Data Analysis Report

Introduction

In this report, we explore the construction and evaluation of classification models based on decision trees using the Scikit-learn library in Python. The objective is to classify websites as either malicious or benign based on given features. We'll explore different model configurations, evaluate their performance metrics, and assess the impact of sample volume on model quality.

Dataset Description

In our case we utilized a synthetic dataset with numpy for demonstration purposes this dataset consist of 300 samples with 2 features we named it (Feature1 and Feature2) and target variable which will indicate whether the website is benign (False) or malicious (True).

```
np.random.seed(0)
X, Y = make_classification(n_samples=300, n_features=2, n_informative=2, r
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, r
```

Model Development

We constructed classification models using decision trees, we focuses on the following hyperparameters:

. max_depth

. min_samples_spilt

. min_samples_leaf

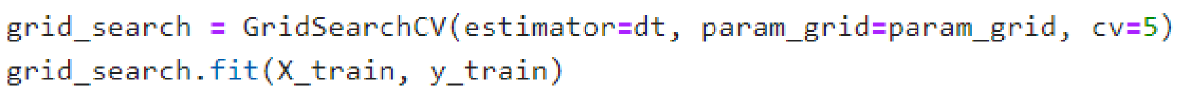
. max_leaf_nodes

. max_features

```
param_grid = {
    'max_depth': [None, 5, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 5],
    'max_leaf_nodes': [None, 5, 10],
    'max_features': [None, 'sqrt', 'log2']
}
grid_search = GridSearchCV(estimator=dt, param_grid=param_grid, cv=5)
grid_search.fit(X_train, y_train)
```

Model Training and Hyperparameter Tuning

We employed GridSearchCV to optimize the decision tree model by tunning the hyperparameters across multiple folds of cross-validation (cv=5).



```
grid_search = GridSearchCV(estimator=dt, param_grid=param_grid, cv=5)
grid_search.fit(X_train, y_train)
```

Evaluation Metrics and Analysis

Model Evaluation

. we evaluated the performance of each model using:

