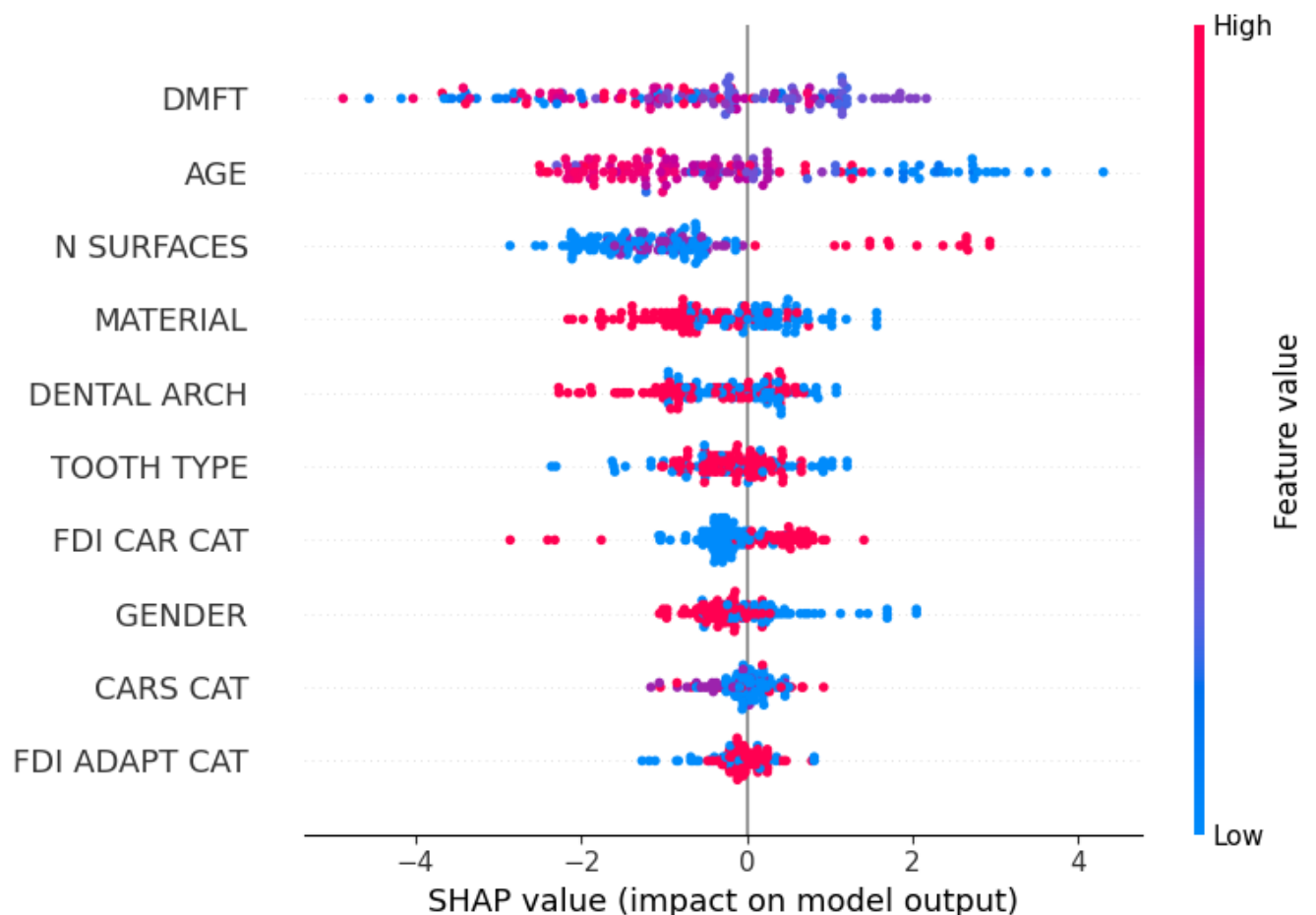
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0

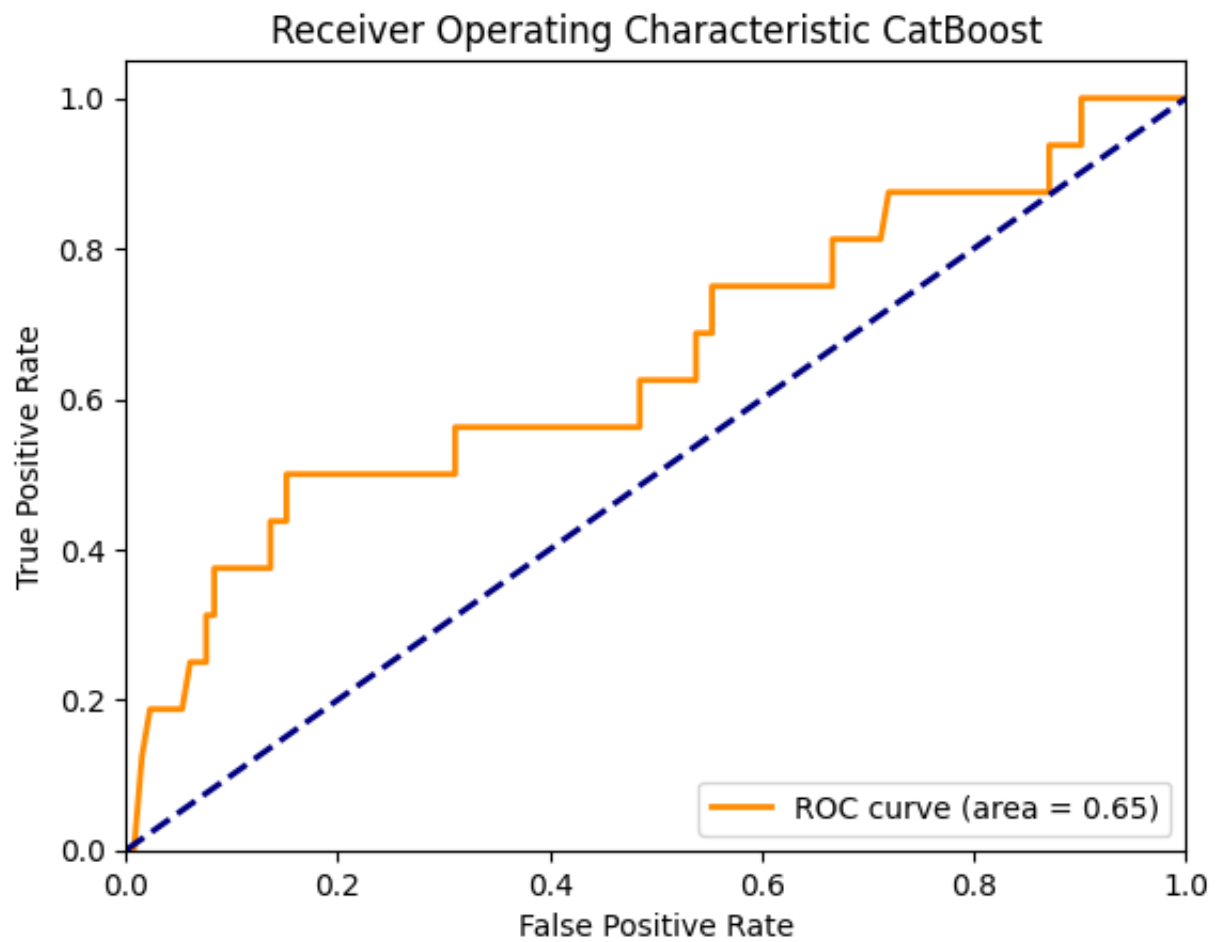
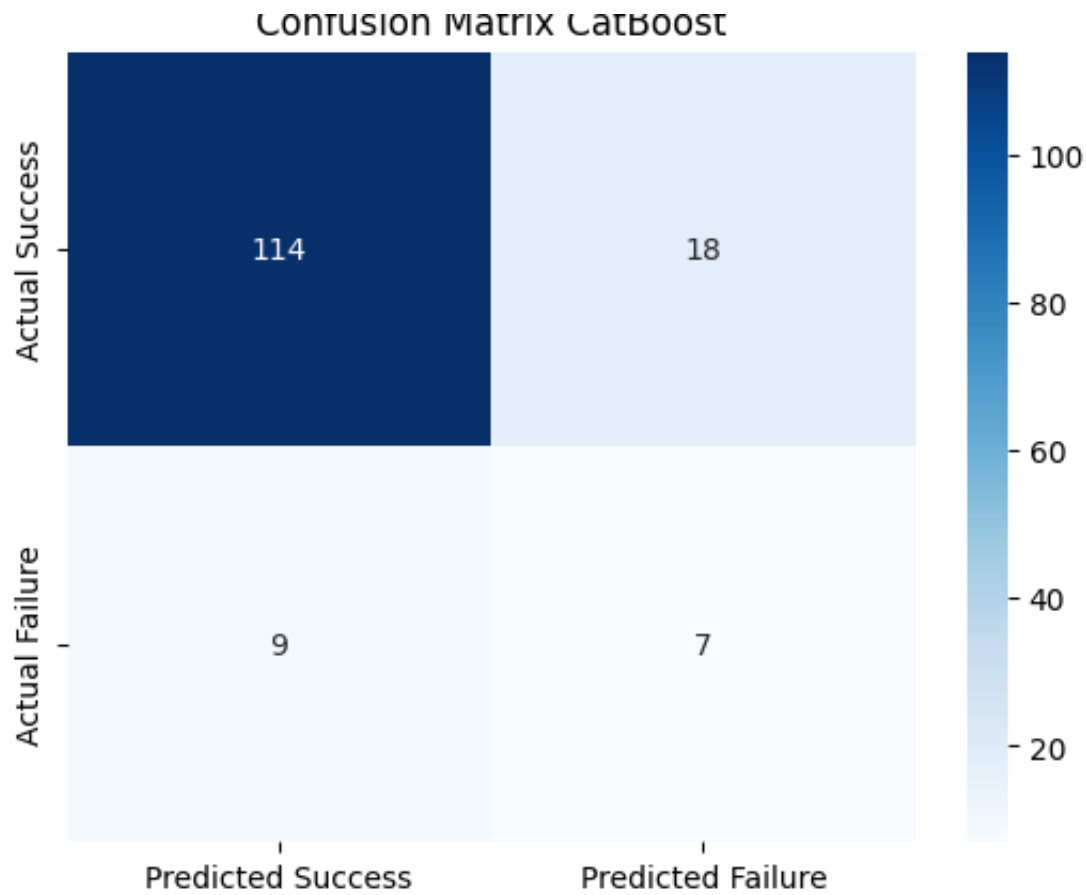
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

SHAP Summary for CatBoost





ROC Curve Metrics:

FPR: [0.0, 0.00757576, 0.01515152, 0.02272727, 0.03787879, 0.0530303, 0.06818182, 0.07575758, 0.07575758, 0.08333333, 0.08333333, 0.13636364]

```

0.00000000 0.01515151 0.01515151 0.08333333 0.08333333 0.15030304
0.13636364 0.15151515 0.15151515 0.17424242 0.1969697 0.21212121
0.25757576 0.28030303 0.31060606 0.31060606 0.34090909 0.35606061
0.43181818 0.4469697 0.48484848 0.48484848 0.5 0.53787879
0.53787879 0.5530303 0.5530303 0.66666667 0.66666667 0.68181818
0.6969697 0.71212121 0.71969697 0.81818182 0.84848485 0.87121212
0.87121212 0.90151515 0.90151515 1. ]
TPR: [0. 0. 0.125 0.1875 0.1875 0.1875 0.25 0.25 0.3125 0.3125
0.375 0.375 0.4375 0.4375 0.5 0.5 0.5 0.5 0.5 0.5
0.5 0.5625 0.5625 0.5625 0.5625 0.5625 0.5625 0.625 0.625 0.625
0.6875 0.6875 0.75 0.75 0.8125 0.8125 0.8125 0.8125 0.875 0.875
0.875 0.875 0.9375 0.9375 1. 1. ]
ROC AUC: 0.651

```

Running evaluation with seed 43
Inside evaluate_catboost function

Evaluating CatBoost with seed 43...

