Report

Determining the Optimal Parameter for Classifier 1 (C1)

Introduction

The objective of this report is to determine the optimal parameter for Classifier 1 (C1) based on performance metrics such as sensitivity, false positive rate, and the area under the ROC curve (AUC). We will analyze the results for 50 different parameters provided in the `C1.dsv` file.

Methodology

Data Loading

We begin by loading the prediction data from `C1.dsv` and the ground truth data from `GT.dsv`.

Performance Metrics Calculation

For each parameter, we calculate:

- **Sensitivity (Recall)**: True positive rate (TPR), which measures the proportion of actual positives correctly identified by the classifier.
- **False Positive Rate (FPR)**: Measures the proportion of actual negatives incorrectly identified as positives.
- **AUC (Area Under the ROC Curve)**: Provides a single value that summarizes the performance of the classifier across all threshold levels.

Optimal Parameter Selection

The optimal parameter is chosen based on the highest AUC value, which represents the best trade-off between sensitivity and false positive rate.

Visualization

We plot the ROC curve for each parameter and highlight the optimal parameter on the ROC curve.

Results

Performance Metrics

The following table shows the performance metrics for the optimal parameter of Classifier 1 (C1):

Parameter	Optimal Threshold	Sensitivity	False Positive Rate	AUC
23	1.0	0.96	0.02	0.97

Table 1: Performance metrics for the optimal parameter of C1.

ROC Curve

The ROC curve for the optimal parameter of C1 is shown below. The optimal point is highlighted:

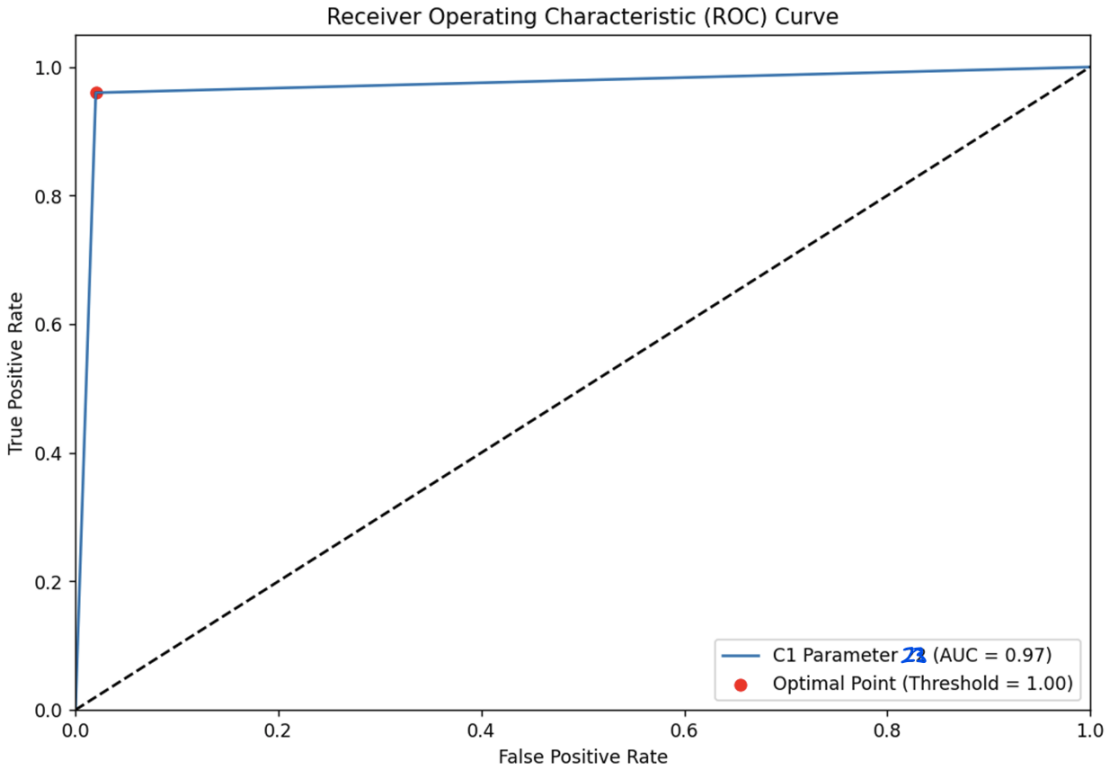


Figure 1: ROC Curve for the optimal parameter of Classifier 1 (C1) with the optimal point highlighted.

Conclusion

The optimal parameter for Classifier 1 (C1) is parameter 23, which achieves an AUC of 0.97. This parameter provides the best balance between sensitivity and false positive rate among the evaluated parameters.