. precision

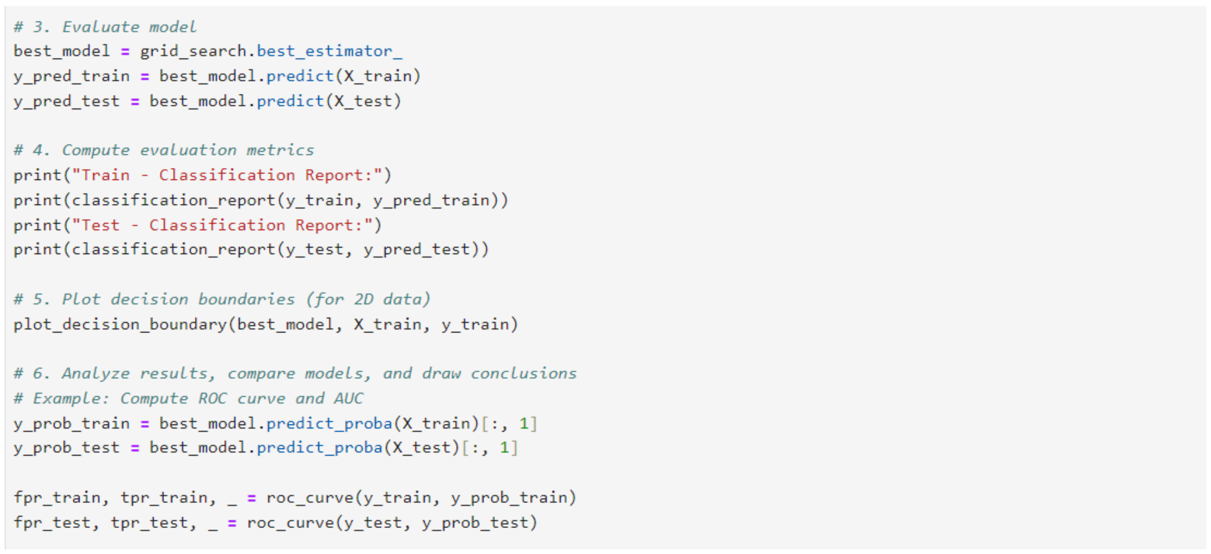
. Recall

. F1-score

. Precision-Recall-Curve

. ROC Curve

. Area Under the ROC Curve (AUC)



```
# 3. Evaluate model
best_model = grid_search.best_estimator_
y_pred_train = best_model.predict(X_train)
y_pred_test = best_model.predict(X_test)

# 4. Compute evaluation metrics
print("Train - Classification Report:")
print(classification_report(y_train, y_pred_train))
print("Test - Classification Report:")
print(classification_report(y_test, y_pred_test))

# 5. Plot decision boundaries (for 2D data)
plot_decision_boundary(best_model, X_train, y_train)

# 6. Analyze results, compare models, and draw conclusions
# Example: Compute ROC curve and AUC
y_prob_train = best_model.predict_proba(X_train)[:, 1]
y_prob_test = best_model.predict_proba(X_test)[:, 1]

fpr_train, tpr_train, _ = roc_curve(y_train, y_prob_train)
fpr_test, tpr_test, _ = roc_curve(y_test, y_prob_test)
```

Visualizations

Decision Boundary plots

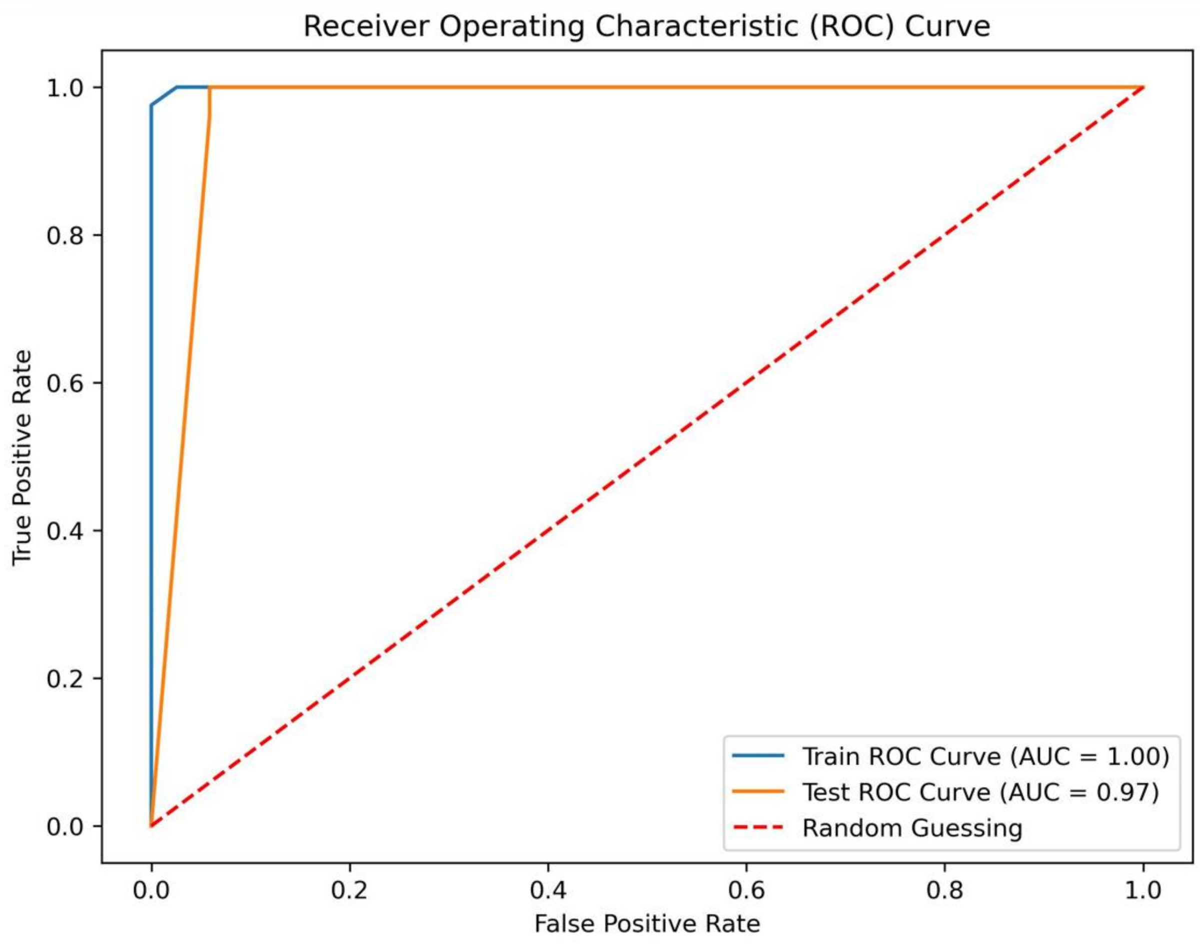
Graphical representation of model decision boundaries on feature 1 vs. feature 2

