**TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES – MANILA**
College of Engineering
Electronics Engineering Department
Ayala Blvd., Ermita, Manila, 1000, Philippines
Tel. No. +632-301-3001 | Fax No. +632-521-4063

Programming Assignment 4

# SUPPORT VECTOR MACHINE

In this assignment, you will train a classifier using Support Vector Machine (SVM) to predict whether a breast tumor is benign (0) or malignant (1).

## Dataset

We will use the Breast Cancer Wisconsin dataset, built into sklearn.datasets. The dataset contains 569 samples of breast tumors, with 30 numerical features, including:

- Mean radius
- Mean texture
- Mean perimeter
- Mean area
- Mean smoothness, etc.

The target variable (y) represents tumor status:

- 0 = Benign
- 1 = Malignant

## General Guidelines

1. Load the dataset using sklearn.datasets.load_breast_cancer().
2. Split the dataset into 70% Training and 30% Testing, ensuring class distribution is maintained (use stratify=y in train_test_split).
3. Preprocess the data:
    - Use StandardScaler to normalize feature values.
4. Train an SVM model using sklearn.svm.SVC with default parameters.
5. Perform Hyperparameter Tuning using GridSearchCV to optimize:
    - C (Regularization parameter)
    - Gamma (Kernel coefficient for RBF)
    - Kernel type (linear, rbf, poly)
6. Evaluate the trained model:
    - Compute training and testing accuracy.
    - Identify the most important features using SelectFromModel.
    - Generate a confusion matrix and calculate:
        - F1-score
        - Precision
        - Recall
        - False Alarm Rate

## Guide Questions

Answer the following questions based on your results:

1. How did you preprocess the dataset (feature scaling, handling missing values if any, etc.)?
2. Why is it necessary to split the dataset into training and testing sets?
3. What is the role of StandardScaler in SVM training?
4. How does C affect the performance of an SVM model?
5. What is the purpose of the kernel function in SVM?
6. What were the best hyperparameters found using GridSearchCV?
7. What is a confusion matrix, and how is it interpreted?
8. How are Precision, Recall, and F1-score calculated from the confusion matrix?
9. If the model does not perform well, what adjustments can be made to improve it?

## Requirements

- Ensure that your code is clean, well-commented, and organized.
- Use Python libraries such as `numpy` and `pandas` for data manipulation and `matplotlib` or `seaborn` for visualization.

## Submission

1. Submit your work as a Jupyter Notebook (`.ipynb`) file.
2. Upload your Jupyter Notebook to your GitHub repository. Ensure the notebook is well-documented with markdown cells explaining each step and the corresponding results.
3. Provide the link to your GitHub repository for grading.