Best parameters for CatBoost: {'border_count': 128, 'depth': 6, 'iterations

--- ROC Data ---

```

FPR = [0.0, 0.007575757575757576, 0.015151515151515152, 0.02272727272727272
TPR = [0.0, 0.0, 0.125, 0.1875, 0.1875, 0.25, 0.25, 0.25, 0.25, 0.3125, 0.3
AUC = 0.6323390151515152

```

--- End ROC Data ---

Training - Accuracy: 0.995, Sensitivity: 0.997, Specificity: 0.993, F1: 0.9

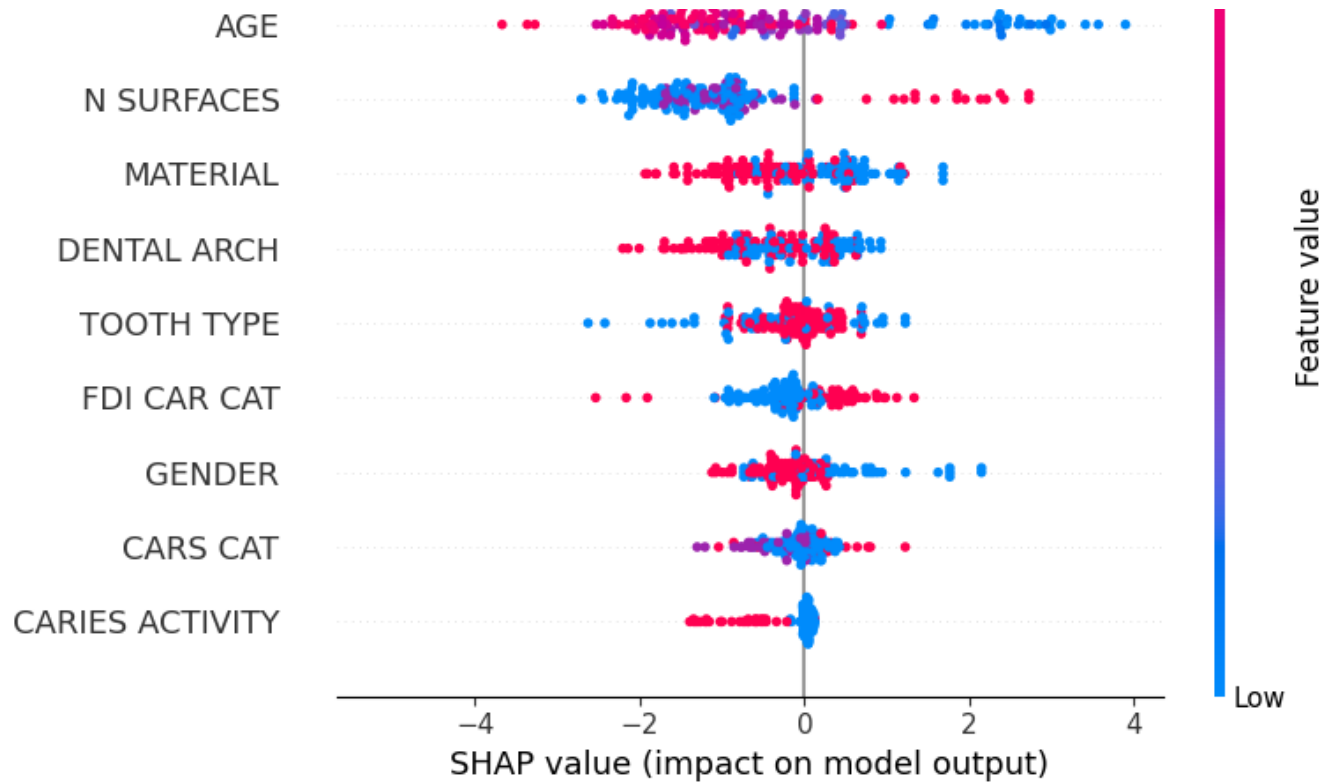
Test Metrics for manual threshold 0.5:

```

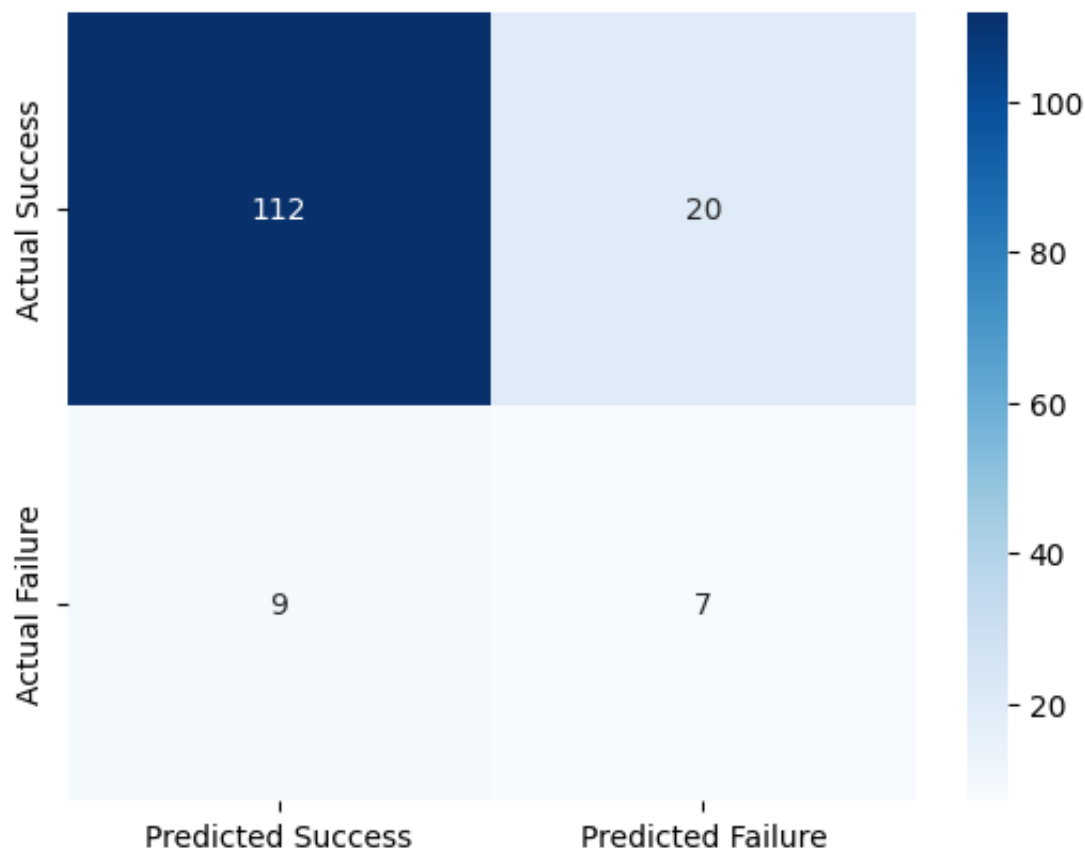
Accuracy: 0.804, Sensitivity: 0.438, Specificity: 0.848, F1: 0.326, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.5405405405405406, 'Sensitivity': 0
Threshold: 0.15, Metrics: {'Accuracy': 0.6283783783783784, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.6756756756756757, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.7027027027027027, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7027027027027027, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.7162162162162162, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7635135135135135, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7905405405405406, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.8040540540540541, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.60, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8513513513513513, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for CatBoost

```

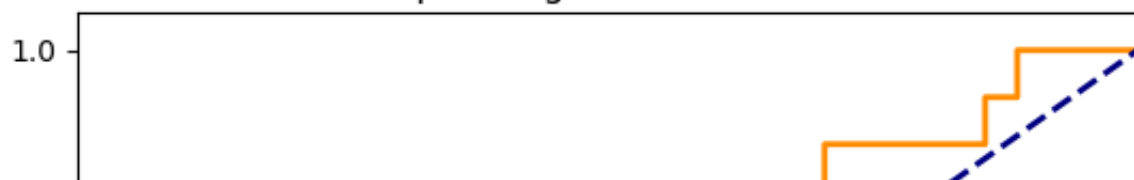


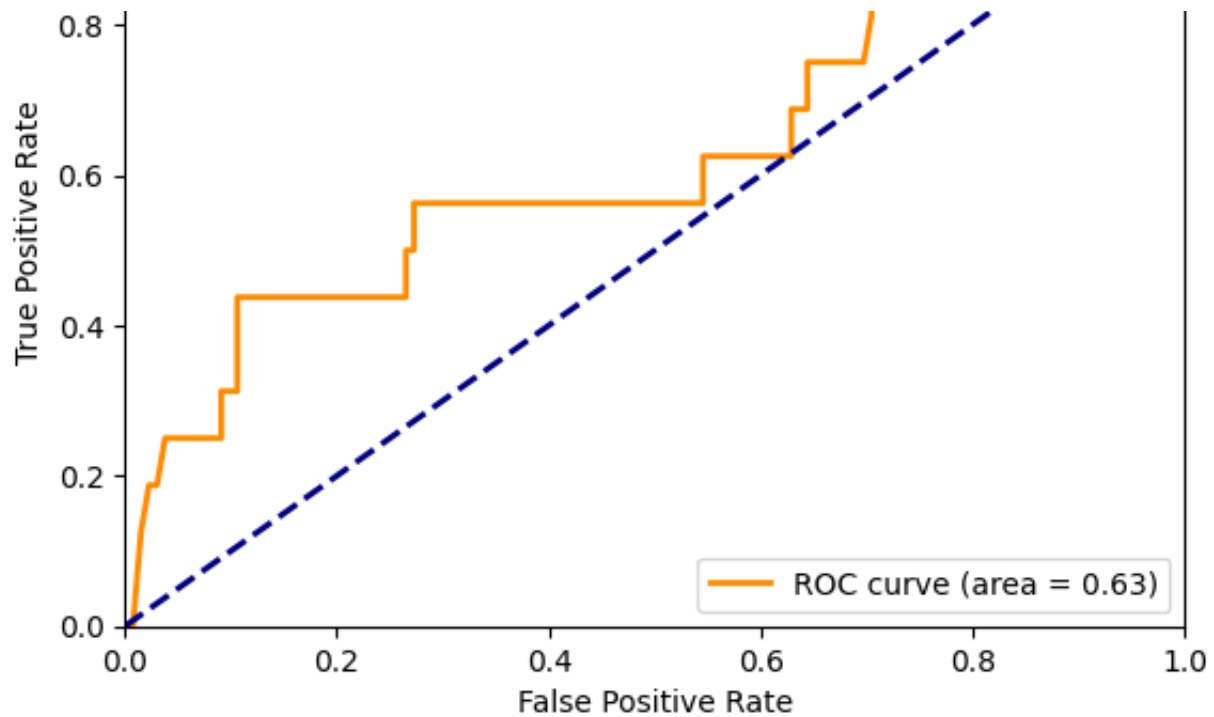


Confusion Matrix CatBoost



Receiver Operating Characteristic CatBoost





ROC Curve Metrics:

```
FPR: [0.          0.00757576 0.01515152 0.02272727 0.03030303 0.03787879
0.06060606 0.07575758 0.09090909 0.09090909 0.10606061 0.10606061
0.12878788 0.15151515 0.18181818 0.1969697  0.26515152 0.26515152
0.27272727 0.27272727 0.28787879 0.31060606 0.34848485 0.36363636
0.39393939 0.40909091 0.53030303 0.54545455 0.54545455 0.61363636
0.62878788 0.62878788 0.64393939 0.64393939 0.6969697  0.70454545
0.70454545 0.76515152 0.78030303 0.81818182 0.83333333 0.85606061
0.85606061 0.88636364 0.88636364 1.          ]
TPR: [0.          0.          0.125  0.1875 0.1875 0.25    0.25    0.25    0.25    0.3125
0.3125 0.4375 0.4375 0.4375 0.4375 0.4375 0.4375 0.5     0.5     0.5625
0.5625 0.5625 0.5625 0.5625 0.5625 0.5625 0.5625 0.5625 0.625  0.625
0.625  0.6875 0.6875 0.75   0.75   0.8125 0.875  0.875  0.875  0.875
0.875  0.875  0.9375 0.9375 1.     1.     ]
ROC AUC: 0.632
```

Running evaluation with seed 44

Inside evaluate_catboost function

Evaluating CatBoost with seed 44...

