National University of Sciences and Technology
School of Electrical Engineering and Computer Science
Department of Computing

CS 471 Machine Learning

Fall 2024

Assignment 2

Support Vector Machine

Linear & Non-Linear SVM For Classification & Regression Problems

Announcement Date: 15th Nov 2024

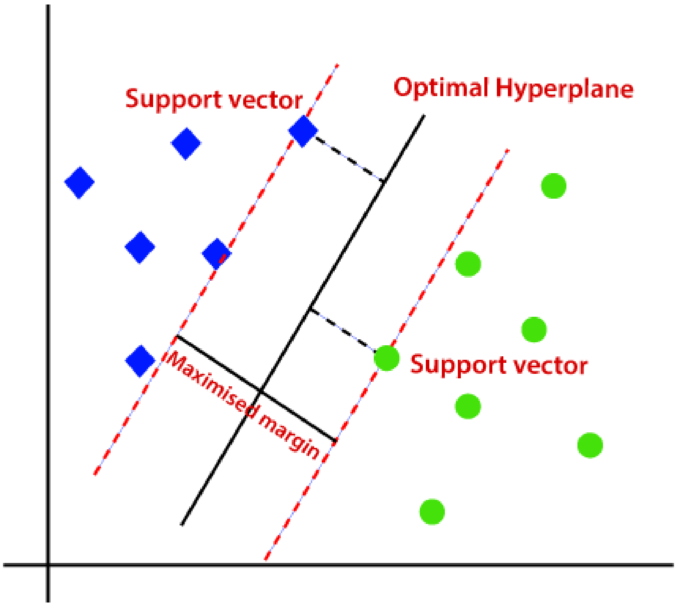
Due Date: 24th Nov 2024 at 11:59 PM (on LMS)

Instructor: Prof. Dr. Muhammad Moazam Fraz

Table of Contents

1. Introduction

In this assignment, you will be able to deepen your understanding of **Support Vector Machines (SVM)** by applying both **linear** and **non-linear** variants to **real-world datasets**. Building on the foundation established in the previous assignment, which focused on logistic regression, this assignment extends to the SVM approach for handling both classification and regression tasks. Through this exercise, you are going to engage with SVM's capability to identify optimal decision boundaries for classification and explore how it generalizes to continuous data for regression. The objective is to not only comprehend the mathematical and conceptual underpinnings of SVM but also to gain practical experience in handling diverse data types, preprocessing techniques, and model evaluation metrics specific to SVMs. You will also have the opportunity to experiment with **different kernel functions**, analyze the **effects of tuning hyperparameters**, and compare the performance between linear and non-linear approaches, thereby equipping yourself with versatile skills essential in machine learning.



1.1 Problem Statement

In this assignment, you will explore the application of Support Vector Machines (SVM) to solve two distinct problems: a classification problem and a regression problem. The goal is to use both linear and non-linear SVM models to analyze and predict outcomes based on real-world data.

Classification Problem - You will work with the **Rock Classification Dataset**, which involves classifying different types of rocks or minerals based on various physical features. Each instance in this dataset represents a rock sample, and the objective is to predict the rock's class (a discrete value) based on its attributes (surface area, perimeter, eccentricity, etc.) You will apply both linear and non-linear SVM techniques to evaluate their performance on this classification task.

Regression Problem - The **Theme Park Visitor Count Dataset** will be used to predict the total number of visitors to a theme park on an hourly basis. The goal is to build a regression model that estimates the number of visitors based on multiple features (weather conditions, temperature, time of day, etc.) Once again, you will explore the use of linear and non-linear SVM for regression and compare their efficacy in predicting continuous values.

1.2 Purpose and Objectives

The purpose of this assignment is to provide you with a practical understanding of how to apply Support Vector Machines (SVM) for solving real-world machine learning problems in both classification and regression tasks. By using both linear and non-linear SVM models, you will be able to develop valuable understanding for the behavior of SVMs under different conditions and learn how to select the most appropriate model for specific tasks.