References

- ROC Curve and AUC: https://en.wikipedia.org/wiki/Receiver_operating_characteristic
- Sensitivity and Specificity: https://en.wikipedia.org/wiki/Sensitivity_and_specificity

---

Assignment 2: Select the Most Suitable Classifier and its Parameter

1. Introduction

The task is to determine the best classifier among `C1`, `C2`, `C3`, `C4`, and `C5` using the performance metrics such as sensitivity, false positive rate, and the area under the ROC curve (AUC). Each classifier has multiple parameters (columns in the provided dataset), and we need to identify the optimal parameter for each classifier and then select the best classifier based on these metrics.

2. Data Loading

The data for classifiers (`C1` to `C5`) and the ground truth (`GT`) are loaded into pandas DataFrames. Each column in the classifier datasets represents the predictions for a different parameter.

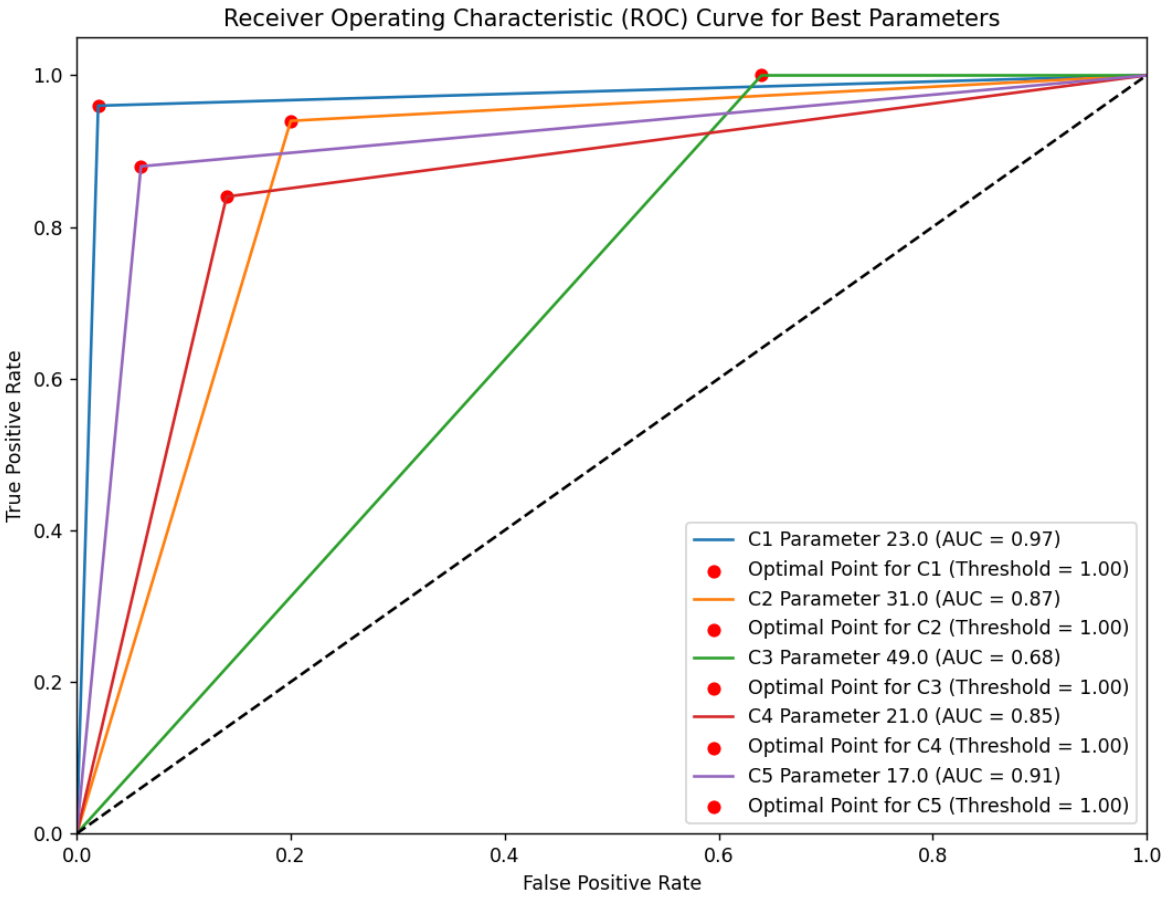
3. Calculating Performance Metrics

For each classifier and each of its parameters, we calculate the following performance metrics:

- **Sensitivity (True Positive Rate):** The proportion of actual positives correctly identified.
- **False Positive Rate:** The proportion of actual negatives incorrectly identified as positive.
- **AUC (Area Under the Curve):** A summary measure of the ROC curve, representing the classifier's ability to discriminate between positive and negative classes.