Best parameters for CatBoost: {'border_count': 128, 'depth': 6, 'iterations

--- ROC Data ---

FPR = [0.0, 0.007575757575757576, 0.015151515151515152, 0.022727272727272722

TPR = [0.0, 0.0, 0.125, 0.125, 0.1875, 0.1875, 0.1875, 0.25, 0.25, 0.3125,

AUC = 0.6541193181818182

--- End ROC Data ---

Training - Accuracy: 0.995, Sensitivity: 1.000, Specificity: 0.990, F1: 0.9

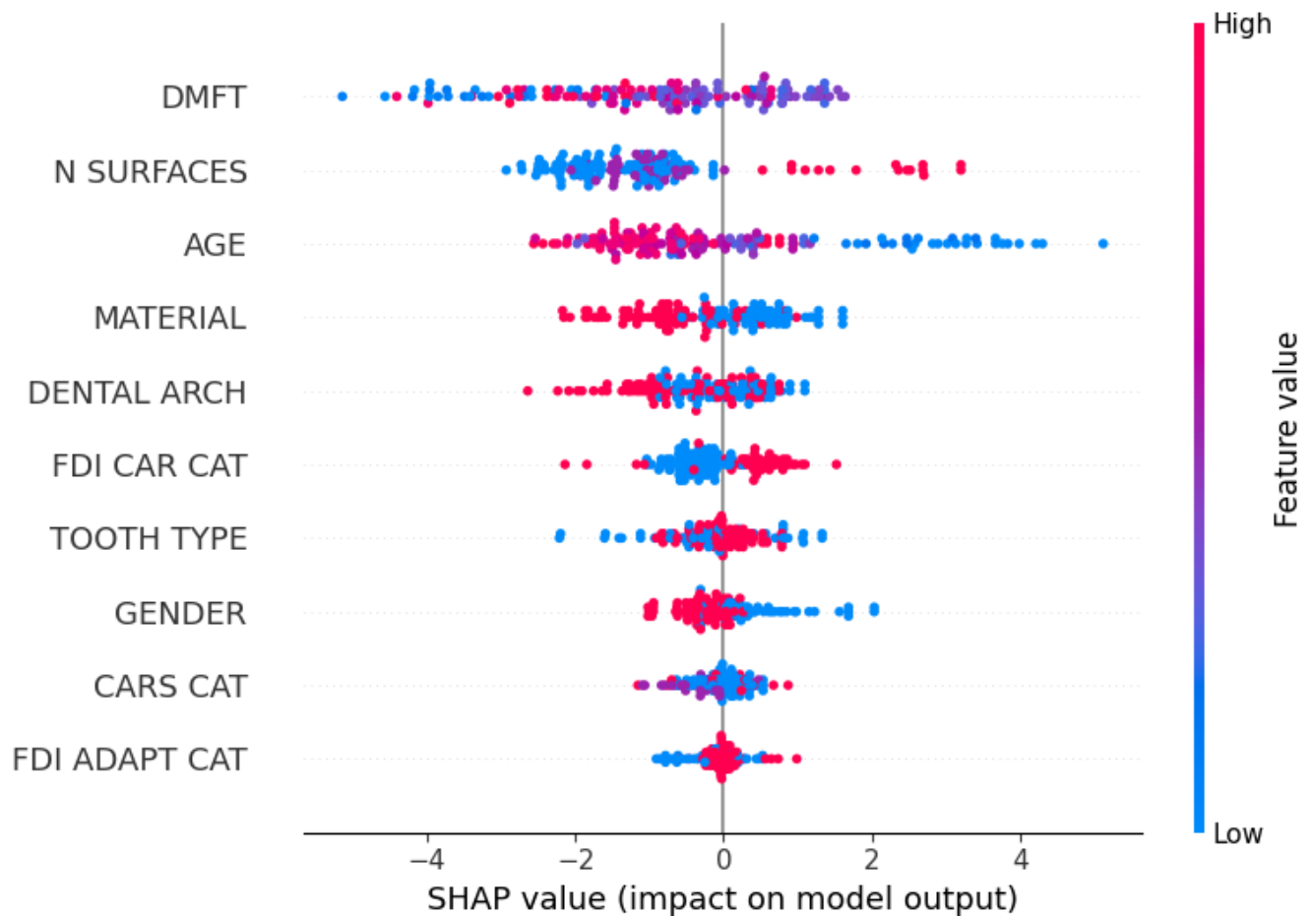
Test Metrics for manual threshold 0.5:

Accuracy: 0.818, Sensitivity: 0.438, Specificity: 0.864, F1: 0.341, ROC AUC

Threshold: 0.10, Metrics: {'Accuracy': 0.6013513513513513, 'Sensitivity': 0

Threshold: 0.15, Metrics: {'Accuracy': 0.6554054054054054, 'Sensitivity': 0

```
Threshold: 0.20, Metrics: {'Accuracy': 0.7027027027027027, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.7297297297297297, 'Sensitivity': 0
Threshold: 0.30, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0
Threshold: 0.35, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.777027027027027, 'Sensitivity': 0.
Threshold: 0.45, Metrics: {'Accuracy': 0.8108108108108109, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.70, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8513513513513513, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for CatBoost
```



Confusion Matrix CatBoost






```

0.5625 0.5625 0.5625 0.5625 0.5625 0.625 0.625 0.625 0.6875 0.6875
0.75 0.75 0.75 0.75 0.8125 0.8125 0.8125 0.875 0.875 0.875
0.875 0.9375 0.9375 1. 1. ]
ROC AUC: 0.654

```

Running evaluation with seed 45
Inside evaluate_catboost function

Evaluating CatBoost with seed 45...

Best parameters for CatBoost: {'border_count': 128, 'depth': 6, 'iterations

--- ROC Data ---

FPR = [0.0, 0.007575757575757576, 0.015151515151515152, 0.02272727272727272

TPR = [0.0, 0.0, 0.125, 0.125, 0.1875, 0.1875, 0.25, 0.25, 0.25, 0.3125, 0.

AUC = 0.6560132575757576

--- End ROC Data ---

Training - Accuracy: 0.995, Sensitivity: 0.997, Specificity: 0.993, F1: 0.9

Test Metrics for manual threshold 0.5:

Accuracy: 0.804, Sensitivity: 0.375, Specificity: 0.856, F1: 0.293, ROC AUC

Threshold: 0.10, Metrics: {'Accuracy': 0.5945945945945946, 'Sensitivity': 0

Threshold: 0.15, Metrics: {'Accuracy': 0.6283783783783784, 'Sensitivity': 0

Threshold: 0.20, Metrics: {'Accuracy': 0.6554054054054054, 'Sensitivity': 0

Threshold: 0.25, Metrics: {'Accuracy': 0.7162162162162162, 'Sensitivity': 0

Threshold: 0.30, Metrics: {'Accuracy': 0.7364864864864865, 'Sensitivity': 0

Threshold: 0.35, Metrics: {'Accuracy': 0.7432432432432432, 'Sensitivity': 0

Threshold: 0.40, Metrics: {'Accuracy': 0.7837837837837838, 'Sensitivity': 0

Threshold: 0.45, Metrics: {'Accuracy': 0.7905405405405406, 'Sensitivity': 0

Threshold: 0.50, Metrics: {'Accuracy': 0.8040540540540541, 'Sensitivity': 0

Threshold: 0.55, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0

Threshold: 0.60, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0

Threshold: 0.65, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.

Threshold: 0.70, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0

Threshold: 0.75, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0

Threshold: 0.80, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0

Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0

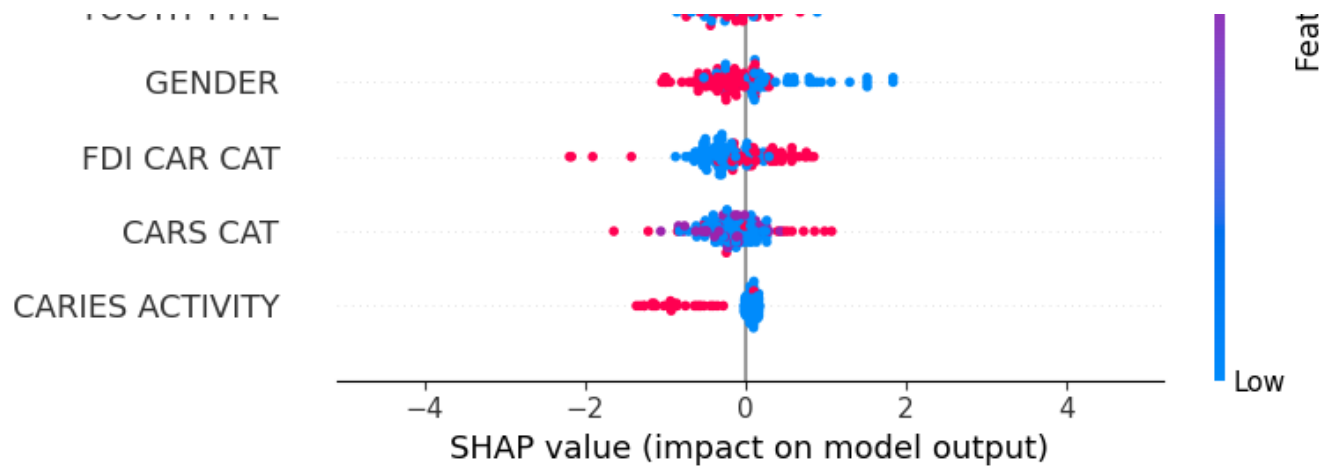
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

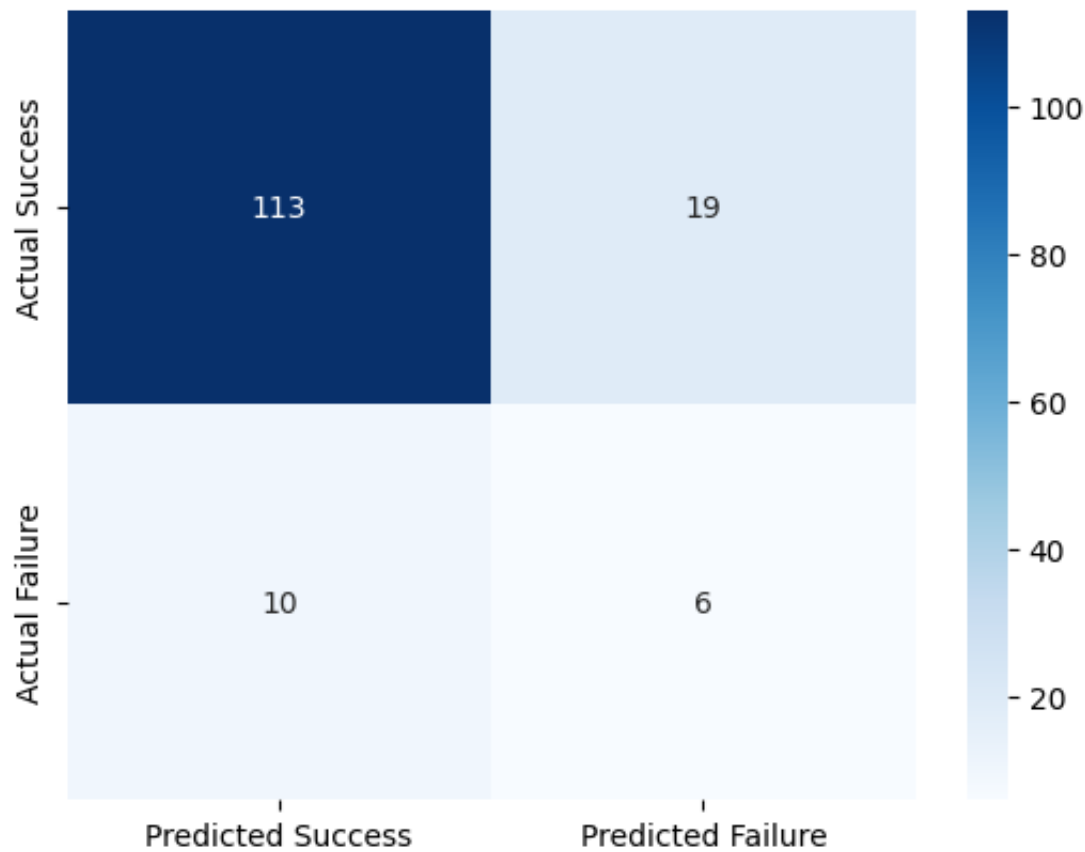
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