The purpose of this assignment is to:

- To familiarize you with the **core principles** of Support Vector Machines.
- To implement and evaluate SVM on the **Rock Classification Dataset**, where the goal is to predict the type of rock or mineral based on its physical attributes.
- To apply SVM for regression using the **Theme Park Visitor Count Dataset**, where the goal is to predict the total number of visitors based on various features.
- To guide you through essential steps in **data exploration** and **preprocessing**.
- To encourage you to **critically analyze** your results, reflecting on the **model selection process**, and drawing conclusions about the **practical implications** of SVM.

1.3 Assignment Structure

This assignment is structured into two main tasks: a Classification Task and a Regression Task, each with specific objectives designed to guide you through the practical application of Support Vector Machines (SVM).

- **Classification Task -** In this section, you will be introduced to the Rock Classification Dataset. You need to explore the dataset, perform exploratory data analysis (EDA), and carry out necessary preprocessing steps. Implement both linear and non-linear (RBF and polynomial) SVM models for the classification task. Make sure to focus on evaluating the classification models accordingly.
- **Regression Task -** In this section, you will work with the Theme Park Visitor Count Dataset. You need to explore the dataset, clean the data, and perform necessary preprocessing steps, like handling outliers and scaling features. Implement both linear and non-linear SVM regression models and then evaluate them accordingly.
- **Comparative Analysis -** Critically analyze the performance of linear and non-linear SVM models for both classification and regression.
- **Submission Guidelines -** Detailed instructions on how to submit the completed assignment, including required file formats.
- **Grading Rubric -** Evaluation criteria for the assignment, detailing how your submissions will be assessed.

2. Classification Task (30 Marks)

2.1 Dataset Overview (Rock Classification Dataset)

The **Rock Classification Dataset** contains various physical attributes of rock and mineral samples. The goal of this dataset is to classify the type of rock or mineral based on its measurable characteristics. The dataset consists of several features related to the shape and size of the samples, which are used to determine their classification. Each record in the dataset represents a unique rock sample, and the target variable is the Class, which denotes the type of rock or mineral (with 7 possible categories). The dataset includes the following features:

- **Area** - The total area of the rock/mineral sample, providing an indication of the sample's size.
- **Perimeter** - The length of the boundary or perimeter of the rock/mineral sample, used to describe the shape of the sample.
- **MajorAxisLength** - The longest axis measured within the sample, useful for understanding the overall shape.
- **MinorAxisLength** - The shortest axis measured perpendicular to the longest dimension, providing a measure of the sample's elongation.
- **AspectRation** - The ratio of the longest dimension to the shortest dimension, indicating how elongated or compact the sample is.
- **Eccentricity** - A measure of the shape of the rock/mineral, representing how much it deviates from being a perfect circle (0 means a circle, 1 indicates a more elongated shape).
- **ConvexArea** - The area of the smallest convex polygon that can enclose the sample, often used to quantify compactness and shape.
- **EquivDiameter** - The diameter of a circle with the same area as the sample, helping to compare the roundness and compactness of different shapes.
- **Extent** - The ratio of the sample's area to the area of its bounding box, providing a measure of how well the sample fills its bounding box.
- **Solidity** - The ratio of the convex area to the actual area of the sample, indicating how compact or fragmented the shape is.
- **Roundness** - A metric to quantify the roundness of the sample, which is calculated using the sample's area and perimeter.
- **Compactness** - A measurement of how close the shape is to a perfect circle, derived from the equivalent diameter and longest dimension.
- **ShapeFactor1 (SF1)**, **ShapeFactor2 (SF2)**, **ShapeFactor3 (SF3)**, **ShapeFactor4 (SF4)**: A series of shape factors that provide additional shape-related information, useful for distinguishing between different types of rocks or minerals.
- **Class** - The target variable, representing the type of rock or mineral. It is a categorical variable with 7 possible classes (1, 2, 3, 4, 5, 6, 7).

