ML: Assignment - 1 (Q.2)

Mohd Talha Patrawala

CMPN-B

23102B0025

```
pip install ucimlrepo
```

```
Collecting ucimlrepo
  Downloading ucimlrepo-0.0.7-py3-none-any.whl.metadata (5.5 kB)
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) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlrepo)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlrepo) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlrepo) (2025.3
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas>=1.0.0->
Downloading ucimlrepo-0.0.7-py3-none-any.whl (8.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.7
```

```
from ucimlrepo import fetch_ucirepo

# fetch dataset
breast_cancer_wisconsin_diagnostic = fetch_ucirepo(id=17)

# data (as pandas dataframes)
X = breast_cancer_wisconsin_diagnostic.data.features
y = breast_cancer_wisconsin_diagnostic.data.targets

# metadata
print(breast_cancer_wisconsin_diagnostic.metadata)

# variable information
print(breast_cancer_wisconsin_diagnostic.variables)
```

```
{'uci_id': 17, 'name': 'Breast Cancer Wisconsin (Diagnostic)', 'repository_url': 'https://archive.ics.uci.edu/dataset/17/breas
                  name     role         type demographic description units  \
0                   ID       ID  Categorical        None        None  None
1            Diagnosis   Target  Categorical        None        None  None
2              radius1  Feature   Continuous        None        None  None
3             texture1  Feature   Continuous        None        None  None
4           perimeter1  Feature   Continuous        None        None  None
5                area1  Feature   Continuous        None        None  None
6          smoothness1  Feature   Continuous        None        None  None
7         compactness1  Feature   Continuous        None        None  None
8           concavity1  Feature   Continuous        None        None  None
9      concave_points1  Feature   Continuous        None        None  None
10           symmetry1  Feature   Continuous        None        None  None
11  fractal_dimension1  Feature   Continuous        None        None  None
12             radius2  Feature   Continuous        None        None  None
13            texture2  Feature   Continuous        None        None  None
14          perimeter2  Feature   Continuous        None        None  None
15               area2  Feature   Continuous        None        None  None
16         smoothness2  Feature   Continuous        None        None  None
17        compactness2  Feature   Continuous        None        None  None
18          concavity2  Feature   Continuous        None        None  None
19     concave_points2  Feature   Continuous        None        None  None
20           symmetry2  Feature   Continuous        None        None  None
21  fractal_dimension2  Feature   Continuous        None        None  None
22             radius3  Feature   Continuous        None        None  None
23            texture3  Feature   Continuous        None        None  None
24          perimeter3  Feature   Continuous        None        None  None
25               area3  Feature   Continuous        None        None  None
26         smoothness3  Feature   Continuous        None        None  None
27        compactness3  Feature   Continuous        None        None  None
28          concavity3  Feature   Continuous        None        None  None
29     concave_points3  Feature   Continuous        None        None  None
30           symmetry3  Feature   Continuous        None        None  None
31  fractal_dimension3  Feature   Continuous        None        None  None

   missing_values
0              no
```

```
1             no
2             no
3             no
4             no
5             no
6             no
7             no
8             no
9             no
10            no
11            no
12            no
13            no
14            no
15            no
16            no
17            no
18            no
19            no
20            no
```

```
%%writefile task1_cancer_classification.py
from ucimlrepo import fetch_ucirepo
import argparse
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix

parser = argparse.ArgumentParser()
parser.add_argument("--test_size", type=float, default=0.2)
parser.add_argument("--model", type=str, default="logistic")
args = parser.parse_args()

dataset = fetch_ucirepo(id=17)

X = dataset.data.features
y = dataset.data.targets["Diagnosis"].map({"M": 1, "B": 0})

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

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

if args.model == "logistic":
    model = LogisticRegression(max_iter=1000)
    Xtr, Xte = X_train_s, X_test_s
else:
    model = DecisionTreeClassifier(random_state=42)
    Xtr, Xte = X_train, X_test

model.fit(Xtr, y_train)

train_pred = model.predict(Xtr)
test_pred = model.predict(Xte)

print("\nModel:", args.model.upper())
print("Train Error:", 1 - accuracy_score(y_train, train_pred))
print("Test Error :", 1 - accuracy_score(y_test, test_pred))

print("\nAccuracy :", accuracy_score(y_test, test_pred))
print("Precision:", precision_score(y_test, test_pred))
print("Recall   :", recall_score(y_test, test_pred))
print("F1-score :", f1_score(y_test, test_pred))

print("\nConfusion Matrix:")
print(confusion_matrix(y_test, test_pred))

print("\nConclusion:")
if args.model == "logistic":
    print(
        "The Logistic Regression model shows a small difference between training and test error, indicating good generalizatior
        "Its linear nature and regularization prevent it from memorizing noise, making it a strong baseline for this dataset."
    )
```

```
    else:
        print(
            "The Decision Tree model shows clear signs of overfitting. It achieves very high training performance but lower test pe
            "This occurs because an unconstrained tree learns overly complex decision rules that do not generalize well to unseen d
        )
```

```
Overwriting task1_cancer_classification.py
```

```
!python task1_cancer_classification.py --test_size 0.2 --model logistic
```

```
Model: LOGISTIC
Train Error: 0.01318681318681314
Test Error : 0.03508771929824561

Accuracy : 0.9649122807017544
Precision: 0.975
Recall   : 0.9285714285714286
F1-score : 0.9512195121951219

Confusion Matrix:
[[71  1]
 [ 3 39]]

Conclusion:
The Logistic Regression model shows a small difference between training and test error, indicating good generalization with no s
Its linear nature and regularization prevent it from memorizing noise, making it a strong baseline for this dataset.
```

```
!python task1_cancer_classification.py --test_size 0.2 --model tree
```

```
Model: TREE
Train Error: 0.0
Test Error : 0.07017543859649122

Accuracy : 0.9298245614035088
Precision: 0.9047619047619048
Recall   : 0.9047619047619048
F1-score : 0.9047619047619048

Confusion Matrix:
[[68  4]
 [ 4 38]]

Conclusion:
The Decision Tree model shows clear signs of overfitting. It achieves very high training performance but lower test performance,
This occurs because an unconstrained tree learns overly complex decision rules that do not generalize well to unseen data.
```

Comparison:-

Logistic Regression (Baseline Model):

- Training and test performance are very close
- Small generalization gap indicates good generalization
- No significant overfitting observed due to model simplicity and regularization

---

Decision Tree (Non-linear Model):

- Very high training performance
- Noticeably lower test performance
- Large generalization gap indicates overfitting

Relevant Machine Learning Issues:-

1. **Feature Scaling:** Logistic Regression is sensitive to feature scale, so standardization is necessary. Decision Trees are scale-invariant and do not require normalization.
2. **Feature Correlation (Multicollinearity):** Many features in the dataset are highly correlated, which can affect coefficient stability in Logistic Regression and lead to redundant splits in Decision Trees.

3. **Class Imbalance:** The dataset contains more benign than malignant samples, making accuracy alone insufficient; precision, recall, and F1-score are needed for reliable evaluation.