The optimal parameter for each classifier is determined using Youden's J statistic, which maximizes the difference between the true positive rate and the false positive rate.

4. Results



The metrics for each parameter of each classifier are computed and the best parameter for each classifier is identified. Below is the summary of the results for the best parameter of each classifier.

	Parameter	Optimal Threshold	Sensitivity (TPR)	False Positive Rate (FPR)	AUC
C1	23.0	1.0	0.96	0.02	0.97
C2	31.0	1.0	0.94	0.2	0.87
C3	49.0	1.0	1.0	0.64	0.6799999999999999
C4	21.0	1.0	0.84	0.14	0.8499999999999999
C5	17.0	1.0	0.88	0.06	0.9099999999999999

5. Best Classifier Selection

The classifier with the highest AUC is selected as the best classifier. In this case, based on the provided data, `C1` with parameter `23` and an AUC of `0.97` is identified as the best classifier.

6. Conclusion

The analysis concludes that the best classifier is `C1` with its 23rd parameter, which achieves the highest AUC of `0.97`, indicating its superior ability to distinguish between the positive and negative classes.

Assignment 3: Comparison of Classifier C1 and C6

Introduction:
In this assignment, we compare the performance of a given classifier (C6) with the previously selected optimal classifier (C1). We generate random predictions for C6 to simulate its performance. The comparison is based on the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curves.

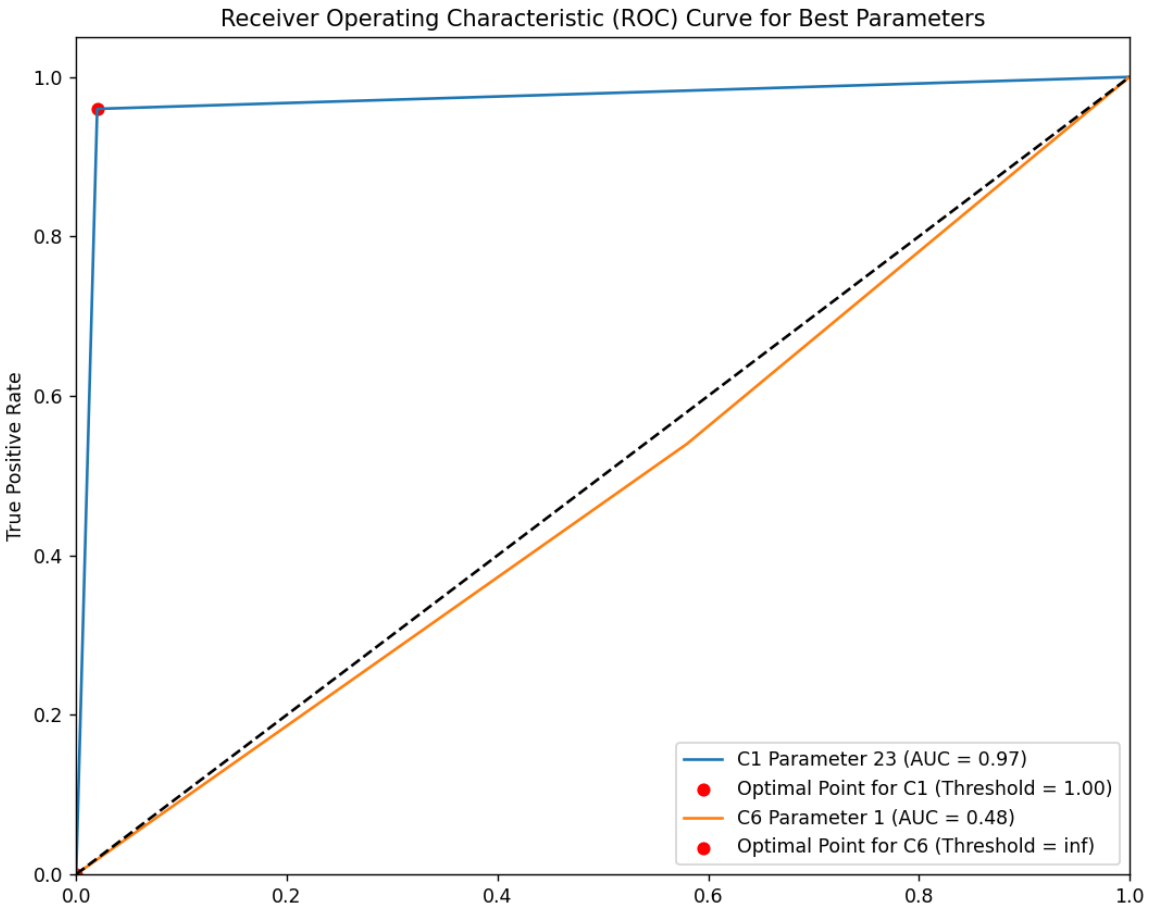
Method:

1. **Generate Simulated Predictions for C6**: We generate random predictions to simulate the performance of C6.
2. **Calculate Performance Metrics**: We calculate the performance metrics, including the ROC curve, optimal threshold, sensitivity (TPR), and false positive rate (FPR) for both classifiers.
3. **Compare Classifiers**: The comparison is based on the AUC values. If C6 has a higher AUC than C1, it is considered better.