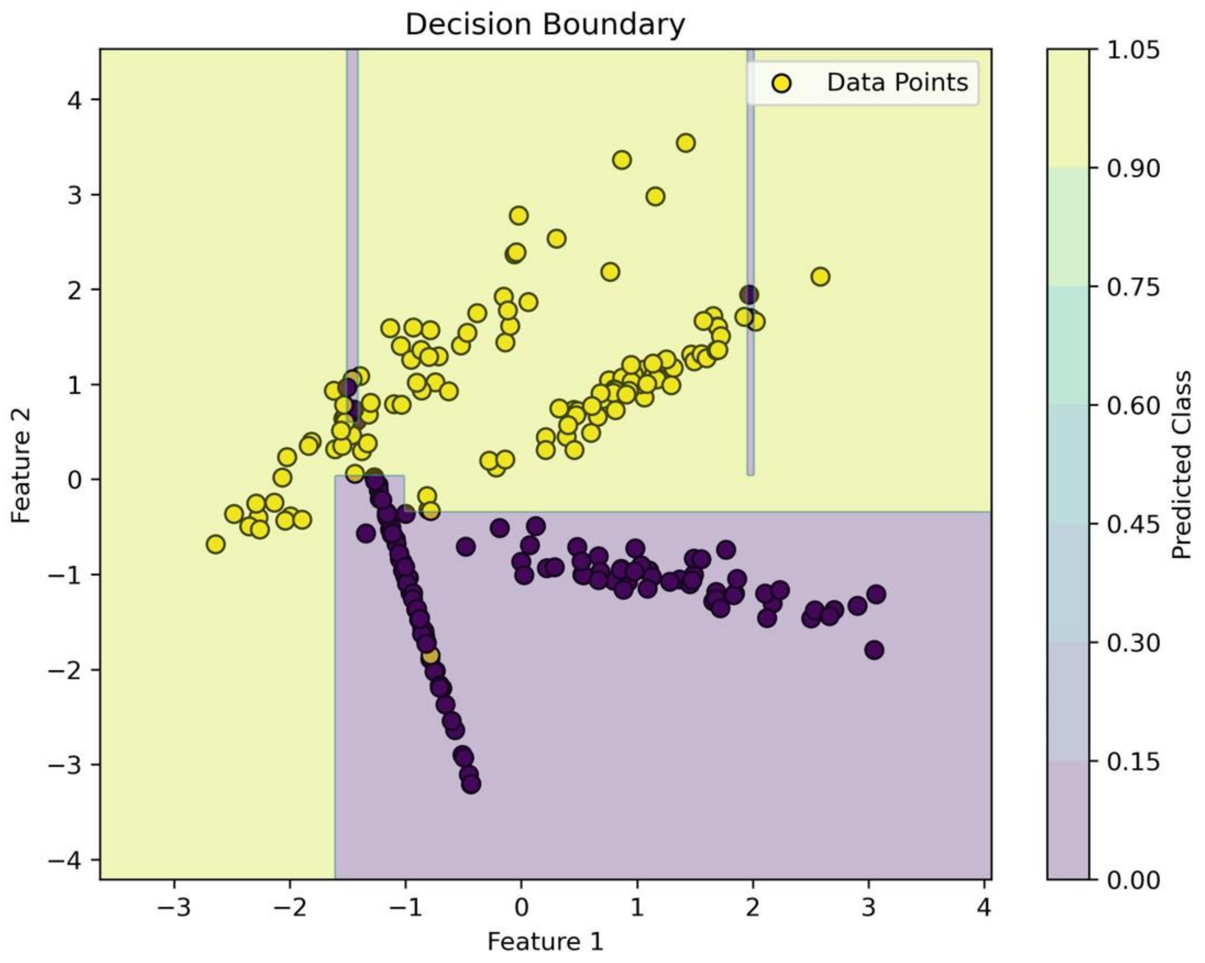
Receiver operating curve (Roc)

