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	Capili, Judeelyn M	**Section**	BSCS 3A
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).

In this activity, I verified my Python installation and used the Scikit-Learn library to explore two datasets: the Iris dataset and the California Housing dataset. The Iris dataset is used for classification tasks, while the California Housing dataset is used for regression. The Iris dataset contains 150 samples of iris flowers, each described by four numeric features: sepal length, sepal width, petal length, and petal width. The goal of the model is to predict the species of the flower — either Setosa, Versicolor, or Virginica. The California Housing dataset, on the other hand, includes 20,640 housing records. Each record contains eight numerical features: median income, house age, average number of rooms, average number of bedrooms, population, average occupancy, latitude, and longitude. The

model's goal is to predict the median house value of each district in hundreds of thousands of dollars.

These two datasets demonstrate two fundamental types of machine learning tasks. The Iris dataset represents a classification problem, where the output consists of discrete categories, while the California Housing dataset represents a regression problem, where the output is a continuous numeric value. Because both datasets include known outputs during training, they are examples of supervised learning, where the algorithm learns patterns from labeled data to make accurate predictions.

Mini-task: Students answer:

- What is the **input (features)**?

  The input features for the Iris dataset are sepal length, sepal width, petal length, and petal width.

  The input features for the California Housing dataset are median income, house age, average number of rooms, average number of bedrooms, population, average occupancy, latitude, and longitude.

- What is the **output (label)**?

  The output for the Iris dataset is the species of the flower (Setosa, Versicolor, or Virginica).

  The output for the California Housing dataset is the median house value of each district (in $100,000 units).

- Is this **supervised or unsupervised learning**?

  Both are supervised learning tasks because the datasets include labeled outputs that the models use to learn and make predictions.

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:

Mini-task: Students compute model accuracy.

After exploring the datasets, each one was divided into 80% training data and 20% testing data using Scikit-Learn's train_test_split() function. The Iris dataset was used to train a Logistic Regression model, a basic algorithm commonly applied to classification tasks. This model learned to distinguish between three species of iris flowers using the four flower measurements. It was trained using 120 samples and tested on 30 samples.

Meanwhile, the California Housing dataset was used to train a Linear Regression model, which fits a best-fit line (or hyperplane) to estimate housing prices. This model learned from 16,512 training samples and was tested on 4,128 samples. Linear Regression is a simple yet powerful method for establishing relationships between multiple features and a continuous target variable. Both models were trained successfully, and predictions were generated for the test sets to evaluate how accurately the models could generalize to new, unseen data.

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?"

To evaluate model performance, I applied the appropriate metrics for each learning type. For the Iris classification model, evaluation was done using the confusion matrix, precision, and recall. The confusion matrix showed that all samples were correctly classified into their respective species, resulting in zero misclassifications. The Logistic Regression model achieved an accuracy of 100%, with precision and recall both equal to 1.0, meaning the model perfectly predicted each flower species without errors.

For the California Housing regression model, evaluation was done using the Root Mean Squared Error (RMSE). The computed RMSE was 0.722, which means the model's predictions of house prices were off by an average of about $72,200. This indicates that while the Linear Regression model provides reasonable predictions, it does not perfectly fit the complex real-world housing data.

Through this evaluation, several machine learning challenges became evident. One issue is overfitting, which happens when a model learns the training data too well but performs poorly on new data because it fails to generalize. Another problem is underfitting, which occurs when a model is too simple, like Linear Regression failing to capture complex relationships in housing data. Additionally, bad data — such as missing, duplicated, or incorrect values — can greatly reduce model performance and reliability. Data preprocessing, cleaning, and normalization are essential steps to address these issues and ensure model accuracy.

If the dataset contained missing or wrong values, the models would likely produce inaccurate predictions or fail to train properly. For example, missing income data could distort house price predictions, while incorrect flower measurements could lead to misclassification. This shows how important data quality is in machine learning — a model is only as good as the data it learns from.

In terms of real-world applications, the Iris classification model demonstrates how ML can be used in biological research or agriculture to automatically identify species based on measurements. Meanwhile, the California Housing regression model reflects real-world predictive modeling in economics and real estate, where ML can estimate property values or assess market trends. These applications highlight the practical usefulness of ML systems and show how supervised learning supports data-driven decision-making.

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?

    This lab applied Supervised Learning through two models: Logistic Regression for classification (Iris dataset) and Linear Regression for regression (California Housing dataset). The classification model achieved perfect accuracy, while the regression model had an RMSE of 0.722. The main challenges faced include overfitting, underfitting, and poor data quality, all of which can reduce prediction reliability. If the data had missing or wrong values, the models' accuracy and stability would drop significantly. Overall, this activity helped me understand the end-to-end ML workflow — from dataset exploration to model training, evaluation, and reflection on real-world implications.

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