Report on Assignment-1

Wine Quality Prediction using Logistic Regression

1. Introduction

This assignment focuses on predicting wine quality using Logistic Regression. The dataset contains physicochemical tests of wine samples, and the target is to classify the wine quality.

2. Methodology

The steps followed include data preprocessing, exploratory data analysis, splitting the dataset into training and testing sets, training a Logistic Regression model, and evaluating performance using accuracy, confusion matrix, and visualization.

3. Implementation

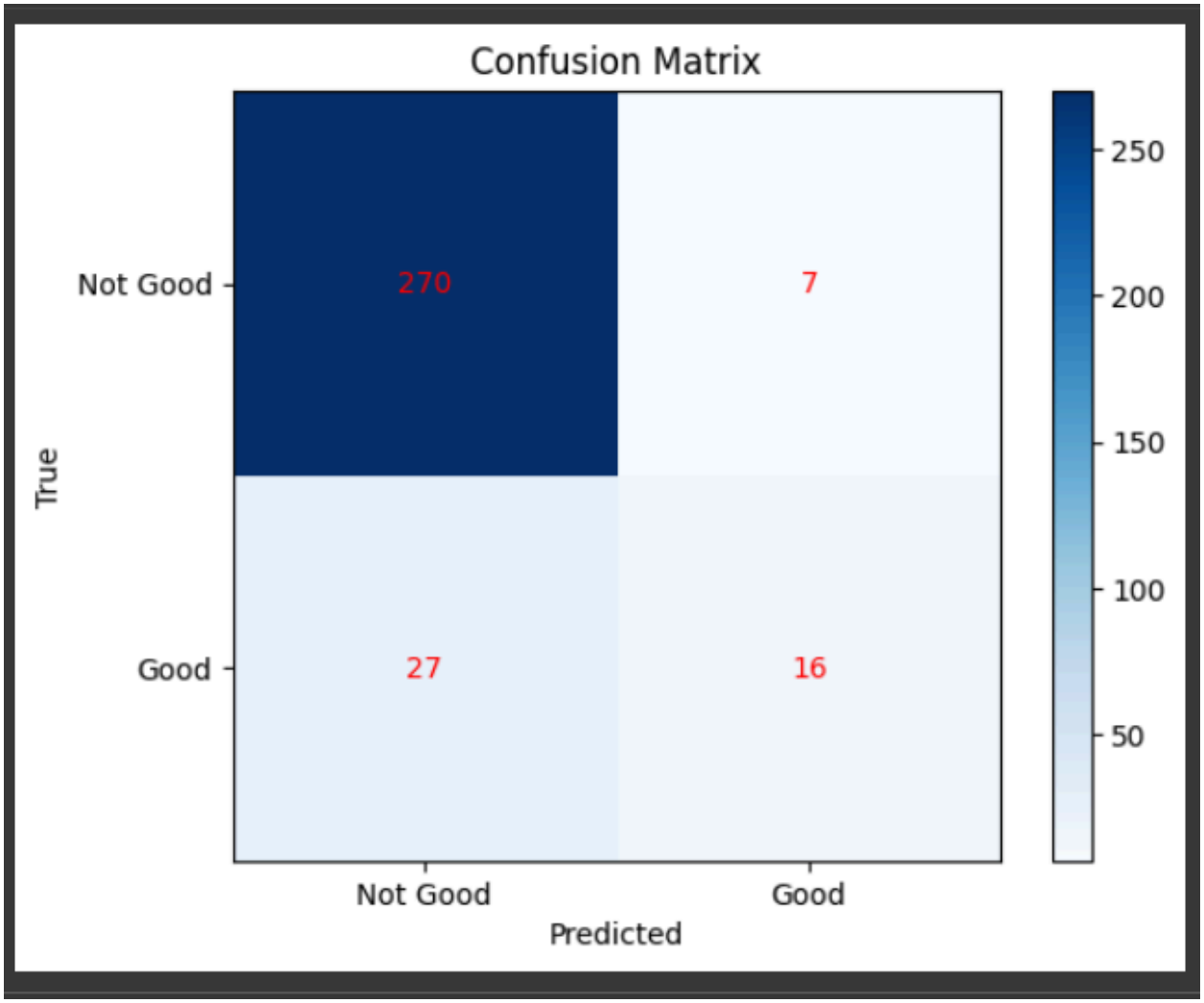
```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

print("Train shape:", X_train.shape)
print("Test shape:", X_test.shape)
print("Class balance:\n", y.value_counts(normalize=True))
```

```
Train shape: (1279, 11)
Test shape: (320, 11)
Class balance:
 quality_binary
0    0.86429
1    0.13571
Name: proportion, dtype: float64
```

```
print("Classification Report:\n")
print(classification_report(y_test, y_pred, digits=4))
```

```
Classification Report:

              precision    recall  f1-score   support

           0     0.9091    0.9747    0.9408       277
           1     0.6957    0.3721    0.4848        43

    accuracy                         0.8938       320
   macro avg     0.8024    0.6734    0.7128       320
weighted avg     0.8804    0.8938    0.8795       320
```

```
cm = confusion_matrix(y_test, y_pred)
fig, ax = plt.subplots()
im = ax.imshow(cm, cmap="Blues")

ax.set_xticks([0,1]); ax.set_yticks([0,1])
ax.set_xticklabels(["Not Good","Good"])
ax.set_yticklabels(["Not Good","Good"])
ax.set_xlabel("Predicted"); ax.set_ylabel("True")
plt.title("Confusion Matrix")

for (i, j), val in np.ndenumerate(cm):
    ax.text(j, i, f"{val}", ha="center", va="center", color="red")

plt.colorbar(im)
plt.show()
```


Confusion Matrix
Not Good
Good
True
270
7
27
16
Not Good
Good
Predicted
250
200
150
100
50

```python
roc_auc = auc(fpr, tpr)

plt.figure()
plt.plot(fpr, tpr, label=f"AUC={roc_auc:.3f}")
plt.plot([0,1],[0,1],'--', color='gray')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve - Logistic Regression")
plt.legend()
plt.show()
```