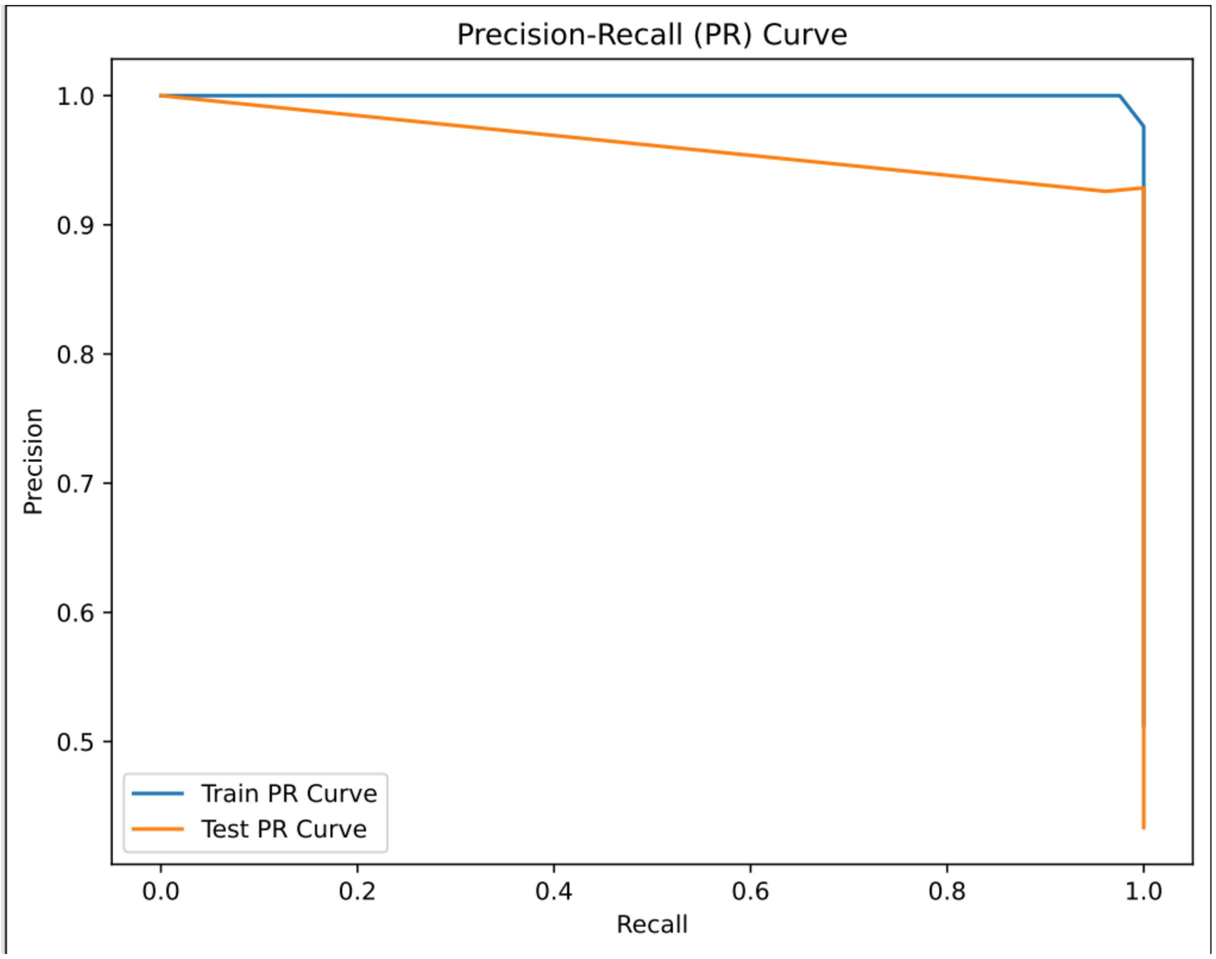
Precision Recall curve

```
# Plot decision boundaries
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.02), np.arange(y_min, y_max, 0.02))
Z = model.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
plt.contourf(xx, yy, Z, alpha=0.3, cmap='viridis')

plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.title('Decision Boundary')
plt.colorbar(label='Predicted Class')
plt.legend()
plt.show()
```