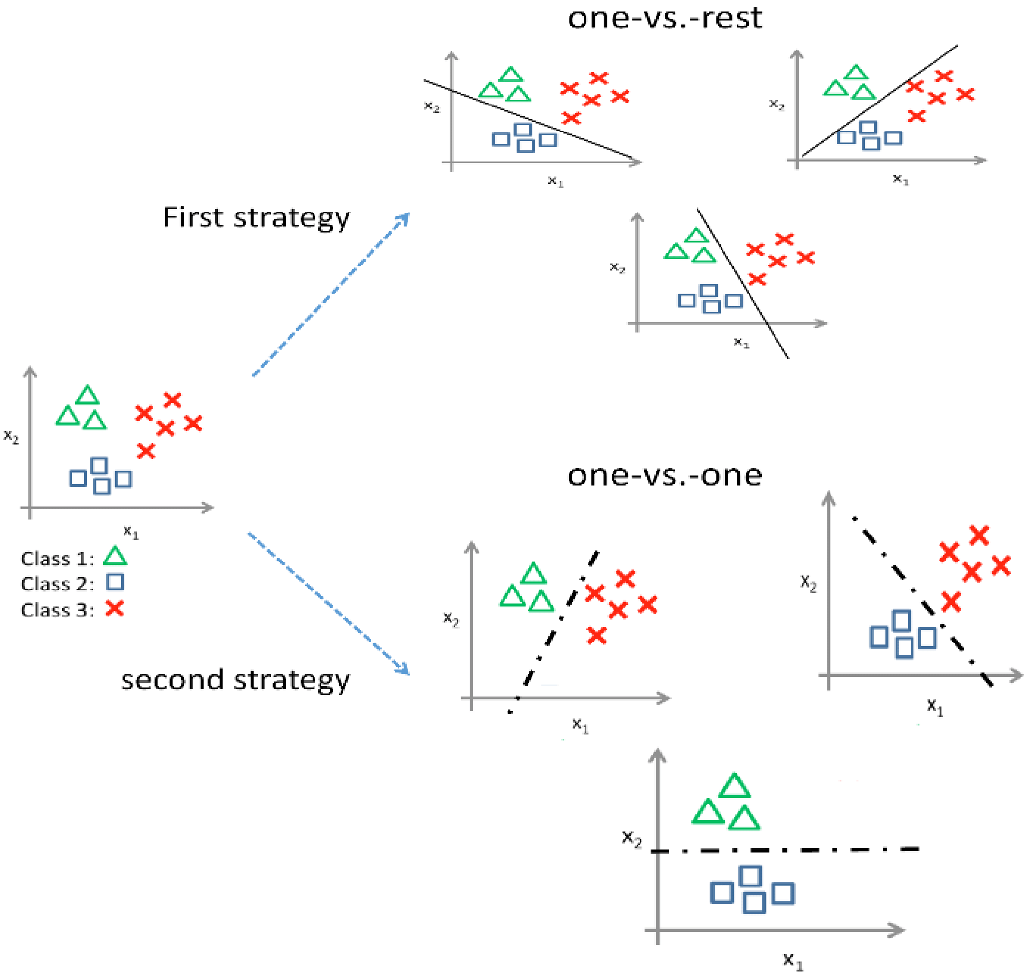
2.2 Data Exploration and Preprocessing (7 Marks)

Begin by examining the distribution of each feature (e.g., Sample_Area, Surface_Perimeter, Aspect_Ratio) through summary statistics and visualizations (histograms, box plots). This will help you identify potential outliers, skewed distributions, or data imbalances. Investigate the correlation between features using a correlation matrix to identify any highly correlated variables that may require dimensionality reduction or removal.

Check for missing or null values in the dataset. If any missing values are found, decide whether to fill them with the mean, median, or mode, or remove rows/columns with excessive missing data. Since SVM is sensitive to the scale of input features, it is essential to normalize or standardize the features. Use techniques such as Min-Max Scaling or Z-score normalization to ensure all features contribute equally to the model. Split the dataset into training and testing sets, typically with an 80-20 or 70-30 ratio. This will allow you to train the model on one subset and evaluate its performance on an unseen set of data.

2.3 SVM Implementation (15 Marks)

Start by implementing a **linear SVM** classifier. This model is suitable for cases where the data is linearly separable or nearly linearly separable. Use the training data to train the model, and select an appropriate value for the **regularization parameter (C)** to prevent overfitting and underfitting. Next, implement a **non-linear SVM** classifier using the **Radial Basis Function (RBF) kernel**. The RBF kernel is commonly used when the data is not linearly separable and requires mapping to a higher-dimensional space for better separation. You will need to fine-tune the kernel parameters (C and gamma) to optimize the model's performance.



