HYDERABAD INSTITUTE OF ARTS, SCIENCE, AND TECHNOLOGY

Artificial Intelligence Lab -3
Instructor: Miss Ayesha Eman
Date: 10/10/2025

Lab Title: Classification Algorithms in Machine Learning

Objective:

The objective of this lab is to introduce and implement classification algorithms in Machine Learning. Students will learn how to train models that classify data into categories (e.g., Pass or Fail) using Logistic Regression and Random Forest Classifier, evaluate their performance, and interpret key metrics like Accuracy, Precision, Recall, F1-Score, and the Confusion Matrix.

1. Introduction to Machine Learning

Machine Learning is a subset of Artificial Intelligence that enables systems to learn patterns and make decisions without being explicitly programmed. After data preprocessing, ML is the next step in the AI pipeline where cleaned and structured data is used to train models that can predict outcomes or uncover hidden patterns.

2. Types of Machine Learning

2.1 Supervised Learning

Supervised learning is a type of ML where the model is trained using labeled data, meaning each input has a known output. The goal is for the model to learn a mapping from inputs to outputs.
Examples of supervised learning include regression and classification problems.

2.2 Unsupervised Learning

In unsupervised learning, the data is unlabeled, the algorithm tries to discover hidden structures or patterns. It is useful for clustering, dimensionality reduction, and anomaly detection.

Example: Grouping students based on their learning behaviors without predefined categories.

3. Introduction to Classification

Classification is a supervised learning approach in which the model predicts a categorical outcome based on input features. It helps in decision-making where outcomes belong to distinct categories such as 'Yes/No', 'Spam/Not Spam', or 'Pass/Fail'. The goal is to train a model that learns from labeled data to predict the class of new, unseen data accurately.

3.1 Logistic Regression

Logistic Regression is a statistical model used for binary classification. It predicts the probability that a given input belongs to a particular class using a logistic (sigmoid) function.

Formula: $h(x) = 1 / (1 + e^{(-z)})$, where $z = w_0 + w_1x_1 + \dots + w_nx_n$

If $h(x) > 0.5 \rightarrow$ Class = 1 (Pass)
If $h(x) \leq 0.5 \rightarrow$ Class = 0 (Fail)

3.2 Random Forest Classifier

Random Forest is an ensemble learning method that builds multiple decision trees and merges their predictions for better accuracy and stability. Each tree is trained on random subsets of data and features, which helps prevent overfitting.

4. Implementation Using Python

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score

import matplotlib.pyplot as plt

import seaborn as sns

# Step 1: Load & Clean Dataset

data = pd.read_excel('students_dataset.xlsx')
```

```
# Drop non-relevant columns

data = data.drop(['Name'], axis=1)

# Handle numeric conversions

numeric_cols = ['Marks', 'Hours_Studied', 'Attendance']

for col in numeric_cols:

    data[col] = pd.to_numeric(data[col], errors='coerce')


# Fill missing numeric values

data = data.fillna(data.median(numeric_only=True))

# Encode categorical columns

le = LabelEncoder()

for col in ['Grade', 'Result']:

    data[col] = data[col].astype(str)

    data[col] = le.fit_transform(data[col])

# Step 2: Split Dataset

X = data.drop('Result', axis=1)

y = data['Result']

X_train, X_test, y_train, y_test = train_test_split(

    X, y, test_size=0.2, random_state=42

)

# Feature scaling for Logistic Regression

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
```

```
X_test_scaled = scaler.transform(X_test)

# Step 3: Train Both Models

# Random Forest

rf = RandomForestClassifier(random_state=42)

rf.fit(X_train, y_train)

rf_pred = rf.predict(X_test)

# Logistic Regression

lr = LogisticRegression(random_state=42)

lr.fit(X_train_scaled, y_train)

lr_pred = lr.predict(X_test_scaled)

# Step 4: Evaluation

print("=== RANDOM FOREST RESULTS ===")

print("Accuracy:", accuracy_score(y_test, rf_pred))

print("Confusion Matrix:\n", confusion_matrix(y_test, rf_pred))

print("\nClassification Report:\n", classification_report(y_test,
rf_pred))

print("\n=== LOGISTIC REGRESSION RESULTS ===")

print("Accuracy:", accuracy_score(y_test, lr_pred))

print("Confusion Matrix:\n", confusion_matrix(y_test, lr_pred))

print("\nClassification Report:\n", classification_report(y_test,
lr_pred))

# Step 5: Visual Comparison

plt.figure(figsize=(12,5))

# Confusion Matrix - Random Forest
```

```
plt.subplot(1, 2, 1)

sns.heatmap(confusion_matrix(y_test, rf_pred), annot=True, fmt='d',
cmap='Greens')

plt.title("Random Forest Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")

# Confusion Matrix - Logistic Regression

plt.subplot(1, 2, 2)

sns.heatmap(confusion_matrix(y_test, lr_pred), annot=True, fmt='d',
cmap='Blues')

plt.title("Logistic Regression Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.tight_layout()

plt.show()
```

4. Evaluation Metrics

To measure the performance of classification models, we use various evaluation metrics:

Accuracy

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision

Precision = TP / (TP + FP)

Recall

Recall = TP / (TP + FN)

F1-Score

F1 = 2 * (Precision * Recall) / (Precision + Recall)

Confusion Matrix

The confusion matrix shows how many predictions were correct and incorrect, divided by class. It helps identify if the model is biased towards a particular class.

6. Visualization and Results

The confusion matrix heatmap provides a visual summary of the model's performance. A high number of correct predictions (True Positives and True Negatives) indicate a well-performing model. In most real-world cases, Random Forest Classifier performs better due to its ensemble learning nature.

7. Discussion Questions

1. What are the main differences between classification and regression in machine learning?
2. How does the choice of algorithm affect model accuracy and interpretability?
3. Why is feature scaling important in certain machine learning algorithms?
4. How does the train-test split ratio impact model performance?
5. What is the significance of evaluation metrics such as accuracy, precision, recall, and F1-score?
6. How can a confusion matrix help evaluate classification performance?