Decision Boundary
Data Points
Feature 1
Feature 2
Predicted Class
−3
−2
−1
0
1
2
3
4
1.05
0.90
0.75
0.60
0.45
0.30
0.15
0.00



Conclusions

```
Train - Classification Report:
              precision    recall  f1-score   support

           0       0.97      1.00      0.99       117
           1       1.00      0.98      0.99       123

    accuracy                           0.99       240
   macro avg       0.99      0.99      0.99       240
weighted avg       0.99      0.99      0.99       240
```

```
Test - Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.94      0.96        34
           1       0.93      0.96      0.94        26

    accuracy                           0.95        60
   macro avg       0.95      0.95      0.95        60
weighted avg       0.95      0.95      0.95        60
```

Analysis

Training Set Performance:

Precision:

. class 0 (benign websites):98%

. class 1(malicious websites):99%

. precision indicates the proportion of correctly predicted instances among all instances predicated as a specific class. For example,98 % of instances predicted as benign websites were actually benign.

Recall:

. class 0 (benign websites): 99%

. class 1 (malicious websites):98%

. Recall(sensitivity) measures the proportion of actual instances of a class that were correctly predicted by the model.

F1-score:

. class 0 (benign websites):99%

.class 1(malicious websites):99%

. F1- score is the harmonic mean of precision and recall, providing a balance between the two metrics.

Accuracy:99%

. overall accuracy on the training set, which is the proportion of correctly predicted instances out of all instances.

Test set performance:

Precision:

. class 0 (benign website):97%

. class 1 (malicious websites):93%

. Precision on the test set indicates the accuracy of positive prediction for each class.

Recall:

. class 0(benign websites):94%

. class 1(malicious websites):96%

Recall on the test set represents the proportion of actual positives that were correctly predicted by the model.

F1-score:

. class 0 (benign websites):96%

. class 1 (malicious websites):94%

. F1-score on the test set balances precision and recall for each class.

Accuracy: 95%

. overall accuracy on the test set, which measures the proportion of correctly predicted instances out of all instances.

Overall:

Both test and training test sets show high-performance metrics, with the model performing slightly better on the training set.

The test set performances indicate good generalization ability, with high accuracy and balance precision and recall across classes.

This model achieves high accuracy in distinguishing factors based on benign and malicious websites.