For both linear and non-linear SVMs, perform **hyperparameter tuning** using **GridSearchCV** or **RandomizedSearchCV** to find the optimal combination of parameters. This will help improve the model's performance by selecting the best parameters that generalize well to unseen data. Train both SVM models on the preprocessed training dataset. After training, evaluate their performance using the testing dataset to assess how well the models generalize to unseen data.

2.4 Model Evaluation (8 Marks)

Use a variety of classification metrics to evaluate model performance.

- **Accuracy** - The proportion of correctly classified instances to the total number of instances.
- **Precision, Recall, F1-Score** - These metrics provide a deeper understanding of the model's performance, especially in imbalanced datasets. Precision measures the accuracy of positive predictions, recall evaluates how well the model identifies positive instances, and the F1-score is the harmonic mean of precision and recall.

- **Confusion Matrix** - Visualize the confusion matrix to see the breakdown of correct and incorrect predictions across all classes. This will help identify if certain classes are being misclassified more than others.
- **ROC-AUC (Receiver Operating Characteristic - Area Under Curve)** - For multi-class problems, calculate the macro-average ROC-AUC score to measure the trade-off between the true positive rate and false positive rate.

Perform **k-fold cross-validation** (e.g., k=5) to make sure that the model's evaluation is robust and not overly dependent on any single split of the data. This will help assess the model's stability and generalizability across different subsets of the dataset.

3. Regression Task (30 Marks)

3.1 Dataset Overview (Theme Park Visitor Count Dataset)

The **Theme Park Visitor Count Dataset** is used for the regression task, where the objective is to predict the **total visitor count** (**cnt**) based on various environmental and operational features. These features describe different aspects of a theme park's operations and environmental conditions that influence the number of visitors.

The dataset contains the following columns:

- **instant** - Unique record index (similar to ticket number).
- **dteday** - Date of the record.
- **season** - The theme park season (1: Early Spring, 2: Summer Break, 3: Fall, 4: Winter Holidays).
- **yr** - Year of the record (0: 2021, 1: 2022).
- **mnth** - Month of the record (1 to 12).
- **hr** - Hour of the day (0 to 23).
- **holiday** - Whether the day is a public holiday (1 if holiday, 0 if not).
- **weekday** - Day of the week (0: Sunday, 6: Saturday).
- **workingday** - Indicates if the day is a regular weekday (1 if yes, 0 if weekend or holiday).
- **weathersit** - Weather conditions at the park (1: Clear, Few clouds, 2: Misty, 3: Light Rain/Snow, 4: Heavy Rain/Fog).
- **temp** - Normalized outdoor temperature in Celsius.
- **atemp** - Normalized "feels like" temperature.
- **hum** - Normalized humidity level.
- **windspeed** - Normalized wind speed.
- **casual** - Count of casual visitors (single-day ticket holders).
- **registered** - Count of registered visitors (annual pass holders).
- **cnt** - Total visitor count (sum of casual and registered visitors, which is the target variable for regression).

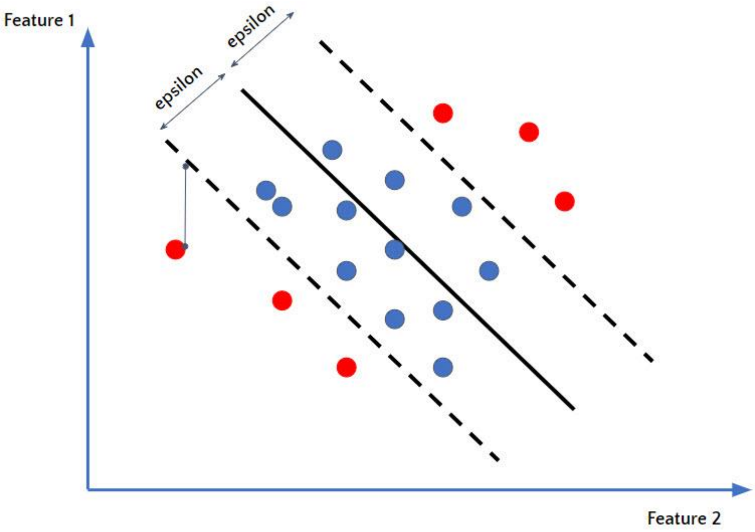
3.2 Data Exploration and Preprocessing (7 Marks)