SHAP Summary for CatBoost

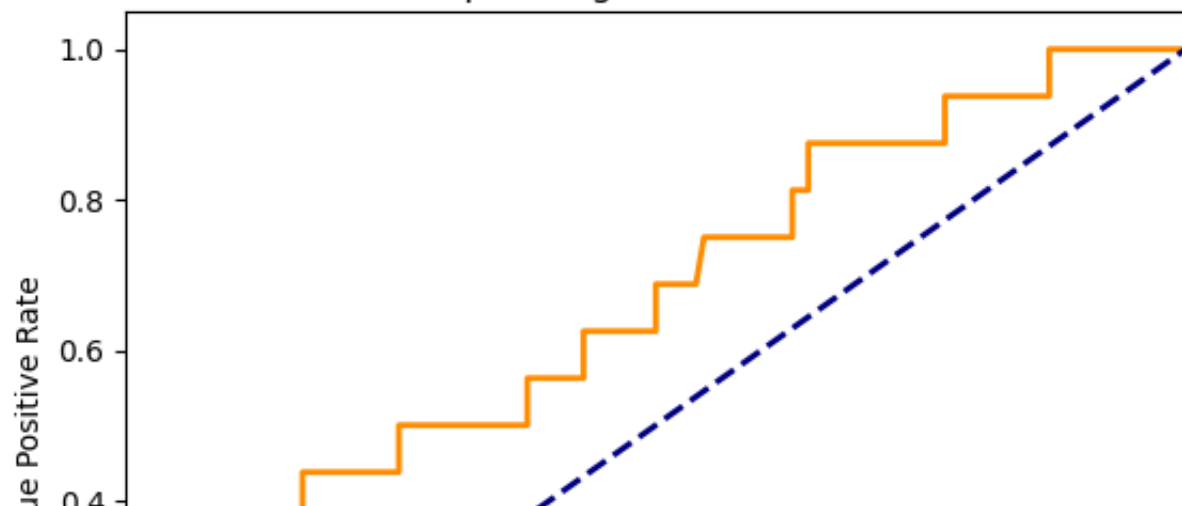


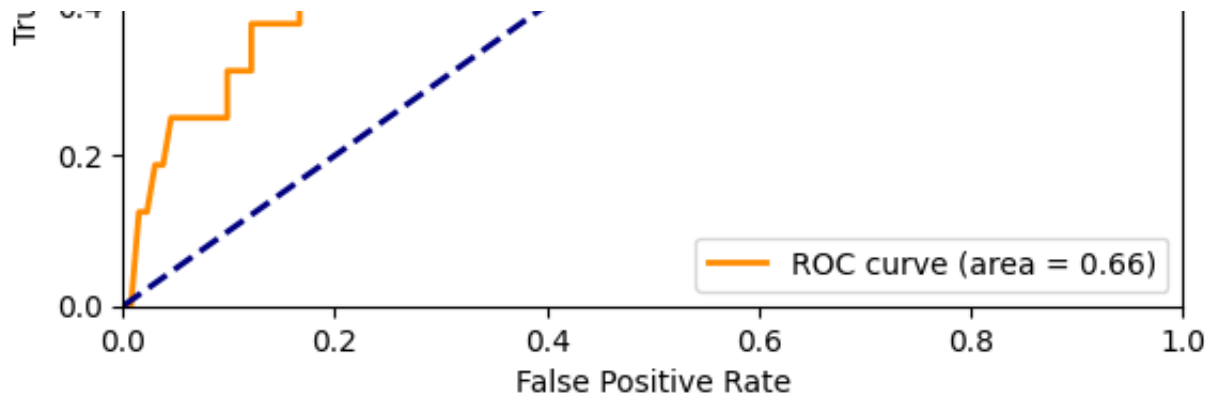


Confusion Matrix CatBoost



Receiver Operating Characteristic CatBoost





ROC Curve Metrics:

```
FPR: [0.          0.00757576 0.01515152 0.02272727 0.03030303 0.03787879
 0.04545455 0.06060606 0.09848485 0.09848485 0.12121212 0.12121212
 0.16666667 0.16666667 0.18181818 0.1969697  0.21969697 0.25757576
 0.25757576 0.26515152 0.28787879 0.29545455 0.31060606 0.37878788
 0.37878788 0.43181818 0.43181818 0.5          0.5          0.51515152
 0.53030303 0.53787879 0.54545455 0.56060606 0.57575758 0.61363636
 0.62878788 0.62878788 0.64393939 0.64393939 0.71212121 0.72727273
 0.77272727 0.77272727 0.83333333 0.84848485 0.87121212 0.87121212
 1.          ]
TPR: [0.          0.          0.125  0.125  0.1875 0.1875 0.25    0.25    0.25    0.3125
 0.3125 0.375   0.375   0.4375 0.4375 0.4375 0.4375 0.4375 0.5     0.5
 0.5    0.5    0.5     0.5    0.5625 0.5625 0.625   0.625   0.6875 0.6875
 0.6875 0.6875 0.75    0.75    0.75    0.75    0.75    0.8125 0.8125 0.875
 0.875  0.875  0.875   0.9375 0.9375 0.9375 0.9375 1.      1.      ]
ROC AUC: 0.656
```

Running evaluation with seed 46

Inside evaluate_catboost function

Evaluating CatBoost with seed 46...

Best parameters for CatBoost: {'border_count': 128, 'depth': 6, 'iterations

--- ROC Data ---

FPR = [0.0, 0.007575757575757576, 0.015151515151515152, 0.02272727272727272

TPR = [0.0, 0.0, 0.125, 0.125, 0.1875, 0.1875, 0.1875, 0.25, 0.25, 0.375, 0

AUC = 0.6569602272727273

--- End ROC Data ---

Training - Accuracy: 0.995, Sensitivity: 0.997, Specificity: 0.993, F1: 0.9

Test Metrics for manual threshold 0.5:

Accuracy: 0.811, Sensitivity: 0.438, Specificity: 0.856, F1: 0.333, ROC AUC

Threshold: 0.10, Metrics: {'Accuracy': 0.5540540540540541, 'Sensitivity': 0

Threshold: 0.15, Metrics: {'Accuracy': 0.6216216216216216, 'Sensitivity': 0

Threshold: 0.20, Metrics: {'Accuracy': 0.6621621621621622, 'Sensitivity': 0

Threshold: 0.25, Metrics: {'Accuracy': 0.7162162162162162, 'Sensitivity': 0

Threshold: 0.30, Metrics: {'Accuracy': 0.7297297297297297, 'Sensitivity': 0

Threshold: 0.35, Metrics: {'Accuracy': 0.75, 'Sensitivity': 0.5, 'Specifici

Threshold: 0.40, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0

Threshold: 0.45, Metrics: {'Accuracy': 0.7972972972972973, 'Sensitivity': 0

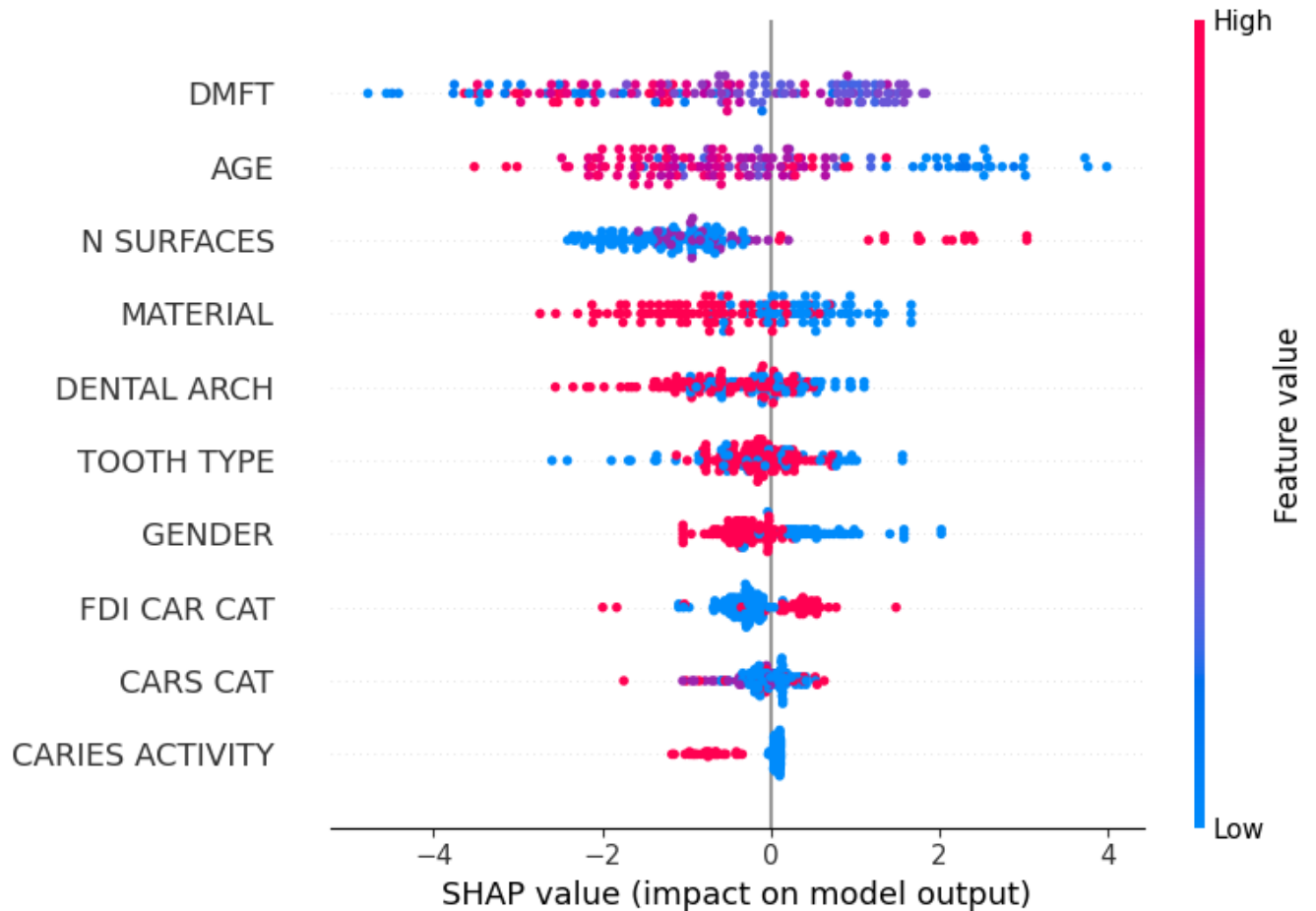
Threshold: 0.50, Metrics: {'Accuracy': 0.8108108108108109, 'Sensitivity': 0

Threshold: 0.55. Metrics: {'Accuracv': 0.831081081081081. 'Sensitivitv': 0.

