ML: Assignment - 3 (Q.2)

Mohd Talha Patrawala

CMPN-B

23102B0025

```
pip install ucimlrepo
```

```
Requirement already satisfied: ucimlrepo in /usr/local/lib/python3.12/dist-packages (0.0.7)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from ucimlrepo) (2.2.2)
Requirement already satisfied: certifi>=2020.12.5 in /usr/local/lib/python3.12/dist-packages (from ucimlrepo) (2026.1.4)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlrepo) (2.0
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlr
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlrepo) (2025
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlrepo) (20
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas>=1.0
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from ucimlrepo import fetch_ucirepo

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (
    confusion_matrix,
    precision_score,
    recall_score,
    f1_score,
    roc_auc_score,
    roc_curve
)
```

```
bank_marketing = fetch_ucirepo(id=222)

X = bank_marketing.data.features
y = bank_marketing.data.targets

y = y['y'].map({'yes': 1, 'no': 0})
```

```
X = pd.get_dummies(X, drop_first=True)

X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.25,
    random_state=42,
    stratify=y
)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

y_prob = model.predict_proba(X_test)[:, 1]
```

```
def evaluate_at_threshold(threshold):
    y_pred = (y_prob >= threshold).astype(int)

    cm = confusion_matrix(y_test, y_pred)
    tn, fp, fn, tp = cm.ravel()

    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    specificity = tn / (tn + fp)

    print(f"\nThreshold = {threshold:.4f}")
    print("Confusion Matrix:")
    print(cm)
```

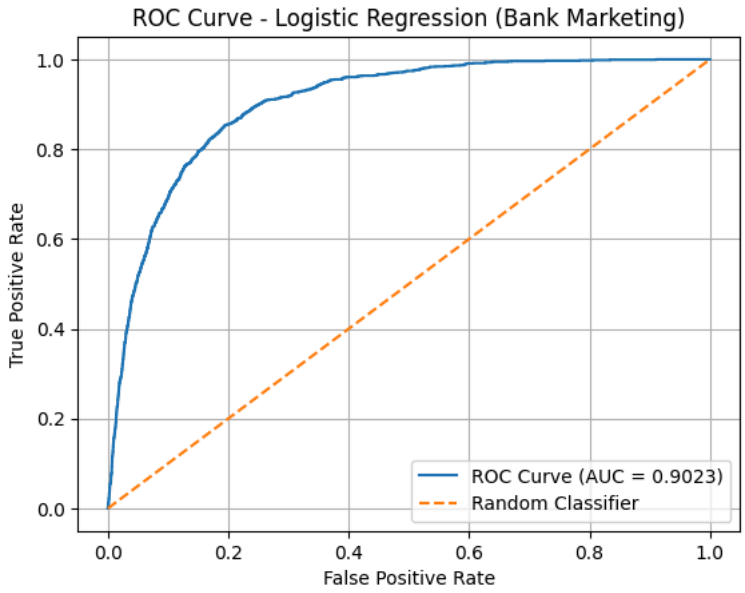
```
    print(f"Precision       : {precision:.4f}")
    print(f"Recall (Sens.)  : {recall:.4f}")
    print(f"Specificity     : {specificity:.4f}")
    print(f"F1-score        : {f1:.4f}")
```

```
print("=== Evaluation at Default Threshold (0.5) ===")
evaluate_at_threshold(0.5)
```

```
=== Evaluation at Default Threshold (0.5) ===

Threshold = 0.5000
Confusion Matrix:
[[9725  256]
 [ 880  442]]
Precision       : 0.6332
Recall (Sens.)  : 0.3343
Specificity     : 0.9744
F1-score        : 0.4376
```

```
roc_auc = roc_auc_score(y_test, y_prob)
print(f"\nROC-AUC Score: {roc_auc:.4f}")

plt.figure()
plt.plot(fpr, tpr, label=f"ROC Curve (AUC = {roc_auc:.4f})")
plt.plot([0, 1], [0, 1], linestyle='--', label="Random Classifier")

plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve - Logistic Regression (Bank Marketing)")
plt.legend()
plt.grid(True)
plt.show()
```

```
ROC-AUC Score: 0.9023
```




```
print("\n=== Evaluation at Optimized Threshold ===")
evaluate_at_threshold(0.221)

output_df = pd.DataFrame({
    "RecordId": y_test.index,
    "Probability(yes)": y_prob,
    "PredictedLabel": (y_prob >= 0.221).astype(int)
})

output_df.to_csv("probabilities.csv", index=False)

print("\nprobabilities.csv generated successfully.")
```

```
=== Evaluation at Optimized Threshold ===

Threshold = 0.2210
Confusion Matrix:
```

```
[[9245  736]
 [ 498  824]]
Precision       : 0.5282
Recall (Sens.)  : 0.6233
Specificity     : 0.9263
F1-score        : 0.5718

probabilities.csv generated successfully.
```

Why ROC curve is useful?

The ROC curve is useful because it shows how well a model performs across all possible thresholds, not just one fixed value. Since logistic regression outputs probabilities, the ROC curve helps evaluate how effectively the model separates positive and negative classes. The ROC–AUC score summarizes this in a single value, where a higher AUC means better overall discrimination.

What changes when the threshold changes?

When the threshold changes, the balance between precision and recall changes. Lowering the threshold increases recall by identifying more positive cases but reduces precision because more false positives occur. Raising the threshold increases precision and specificity but lowers recall by missing more true positives. Thus, changing the threshold adjusts the trade-off between catching more positives and avoiding false alarms.