Begin by examining the distribution of the target variable (**cnt**) and other numerical features. Visualize the data using histograms and box plots to understand their spread and identify potential outliers. Analyze categorical features to understand their distribution and the potential impact on visitor count. Use **pair plots** or **correlation matrices** to identify relationships between features and how they correlate with the target variable (**cnt**). This will help you spot any strong linear or non-linear dependencies.

Check for missing or null values in the dataset. If missing values are found, decide on the appropriate imputation strategy. For numerical features, you can fill missing values with the mean or median, and for categorical features, the most frequent category may be used. Since regression models are sensitive to the scale of input features, normalize or standardize the continuous features to bring them onto a similar scale. Split the dataset into training and testing sets (80% for training and 20% for testing).

3.3 SVM Implementation (15 Marks)

First, implement the linear SVM regression model using Scikit-learn's **SVR** class. A linear SVM will attempt to find a hyperplane that best fits the data in a linear fashion. Configure the model with the **kernel='linear'** parameter to ensure it uses a linear kernel. Train the model on the training dataset and make predictions on the test set. Next, implement the non-linear SVM regression model using the Radial Basis Function (RBF) kernel. The RBF kernel is often useful for capturing complex relationships in data. Set the **kernel='rbf'** parameter in the SVR class to apply the RBF kernel. You may also need to tune the **C**, **epsilon**, and **gamma** hyperparameters for optimal performance. Train the non-linear model on the training set and generate predictions for the test set.



For both models (linear and non-linear), perform **hyperparameter tuning** using grid search (**GridSearchCV**) or randomized search (**RandomizedSearchCV**) to find the best values for parameters like **C** (regularization parameter), **epsilon** (margin of tolerance), and **gamma** (kernel coefficient for RBF). Tune these hyperparameters to minimize the error between predicted and actual visitor counts.

Fit both the linear and non-linear SVM models on the training data. After training, use the models to make predictions on the test set, which will allow you to evaluate the models' performance and compare their results.

3.4 Model Evaluation (8 Marks)

- **Mean Absolute Error (MAE)** - This metric calculates the average absolute difference between the predicted and actual visitor counts. It provides an intuitive understanding of the model's performance in terms of the actual magnitude of error.

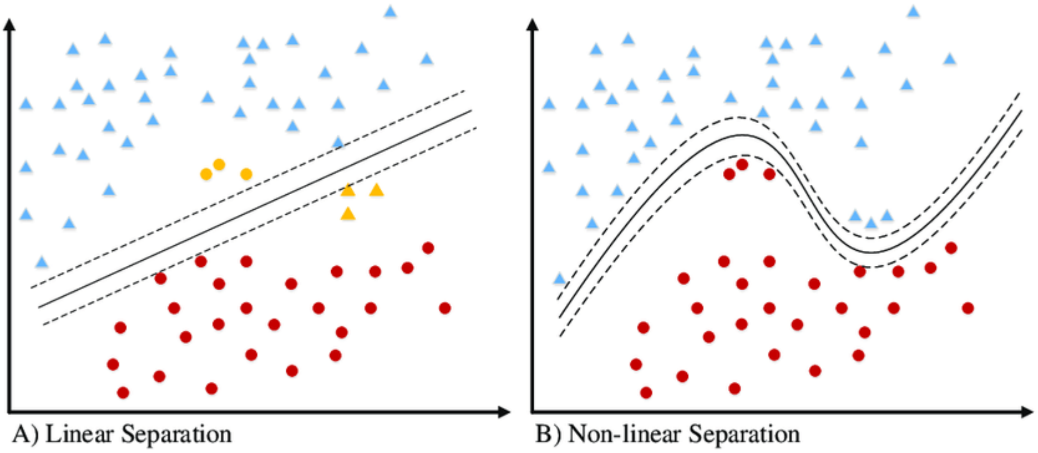
- **Mean Squared Error (MSE)** - This metric calculates the average of the squared differences between predicted and actual values. MSE penalizes larger errors more heavily, making it sensitive to outliers.
- **Root Mean Squared Error (RMSE)** - The square root of MSE provides an error metric in the same unit as the target variable, making it easier to interpret.
- **R-squared (R^2)** - This metric indicates how well the model explains the variance in the target variable. A higher R^2 value (closer to 1) means the model explains more of the variance, while a value closer to 0 suggests a poor fit.