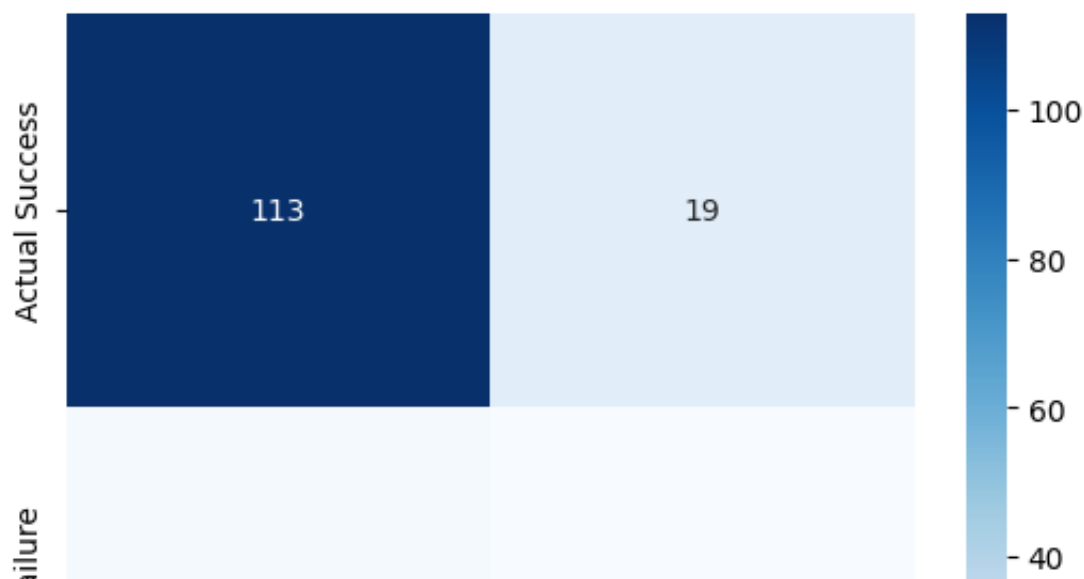
```

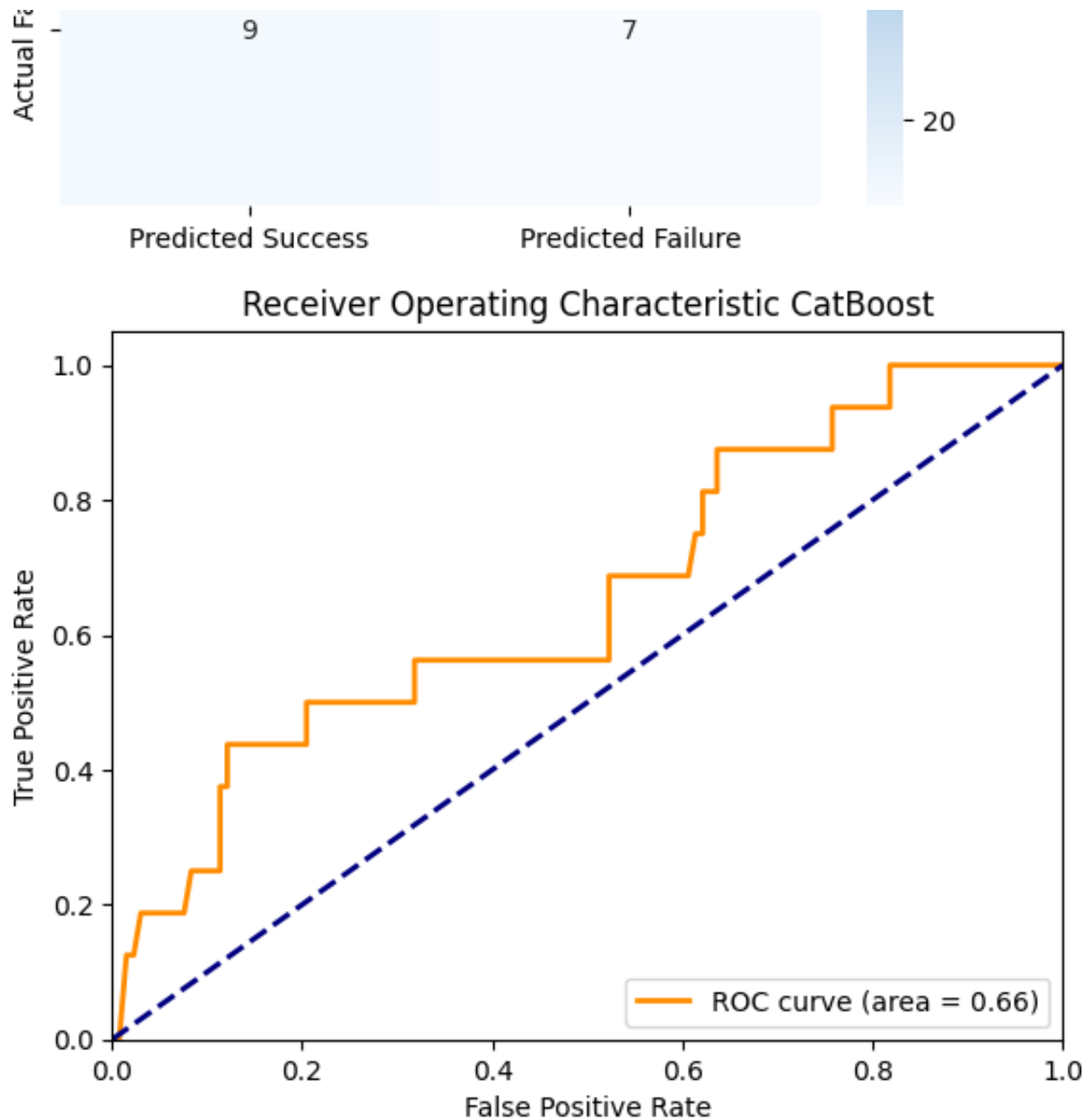
Threshold: 0.60, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
SHAP Summary for CatBoost

```



Confusion Matrix CatBoost





ROC Curve Metrics:

```
FPR: [0.          0.00757576 0.01515152 0.02272727 0.03030303 0.06060606
0.07575758 0.08333333 0.11363636 0.11363636 0.12121212 0.12121212
0.14393939 0.15909091 0.17424242 0.20454545 0.20454545 0.25757576
0.27272727 0.28030303 0.3030303 0.31818182 0.31818182 0.4469697
0.46212121 0.52272727 0.52272727 0.56818182 0.58333333 0.60606061
0.61363636 0.62121212 0.62121212 0.63636364 0.63636364 0.6969697
0.71212121 0.75757576 0.75757576 0.76515152 0.78030303 0.81818182
0.81818182 1.          ]
```

```
TPR: [0.          0.          0.125 0.125 0.1875 0.1875 0.1875 0.25 0.25 0.375
0.375 0.4375 0.4375 0.4375 0.4375 0.4375 0.5 0.5 0.5 0.5
0.5 0.5 0.5625 0.5625 0.5625 0.5625 0.6875 0.6875 0.6875 0.6875
0.75 0.75 0.8125 0.8125 0.875 0.875 0.875 0.875 0.9375 0.9375
0.9375 0.9375 1.          1.          ]
```

ROC AUC: 0.657

Running evaluation with seed 47

Inside evaluate_catboost function

Evaluating CatBoost with seed 47...

Best parameters for CatBoost: {'border_count': 128, 'depth': 6, 'iterations

--- ROC Data ---

FPR = [0.0, 0.007575757575757576, 0.015151515151515152, 0.02272727272727272

TPR = [0.0, 0.0, 0.0625, 0.1875, 0.1875, 0.25, 0.25, 0.25, 0.25, 0.375, 0.3

AUC = 0.6735321969696969

--- End ROC Data ---

Training - Accuracy: 0.995, Sensitivity: 0.997, Specificity: 0.993, F1: 0.9

Test Metrics for manual threshold 0.5:

Accuracy: 0.797, Sensitivity: 0.438, Specificity: 0.841, F1: 0.318, ROC AUC

Threshold: 0.10, Metrics: {'Accuracy': 0.6283783783783784, 'Sensitivity': 0

Threshold: 0.15, Metrics: {'Accuracy': 0.6486486486486487, 'Sensitivity': 0

Threshold: 0.20, Metrics: {'Accuracy': 0.668918918918919, 'Sensitivity': 0.

Threshold: 0.25, Metrics: {'Accuracy': 0.7162162162162162, 'Sensitivity': 0

Threshold: 0.30, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0

Threshold: 0.35, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0

Threshold: 0.40, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0

Threshold: 0.45, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0

Threshold: 0.50, Metrics: {'Accuracy': 0.7972972972972973, 'Sensitivity': 0

Threshold: 0.55, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.

Threshold: 0.60, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.

Threshold: 0.65, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0

Threshold: 0.70, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0

Threshold: 0.75, Metrics: {'Accuracy': 0.8648648648648649, 'Sensitivity': 0

Threshold: 0.80, Metrics: {'Accuracy': 0.8716216216216216, 'Sensitivity': 0

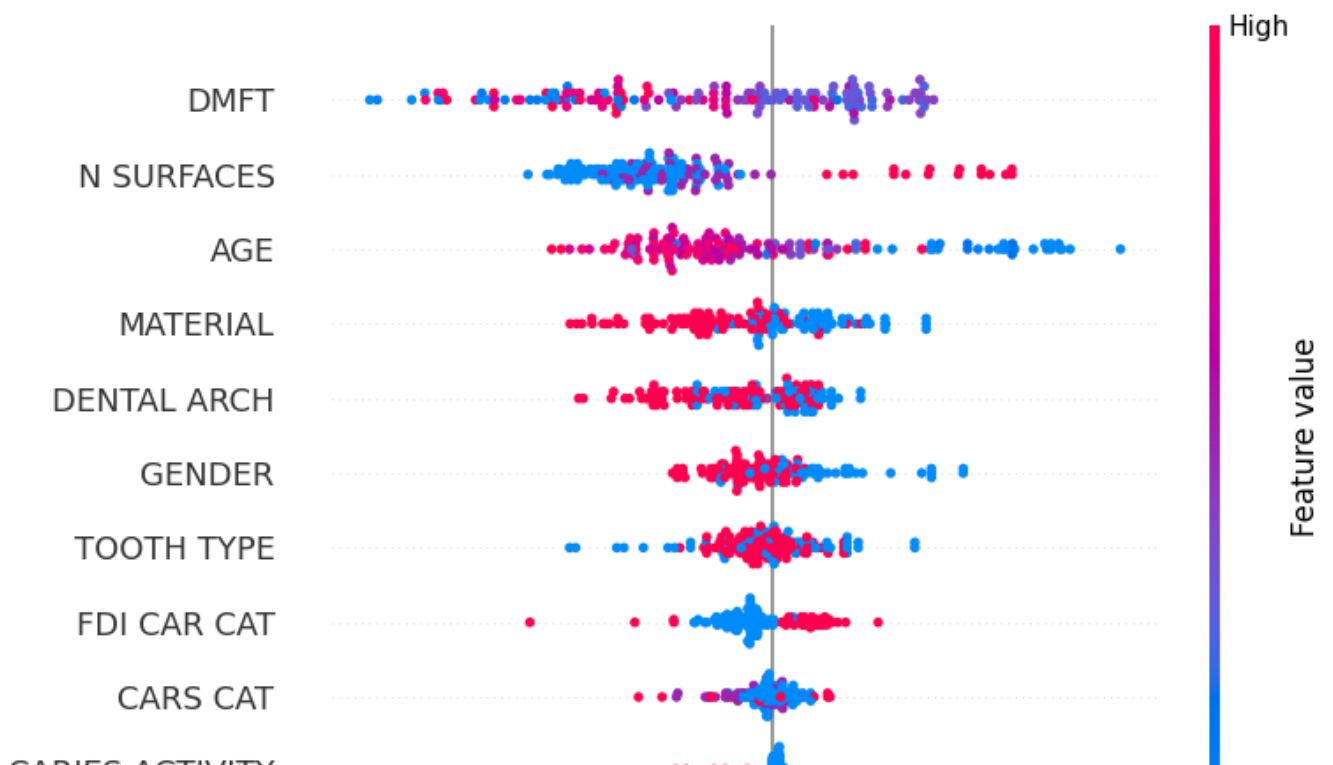
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0

