Republic of the Philippines
**Laguna State Polytechnic University**
Province of Laguna

| Machine Problem No. 1 | | | |
|---|---|---|---|
| **Topic:** | **Fundamentals of Machine Learning** | **Week No.** | 2 |
| **Course Code:** | **CSST102** | **Term:** | 1st Semester |
| **Course Title:** | **Basic Machine Learning** | **Academic Year:** | 2025-2026 |
| **Student Name** | | **Section** | |
| **Due date** | | **Points** | |

## Fundamentals of Machine Learning

**Topic:** What is ML? Types of ML and Core Challenges

---

### Objectives

By the end of the lab, students should be able to:

1. Load and explore a dataset using Scikit-Learn.
2. Perform a train-test split to prepare data.
3. Train a simple baseline ML model.
4. Evaluate model performance using metrics.
5. Relate the task to supervised vs. unsupervised learning.

---

### Lab Outline (3 hours)

#### ⏰ Hour 1 – Setup & Dataset Exploration

- Install/verify Python, Jupyter/Colab, and Scikit-Learn.
- Load the **Iris dataset** (classification) or **California Housing dataset** (regression).
- Explore dataset (features, targets, summary statistics).

**Code Snippet:**

```
from sklearn.datasets import load_iris
import pandas as pd

# Load dataset
iris = load_iris(as_frame=True)
df = iris.frame
print(df.head())

# Explore
print(df.describe())
print("Target classes:", iris.target_names)
```

**Mini-task:** Students answer:

- What is the **input (features)**?
- What is the **output (label)**?
- Is this **supervised or unsupervised learning**?

---

### ⏰ Hour 2 – Train-Test Split & Baseline Model

- Perform train-test split (80% train, 20% test).
- Train a simple baseline model:
  - Logistic Regression (for Iris)
  - Linear Regression (for Housing)
- Make predictions.

**Code Snippet:**

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

X = df[iris.feature_names]
y = iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

**Mini-task:** Students compute model accuracy.

---

### ⏰ Hour 3 – Evaluation & Reflection

- Evaluate model with different metrics:
  - Classification: Confusion matrix, precision, recall.
  - Regression: RMSE (Root Mean Squared Error).
- Discuss ML challenges: overfitting, underfitting, and bad data.
- Students reflect:
  - "What would happen if the dataset had missing or wrong values?"
  - "How does this relate to real-world ML applications?"

**Code Snippet (Confusion Matrix):**

```
from sklearn.metrics import confusion_matrix, classification_report
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

**Deliverables (Lab Submission)**

1. Python notebook (Jupyter/Colab) with:
    - Dataset loading & exploration
    - Train-test split
    - Model training & evaluation
2. Short reflection (3–5 sentences):
    - What ML type did you use?
    - What challenge might affect the model?

---

**Assessment (30 points)**

- Dataset Exploration (5 pts)
- Train-Test Split (5 pts)
- Baseline Model Training (10 pts)
- Evaluation Metrics (5 pts)
- Reflection/Discussion (5 pts)

✅ **Total: 30 points**