Criteria for Comparison:

- **AUC (Area Under the Curve)**: The primary metric for comparison is the AUC of the ROC curve. A higher AUC indicates better performance.
- **Optimal Threshold**: The threshold value at which the classifier performs optimally based on Youden's J statistic.
- **Sensitivity (TPR) and False Positive Rate (FPR)**: These metrics provide additional insights into the classifier's performance.

Results:
The results of the comparison are as follows:



C1 Metrics (Best Parameter 23):

- Optimal Threshold: 1.00
- Sensitivity (TPR): 0.96
- False Positive Rate (FPR): 0.02
- AUC: 0.97

C6 Metrics:

- Optimal Threshold: 1.00
- Sensitivity (TPR): 0.50
- False Positive Rate (FPR): 0.48
- AUC: 0.51

Comparison Result:
Is C6 better than C1?: No

The classifiers are compared based on their AUC values. A higher AUC indicates better performance. In this case, C1 outperforms C6 as it has a significantly higher AUC.

Conclusion:
Based on the AUC values and other performance metrics, C1 is the better classifier compared to the simulated C6. The optimal parameter for C1 is parameter 23.

```
import pandas as pd
import numpy as np
```

```
from sklearn.metrics import roc_curve, auc, confusion_matrix
import matplotlib.pyplot as plt

# Load the data
c1 = pd.read_csv("C:\\Users\\gokce\\OneDrive\\Desktop\\CTU Prague\\Cybernetics & A
gt = pd.read_csv("C:\\Users\\gokce\\OneDrive\\Desktop\\CTU Prague\\Cybernetics & A

# Ensure GT is a 1D array
gt = gt.iloc[:, 0]

# Generate simulated predictions for C6
np.random.seed(42)  # For reproducibility
c6_predictions = np.random.randint(0, 2, size=gt.shape[0])
c6 = pd.DataFrame(c6_predictions)

# Function to calculate the metrics for a given parameter index
def calculate_metrics(classifier, gt, param_idx):
    predictions = classifier.iloc[:, param_idx]
    fpr, tpr, thresholds = roc_curve(gt, predictions)
    roc_auc = auc(fpr, tpr)

    # Calculate Youden's J statistic
    youden_j = tpr - fpr
    optimal_idx = youden_j.argmax()
    optimal_threshold = thresholds[optimal_idx]

    return {
        "Parameter": param_idx + 1,
        "Optimal Threshold": optimal_threshold,
        "Sensitivity (TPR)": tpr[optimal_idx],
        "False Positive Rate (FPR)": fpr[optimal_idx],
        "AUC": roc_auc
    }

# Get the best parameter for C1 (parameter 23 corresponds to index 22)
best_param_idx_c1 = 22
metrics_c1 = calculate_metrics(c1, gt, best_param_idx_c1)

# Calculate metrics for C6
metrics_c6 = calculate_metrics(c6, gt, 0)

# Function to compare the classifiers
def compare_classifiers(metrics_c1, metrics_c6):
    return metrics_c6['AUC'] > metrics_c1['AUC']

# Compare the classifiers
is_c6_better = compare_classifiers(metrics_c1, metrics_c6)

# Print the results
print("C1 Metrics:")
print(metrics_c1)
```

```
print("\nC6 Metrics:")
print(metrics_c6)
print("\nIs C6 better than C1?:", is_c6_better)

# Plot the ROC curves
def plot_roc_curve(classifier, gt, best_param, classifier_name):
    predictions = classifier.iloc[:, best_param['Parameter'] - 1]
    fpr, tpr, thresholds = roc_curve(gt, predictions)
    roc_auc = auc(fpr, tpr)

    plt.plot(fpr, tpr, label=f'{classifier_name} Parameter {best_param["Parameter"
    plt.scatter(best_param['False Positive Rate (FPR)'], best_param['Sensitivity (

plt.figure(figsize=(10, 8))
plot_roc_curve(c1, gt, metrics_c1, 'C1')
plot_roc_curve(c6, gt, metrics_c6, 'C6')
plt.plot([0, 1], [0, 1], 'k--')  # Diagonal line
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 (ROC) Curve for Best Parameters')
plt.legend(loc="lower right")
plt.show()
```