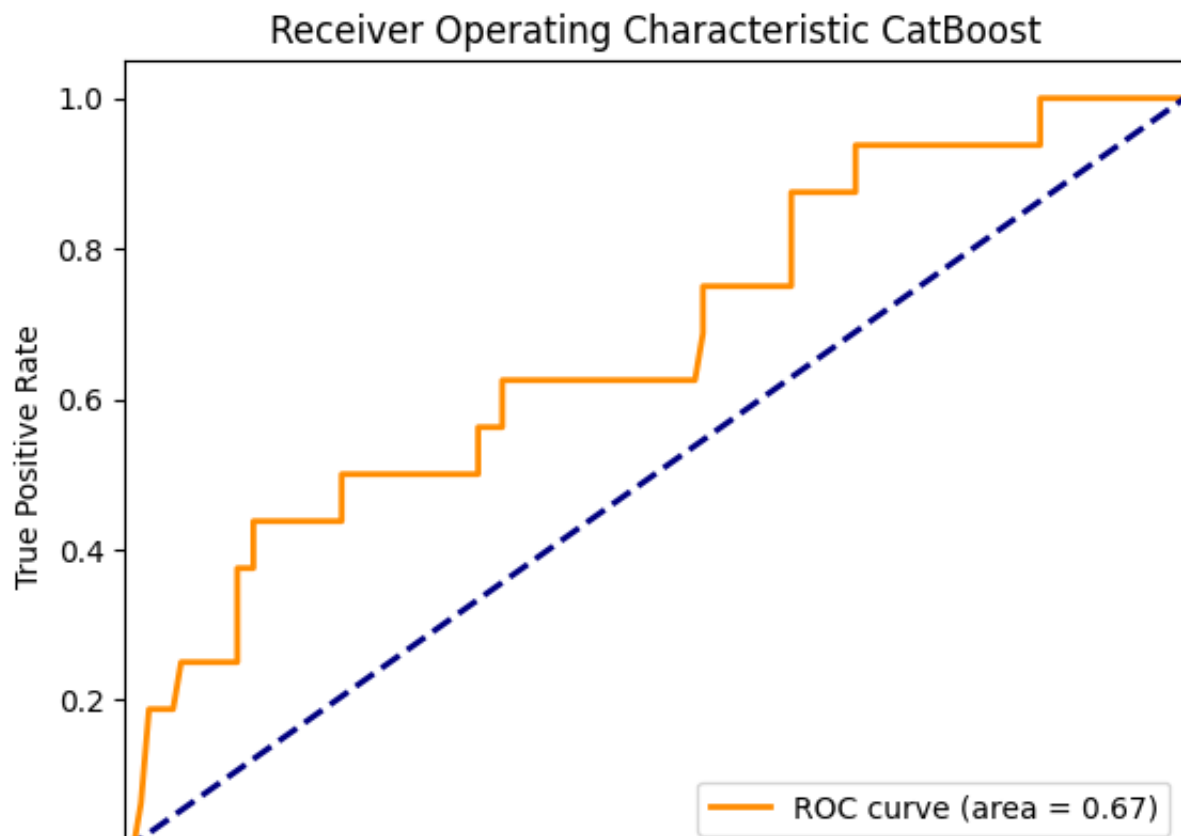
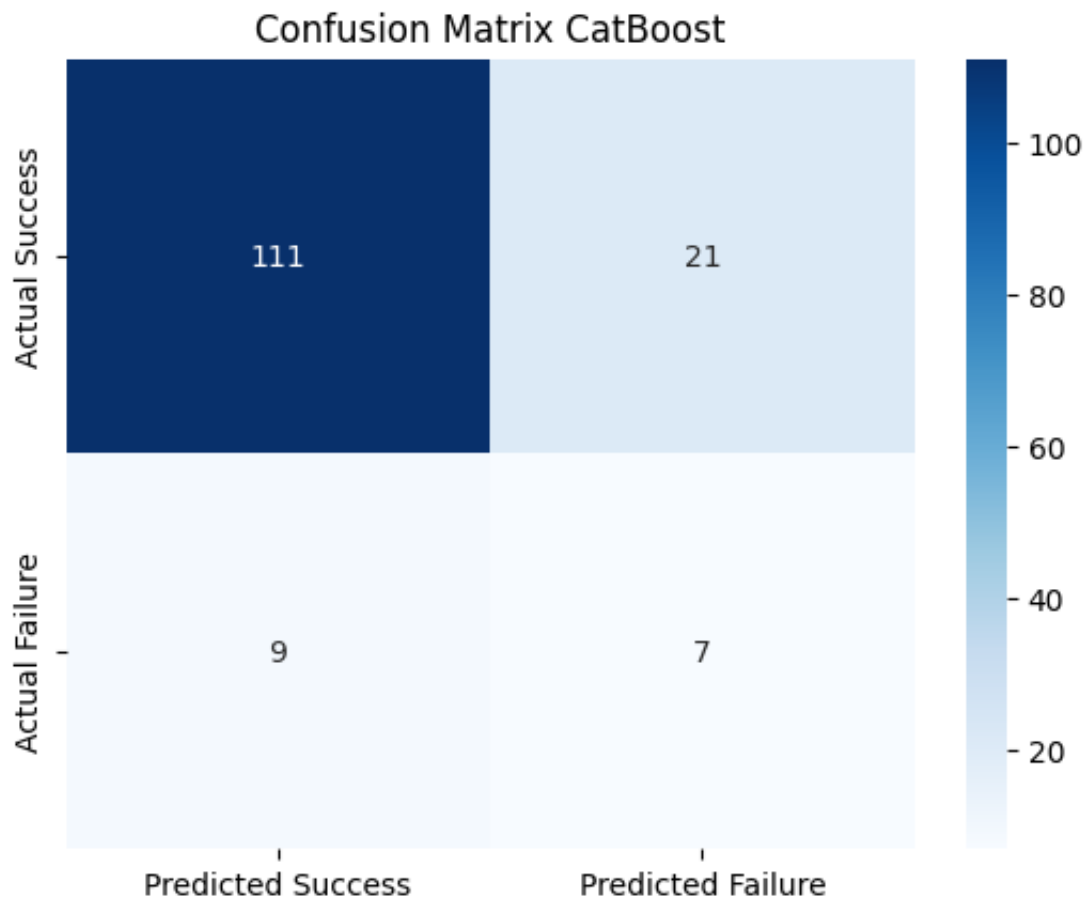
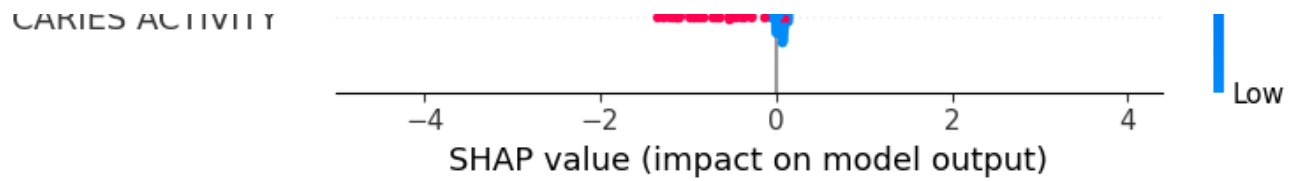
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

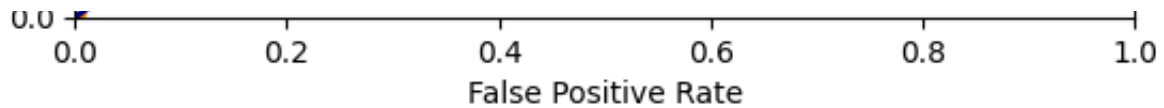
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

SHAP Summary for CatBoost







ROC Curve Metrics:

```
FPR: [0.          0.00757576 0.01515152 0.02272727 0.04545455 0.0530303
 0.06060606 0.07575758 0.10606061 0.10606061 0.12121212 0.12121212
 0.12878788 0.14393939 0.16666667 0.18939394 0.20454545 0.20454545
 0.27272727 0.29545455 0.33333333 0.33333333 0.35606061 0.35606061
 0.40909091 0.42424242 0.52272727 0.53787879 0.54545455 0.54545455
 0.56818182 0.58333333 0.62878788 0.62878788 0.66666667 0.68181818
 0.68939394 0.68939394 0.81060606 0.82575758 0.83333333 0.84848485
 0.86363636 0.86363636 1.          ]
TPR: [0.          0.          0.0625 0.1875 0.1875 0.25      0.25      0.25      0.25      0.25      0.375
 0.375  0.4375 0.4375 0.4375 0.4375 0.4375 0.4375 0.4375 0.5        0.5        0.5
 0.5      0.5625 0.5625 0.625   0.625   0.625   0.625   0.625   0.625   0.6875 0.75
 0.75     0.75   0.75   0.875   0.875   0.875   0.875   0.875   0.9375 0.9375 0.9375
 0.9375 0.9375 0.9375 1.        1.        ]
ROC AUC: 0.674
```

Running evaluation with seed 48

Inside evaluate_catboost function

Evaluating CatBoost with seed 48...

Best parameters for CatBoost: {'border_count': 128, 'depth': 6, 'iterations

--- ROC Data ---

```
FPR = [0.0, 0.007575757575757576, 0.015151515151515152, 0.02272727272727272
TPR = [0.0, 0.0, 0.125, 0.1875, 0.1875, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25,
AUC = 0.6318655303030303
```

