



18B1WCI674: MACHINE LEARNING LAB

Assignment-8 (ANN)

Mar 20, 2026

Instructions

1. Use Jupyter Notebook or Google Colab.
 2. Use TensorFlow/Keras or PyTorch.
 3. Submit notebook and dataset within lab hours.
-

PART A: Binary Classification using ANN

Binary Classification predicts one of two classes using neural networks.

Dataset Link: Spambase Dataset

1. Dataset Loading and Understanding

1. Install required libraries:

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

2. Import required libraries:

```
import pandas as pd
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

3. Load dataset.
 4. Display first 5 rows.
 5. Display dataset shape.
 6. Identify features and target variable.
 7. Check missing values.
-

2. Data Preprocessing

1. Handle missing values.
2. Apply feature scaling using StandardScaler.
3. Convert target into binary format (if needed).
4. Save processed dataset.

3. Train-Test Split

1. Split dataset into 80% training and 20% testing using `random_state = 42`.
2. Verify shapes.

4. ANN Model Design

1. Build ANN using Sequential API.
2. Input Layer.
3. One or more Hidden Layers (ReLU activation).
4. Output Layer (Sigmoid activation).
5. Compile model using:
 - Loss: Binary Crossentropy
 - Optimizer: Adam
 - Metrics: Accuracy

5. Model Training

1. Train ANN using `model.fit()`.
2. Use validation split.
3. Plot training vs validation accuracy.
4. Save trained model.

6. Model Evaluation

1. Load saved model.
2. Predict on test data.
3. Convert probabilities to class labels.
4. Compute Accuracy.
5. Generate Classification Report.
6. Plot Confusion Matrix.

PART B: Multi-Class Classification using ANN

Multi-Class Classification predicts multiple classes.

Dataset Link: Glass Identification Dataset

1. Dataset Loading and Understanding

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

2. Data Preprocessing

1. Handle missing values.
2. Encode labels using One-Hot Encoding.
3. Apply feature scaling.
4. Save processed dataset.

3. Train-Test Split

1. Split dataset (80-20).
2. Verify shapes.

4. ANN Model Design

1. Build ANN model.
2. Input layer.
3. Hidden layers (ReLU activation).
4. Output layer with Softmax activation.
5. Compile model using:
 - Loss: Categorical Crossentropy
 - Optimizer: Adam
 - Metrics: Accuracy

5. Model Training

1. Train model using `model.fit()`.
2. Use validation split.
3. Plot training curves.
4. Save model.

6. Model Evaluation

1. Load saved model.
2. Predict on test data.
3. Compute Accuracy.
4. Generate Confusion Matrix.
5. Compute Precision, Recall, F1-score.

Comparative Analysis

1. Compare ANN vs Decision Tree (Assignment-7).
2. Discuss:
 - Accuracy differences
 - Training time
 - Overfitting behavior
 - Interpretability