After evaluating both the linear and non-linear SVM models using the above metrics, compare their performances to determine which model better predicts the total visitor count. Visualize the predicted versus actual values for both models using scatter plots. Perform **k-fold cross-validation** (e.g., k=5) on both models to get a more robust estimate of their generalization performance. This will help in understanding how the models perform across different subsets of the data, reducing the risk of overfitting.

4. Comparative Analysis

4.1 Classification: Linear vs Non-Linear SVM

In this subsection, you need to compare the performance of **linear** and **non-linear** SVM models for the **Rock Classification Dataset**.



A linear SVM classifier is appropriate when the data is linearly separable or when the relationships between features and classes are simple. It finds a hyperplane that separates the classes with the maximum margin. For this dataset, the linear SVM will work well if the classes can be distinguished using a linear decision boundary. The non-linear SVM, typically using the **Radial Basis Function (RBF)** kernel, is used when the data is not linearly separable. It maps the input data into a higher-dimensional space, where it becomes easier to find a separating hyperplane. For more complex datasets like this one, the RBF kernel can capture intricate patterns in the data that a linear model might miss.

Evaluate both models using accuracy, confusion matrix, and other relevant metrics. The non-linear SVM is expected to perform better on datasets with complex relationships between features and classes, whereas the linear SVM may offer faster training times but could underperform on non-linear problems.

4.2 Regression: Linear vs Non-Linear SVM

Compare the performance of linear and non-linear SVM models for the Theme Park Visitor Count Dataset. The linear SVM is suited for datasets where the relationship between the features and the target variable is approximately linear. It works by finding a linear hyperplane that minimizes the error margin while predicting the visitor count. However, it may struggle to capture more complex relationships in the data. The non-linear SVM, particularly with the **RBF kernel**, is designed to handle complex, non-linear relationships between features and the target. It can better capture intricate patterns in the data, potentially leading to better predictions when the target variable (visitor count) shows non-linear dependencies on the input features.

Evaluate both models using metrics like MAE, MSE, RMSE, and R². While the linear model is faster and easier to interpret, the non-linear model might provide more accurate predictions for complex datasets where a simple linear relationship does not suffice.

5. Submission Guidelines

Prepare a comprehensive report that includes the following:

- Submit a well-documented **Python notebook** with comments explaining each step of your analysis. Use markdown cells to provide context and explanations.
- Summarize your findings, model, details of the training process, and conclusions inside the notebook. Focus on key points, challenges, results, and discussion.
- Include the final **cleaned**, **processed**, and **transformed datasets** with your submission which you used for model training.
- Any challenges you encountered during the entire process and how you overcame them.

Deliverables

- Detailed code along with markdown explanations, visualizations, results, discussion, key findings, and challenges. **(Jupyter / Python Notebook) (code.ipynb)**
- Cleaned, processed, and transformed datasets used for model training. **(Updated CSV Files) (updated_data_classification.csv & updated_data_regression.csv)**

Before submitting, compress both deliverables into a zip file. **(StudentName_012345.zip)**

Make sure to follow the naming convention for both deliverables as well as the final zipped file.

Note: All work submitted must be your own. Adherence to academic integrity is mandatory.

6. Grading Rubric

- Classification Task - **30 marks**
- Regression Task - **30 marks**
- Comparative Analysis (Linear vs Non-Linear SVM) - **15 marks**
- Presentation of Results, Discussion, Explanations - **25 marks**