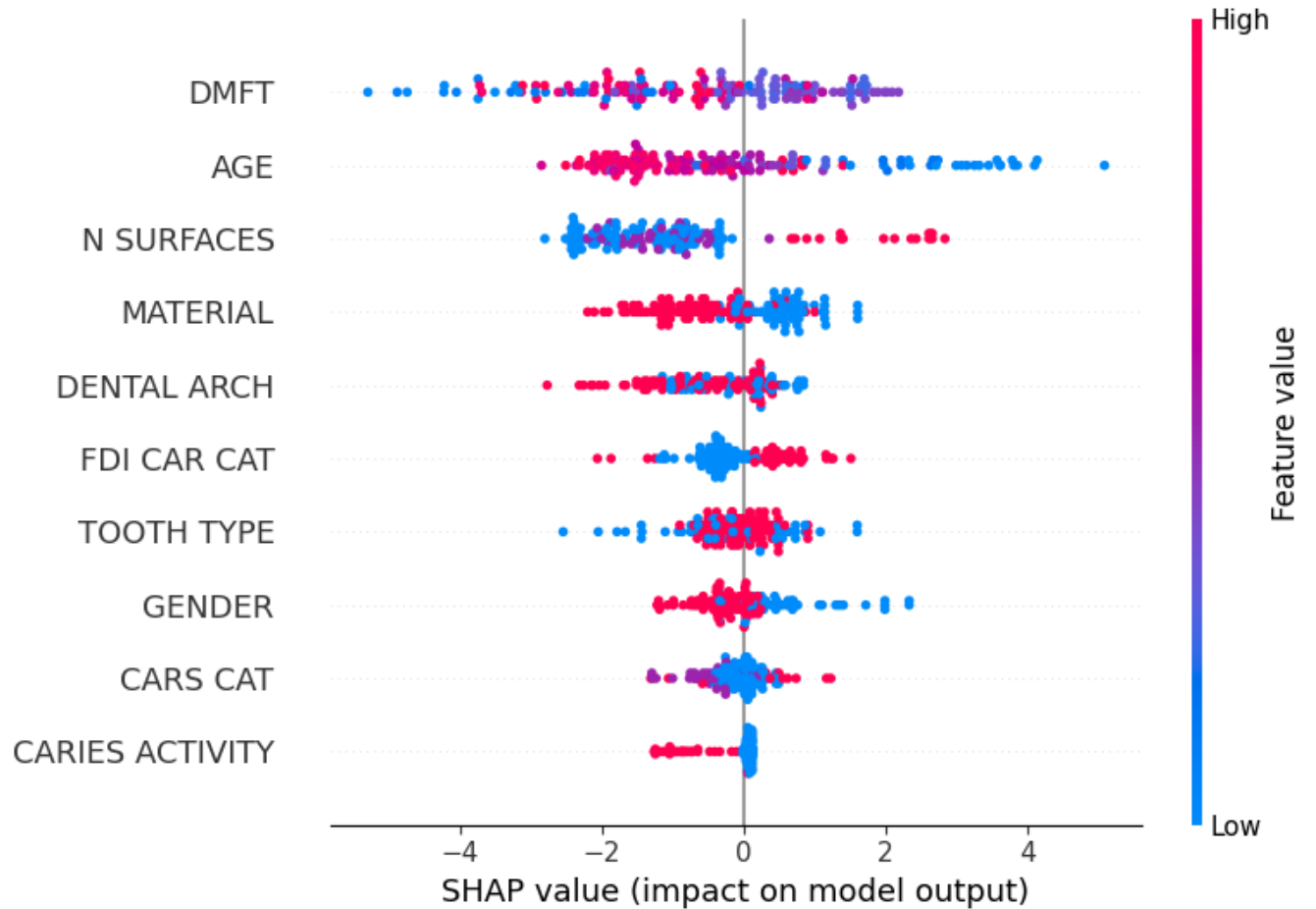
--- End ROC Data ---

Training - Accuracy: 0.995, Sensitivity: 0.997, Specificity: 0.993, F1: 0.9

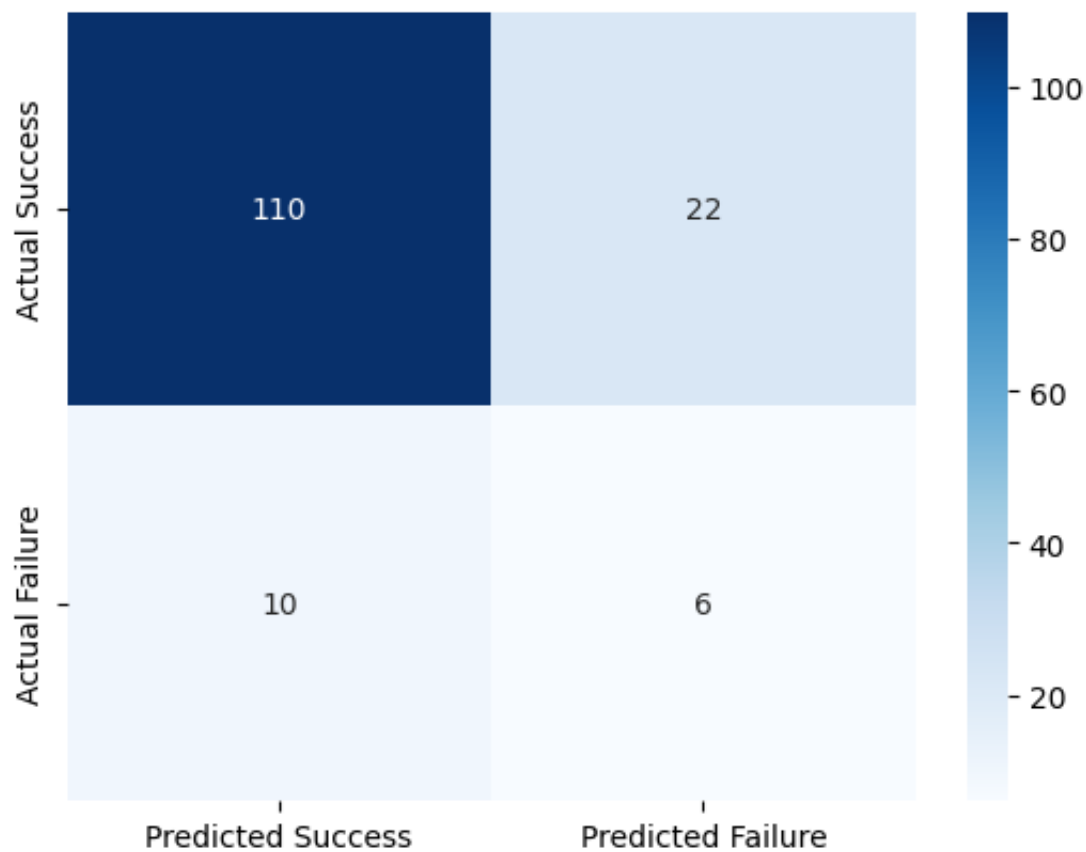
Test Metrics for manual threshold 0.5:

```
Accuracy: 0.784, Sensitivity: 0.375, Specificity: 0.833, F1: 0.273, ROC AUC
Threshold: 0.10, Metrics: {'Accuracy': 0.581081081081081, 'Sensitivity': 0.
Threshold: 0.15, Metrics: {'Accuracy': 0.6283783783783784, 'Sensitivity': 0
Threshold: 0.20, Metrics: {'Accuracy': 0.6891891891891891, 'Sensitivity': 0
Threshold: 0.25, Metrics: {'Accuracy': 0.722972972972973, 'Sensitivity': 0.
Threshold: 0.30, Metrics: {'Accuracy': 0.722972972972973, 'Sensitivity': 0.
Threshold: 0.35, Metrics: {'Accuracy': 0.7432432432432432, 'Sensitivity': 0
Threshold: 0.40, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0
Threshold: 0.45, Metrics: {'Accuracy': 0.7702702702702703, 'Sensitivity': 0
Threshold: 0.50, Metrics: {'Accuracy': 0.7837837837837838, 'Sensitivity': 0
Threshold: 0.55, Metrics: {'Accuracy': 0.7837837837837838, 'Sensitivity': 0
Threshold: 0.60, Metrics: {'Accuracy': 0.8040540540540541, 'Sensitivity': 0
Threshold: 0.65, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.
Threshold: 0.70, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0
Threshold: 0.75, Metrics: {'Accuracy': 0.8581081081081081, 'Sensitivity': 0
Threshold: 0.80, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0
Threshold: 0.85, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0
```

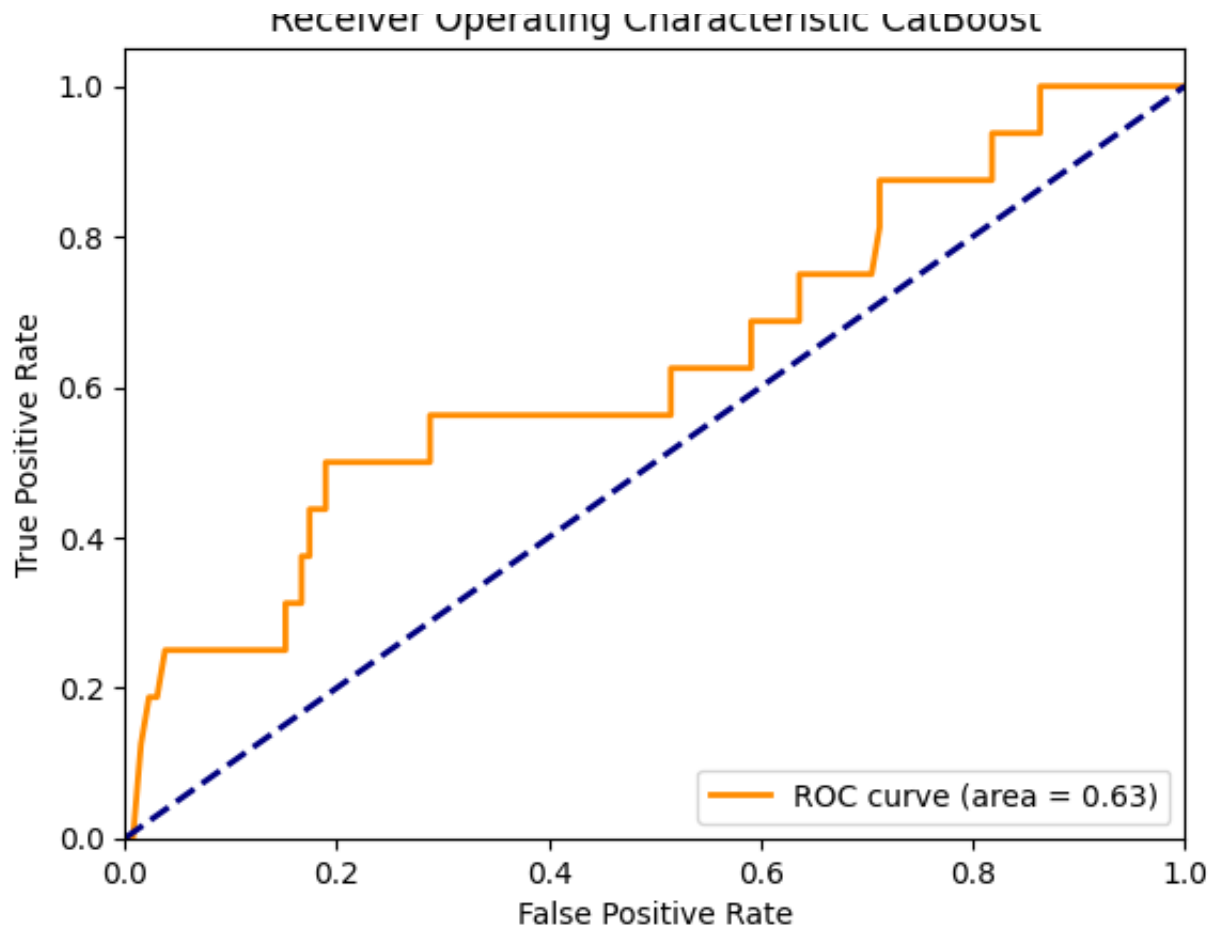
SHAP Summary for CatBoost



Confusion Matrix CatBoost



Receiver Operating Characteristic CatBoost



ROC Curve Metrics:

```
FPR: [0.          0.00757576 0.01515152 0.02272727 0.03030303 0.03787879
0.06818182 0.08333333 0.09848485 0.11363636 0.12878788 0.15151515
0.15151515 0.16666667 0.16666667 0.17424242 0.17424242 0.18939394
0.18939394 0.26515152 0.28787879 0.28787879 0.34090909 0.35606061
0.49242424 0.50757576 0.51515152 0.51515152 0.52272727 0.53787879
0.59090909 0.59090909 0.63636364 0.63636364 0.68939394 0.70454545
0.71212121 0.71212121 0.77272727 0.78787879 0.79545455 0.81060606
0.81818182 0.81818182 0.86363636 0.86363636 1.          ]
TPR: [0.          0.          0.125  0.1875 0.1875 0.25    0.25    0.25    0.25    0.25
0.25    0.25    0.3125 0.3125 0.375  0.375  0.4375 0.4375 0.5     0.5
0.5     0.5625 0.5625 0.5625 0.5625 0.5625 0.5625 0.625  0.625  0.625
0.625  0.6875 0.6875 0.75    0.75    0.75    0.8125 0.875  0.875  0.875
0.875  0.875  0.875  0.9375 0.9375 1.          1.          ]
```

ROC AUC: 0.632

Running evaluation with seed 49

Inside evaluate_catboost function

Evaluating CatBoost with seed 49...

Best parameters for CatBoost: {'border_count': 128, 'depth': 6, 'iterations

--- ROC Data ---

FPR = [0.0, 0.007575757575757576, 0.015151515151515152, 0.02272727272727272

TPR = [0.0, 0.0, 0.125, 0.125, 0.25, 0.25, 0.25, 0.25, 0.3125, 0.3125, 0.31

AUC = 0.6223958333333333

--- End ROC Data ---

Training - Accuracy: 0.995, Sensitivity: 0.997, Specificity: 0.993, F1: 0.9

Test Metrics for manual threshold 0.5:

Accuracy: 0.797, Sensitivity: 0.312, Specificity: 0.856, F1: 0.250, ROC AUC

Threshold: 0.10, Metrics: {'Accuracy': 0.5743243243243243, 'Sensitivity': 0

Threshold: 0.15, Metrics: {'Accuracy': 0.6216216216216216, 'Sensitivity': 0

Threshold: 0.20, Metrics: {'Accuracy': 0.6891891891891891, 'Sensitivity': 0

Threshold: 0.25, Metrics: {'Accuracy': 0.7162162162162162, 'Sensitivity': 0

Threshold: 0.30, Metrics: {'Accuracy': 0.7297297297297297, 'Sensitivity': 0

Threshold: 0.35, Metrics: {'Accuracy': 0.7297297297297297, 'Sensitivity': 0

Threshold: 0.40, Metrics: {'Accuracy': 0.7567567567567568, 'Sensitivity': 0

Threshold: 0.45, Metrics: {'Accuracy': 0.7972972972972973, 'Sensitivity': 0

Threshold: 0.50, Metrics: {'Accuracy': 0.7972972972972973, 'Sensitivity': 0

Threshold: 0.55, Metrics: {'Accuracy': 0.8175675675675675, 'Sensitivity': 0

Threshold: 0.60, Metrics: {'Accuracy': 0.831081081081081, 'Sensitivity': 0.

