

18B1WCI674: MACHINE LEARNING LAB

Assignment-7 (Binary and Multi-Class Classification using Scikit-learn)

Jan 05, 2026

Instructions

1. Use Jupyter Notebook or Google Colab.
2. Use scikit-learn for preprocessing, training and evaluation.
3. Follow the complete machine learning workflow.
4. Submit notebook and dataset within lab hours.

PART A: Binary Classification

Binary Classification involves predicting one of two classes (e.g. Spam or not spam).

Dataset Link: [Spambase Dataset](#)

1. Dataset Loading and Understanding

1. Install required libraries:

```
pip install pandas scikit-learn matplotlib seaborn
```

2. Import required libraries:

```
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

3. Load the dataset.
4. Display first 5 rows.
5. Display dataset shape.
6. Identify feature columns and target variable.
7. Check missing values.

2. Data Preprocessing

1. Handle missing values.
2. Encode categorical features (if any).
3. Apply feature scaling using StandardScaler.
4. Save preprocessed dataset.

3. Train-Test Split

1. Split dataset into 80% training and 20% testing.
2. Use `random_state = 42`.
3. Verify shapes of train and test data.

4. Model Training

1. Train Decision Tree model.
2. Fit models on training data.
3. Predict on test data.
4. Store predictions.
5. save model.

5. Model Evaluation

1. Load Saved Model.
2. Compute Accuracy.
3. Generate Classification Report.
4. Plot Confusion Matrix.

PART B: Multi-Class Classification

Multi-Class Classification involves predicting more than two classes.

Dataset Link: Glass Identification Dataset

1. Dataset Loading and Understanding

1. Load dataset.
2. Convert dataset into pandas DataFrame.
3. Display first 5 rows.
4. Display dataset shape.
5. Identify number of classes.

2. Data Preprocessing

1. Handle missing values (if any).
2. Encode target labels.
3. Apply feature scaling.
4. Save processed data.

3. Train-Test Split

1. Split dataset into 80% training and 20% testing.
2. Use `random_state = 42`.
3. Verify train and test shapes.

4. Model Training

1. Train Decision Tree.
2. Fit models on training data.
3. Predict on test data.
4. Store predictions.
5. save model.

5. Model Evaluation

1. Load Saved Model.
2. Compute Accuracy.
3. Generate Multi-Class Confusion Matrix.
4. Compute Macro and Micro Precision, Recall, F1-score.



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
WAKNAGHAT, P.O.-WAKNAGHAT, TEHSIL- KANDAGHAT,
DISTRICT- SOLAN (H.P.) PIN-173234 (INDIA)
(Established by H.P. State Legislature vide Act No. 14 of
2002)



Comparative Analysis

1. Compare Binary vs Multi-Class results.