Threshold: 0.65, Metrics: {'Accuracy': 0.8378378378378378, 'Sensitivity': 0

Threshold: 0.70, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0

Threshold: 0.75, Metrics: {'Accuracy': 0.8445945945945946, 'Sensitivity': 0

Threshold: 0.80, Metrics: {'Accuracy': 0.8783783783783784, 'Sensitivity': 0

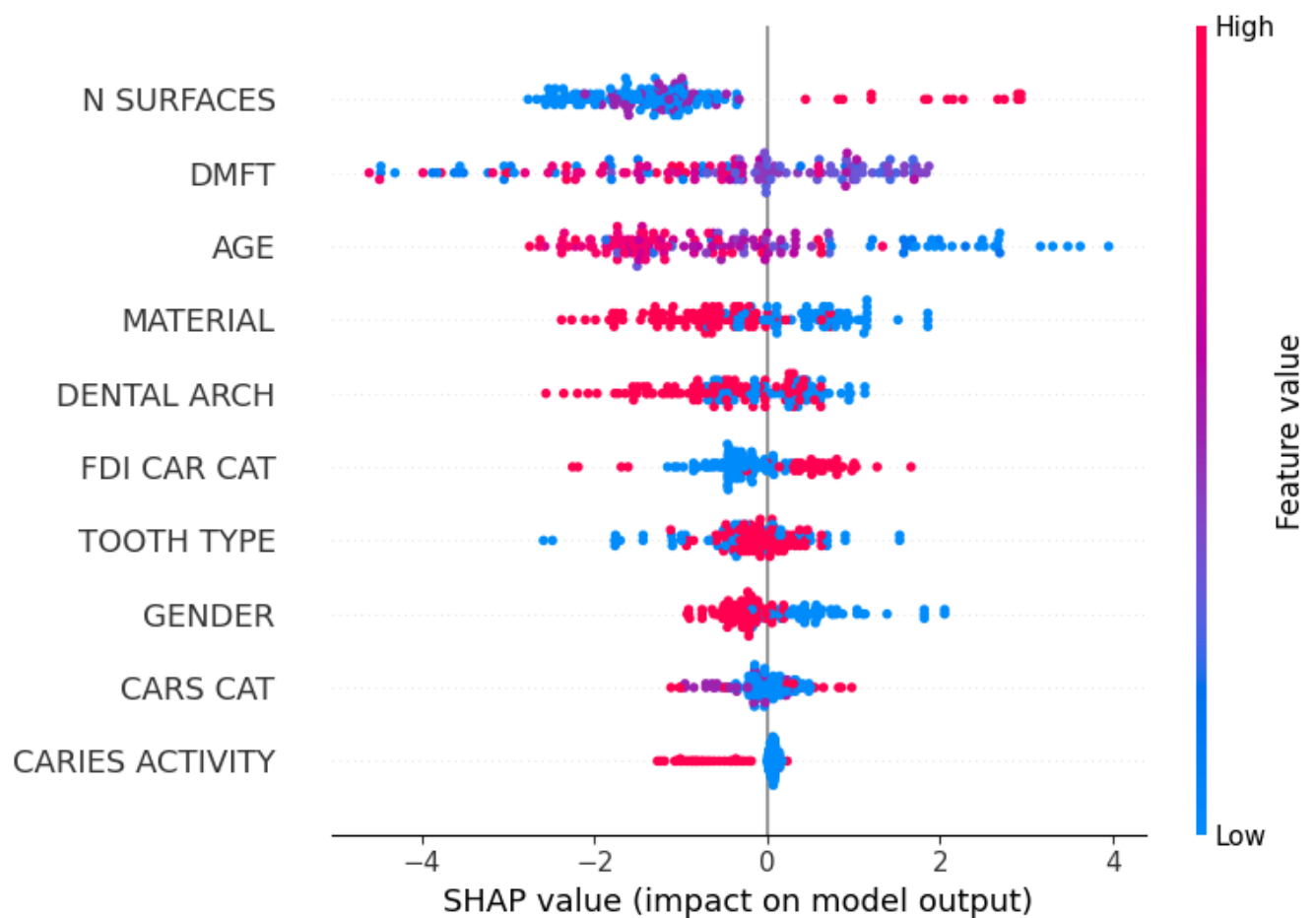
Threshold: 0.85, Metrics: {'Accuracy': 0.8851351351351351, 'Sensitivity': 0

Threshold: 0.90, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

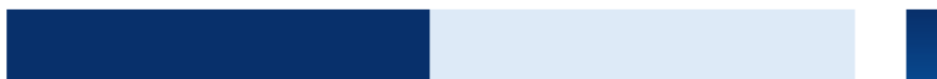
Threshold: 0.95, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

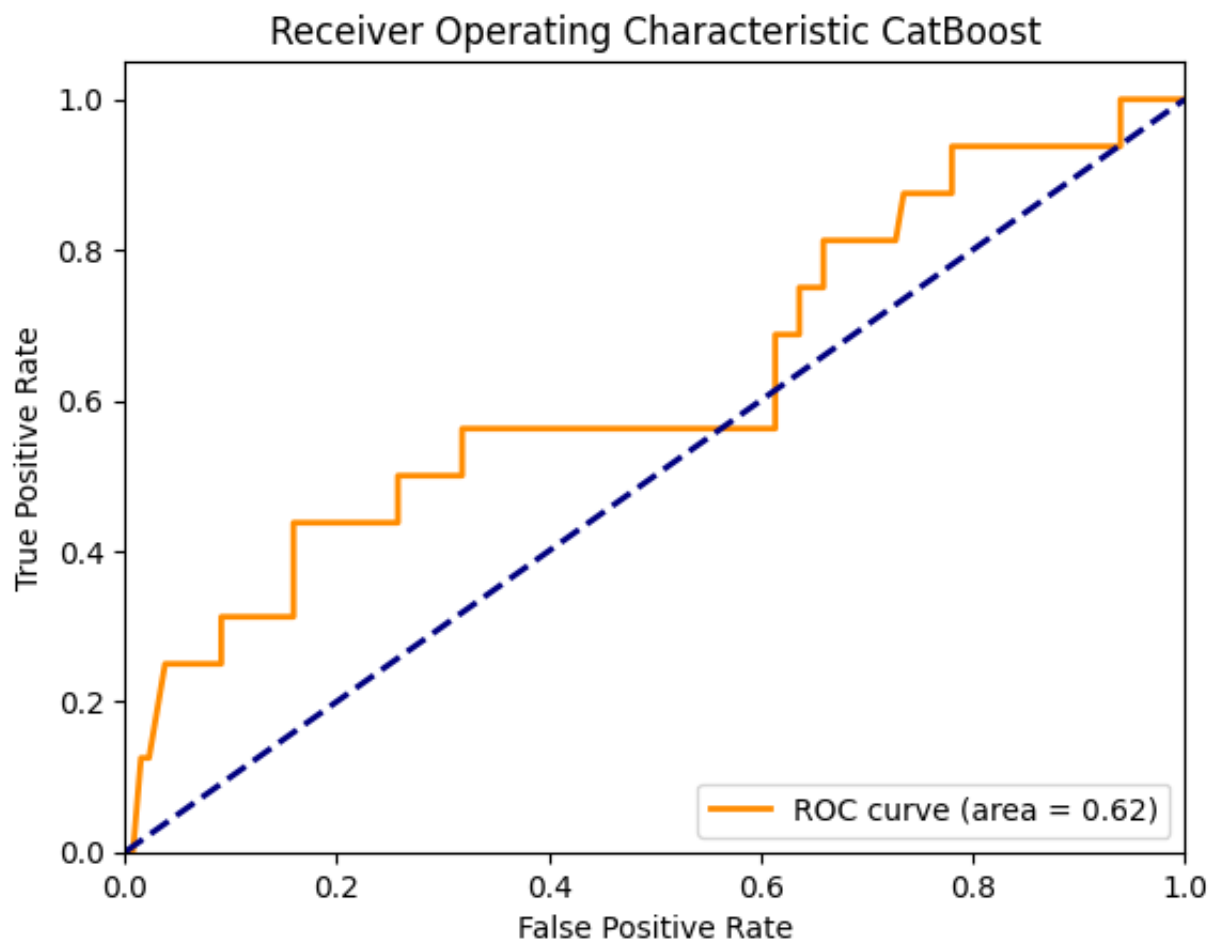
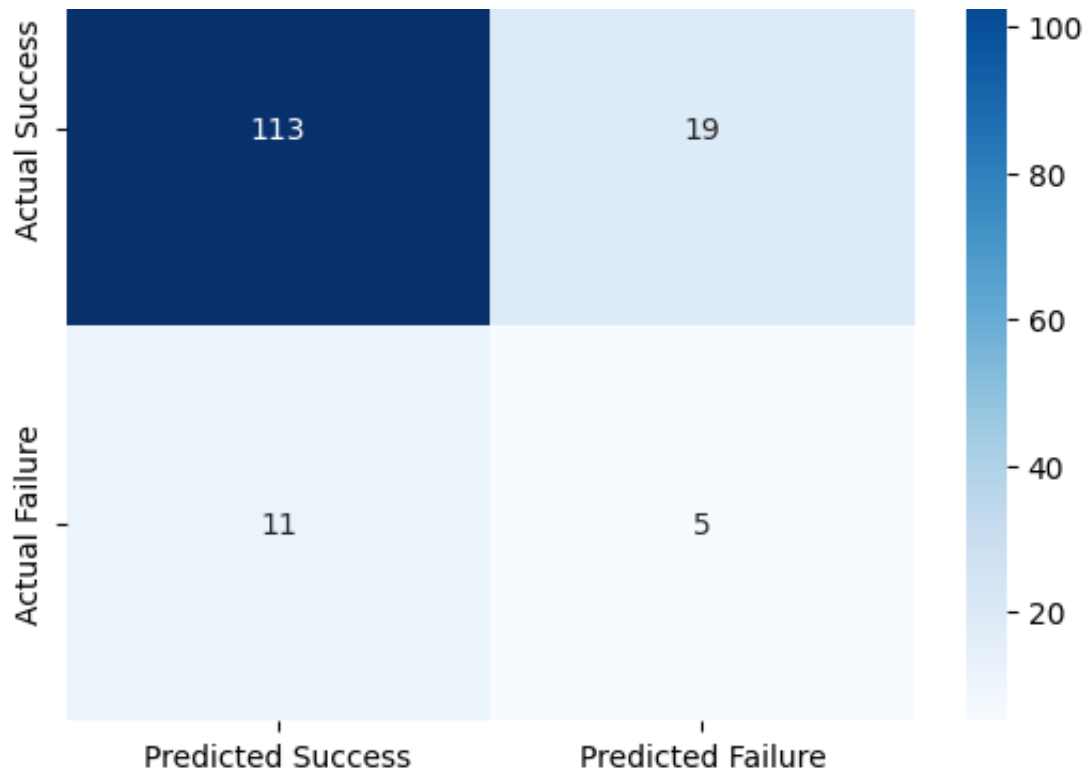
Threshold: 1.00, Metrics: {'Accuracy': 0.8918918918918919, 'Sensitivity': 0

SHAP Summary for CatBoost



Confusion Matrix CatBoost





ROC Curve Metrics:

```
FPR: [0.00000000 0.00757576 0.01515152 0.02272727 0.03787879 0.06060606
0.07575758 0.09090909 0.09090909 0.14393939 0.15909091 0.15909091
0.18181818 0.20454545 0.25757576 0.25757576 0.27272727 0.31818182
0.31818182 0.33333333 0.34848485 0.37121212 0.49242424 0.50757576
0.56818182 0.58333333 0.61363636 0.61363636 0.63636364 0.63636364]
```

```

0.65909091 0.65909091 0.72727273 0.73484848 0.78030303 0.78030303
0.81818182 0.83333333 0.84848485 0.86363636 0.93939394 0.93939394
1.          ]
TPR: [0.          0.          0.125  0.125  0.25   0.25   0.25   0.25   0.3125 0.3125
0.3125 0.4375 0.4375 0.4375 0.4375 0.5     0.5     0.5     0.5625 0.5625
0.5625 0.5625 0.5625 0.5625 0.5625 0.5625 0.5625 0.6875 0.6875 0.75
0.75   0.8125 0.8125 0.875   0.875   0.9375 0.9375 0.9375 0.9375 0.9375
0.9375 1.       1.       ]
ROC AUC: 0.622

```

Aggregated Test Set Metrics Across Seeds:

	accuracy	sensitivity	specificity	f1	roc_auc
0	0.797297	0.3750	0.848485	0.285714	0.653646
1	0.797297	0.4375	0.840909	0.318182	0.633759
2	0.817568	0.4375	0.863636	0.341463	0.650805
3	0.804054	0.4375	0.848485	0.325581	0.632339
4	0.817568	0.4375	0.863636	0.341463	0.654119
5	0.804054	0.3750	0.856061	0.292683	0.656013
6	0.810811	0.4375	0.856061	0.333333	0.656960
7	0.797297	0.4375	0.840909	0.318182	0.673532
8	0.783784	0.3750	0.833333	0.272727	0.631866
9	0.797297	0.3125	0.856061	0.250000	0.622396

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

Accuracy: Mean = 0.803, SE = 0.003, 95% CI = [0.795, 0.810]
Sensitivity: Mean = 0.406, SE = 0.014, 95% CI = [0.375, 0.438]
Specificity: Mean = 0.851, SE = 0.003, 95% CI = [0.844, 0.858]
F1: Mean = 0.308, SE = 0.010, 95% CI = [0.286, 0.330]
Roc_auc: Mean = 0.647, SE = 0.005, 95% CI = [0.635, 0.658]

