

# SVM-Based Classification of Microarrays Gene Expression Data



By

**Kashmala Akhtar**

00000365356

Supervisor

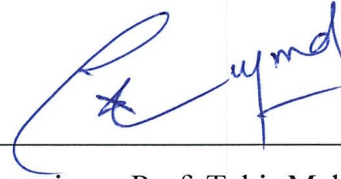
**Dr. Tahir Mehmood**

A thesis submitted in conformity with the requirements for  
the degree of Master of Science in  
Statistics  
Department of Mathematics  
School of Natural Science (SNS)  
National University of Sciences and Technology (NUST)  
Islamabad, Pakistan

## THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS thesis written by **Kashmala Akhtar** (Registration No. **00000365356**), of **School of Natural Sciences** has been vetted by undersigned, found complete in all respects as per NUST statutes/regulations, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS/M.Phil degree. It is further certified that necessary amendments as pointed out by GEC members and external examiner of the scholar have also been incorporated in the said thesis.

Signature: \_\_\_\_\_



Name of Supervisor: Prof. Tahir Mahmood

Date: 30-07-2024

Signature (HoD): \_\_\_\_\_



Date: \_\_\_\_\_

30/7/2024

Signature (Dean/Principal): \_\_\_\_\_




Date: \_\_\_\_\_

31/7/2024

**National University of Sciences & Technology****MS THESIS WORK**

We hereby recommend that the dissertation prepared under our supervision by: "**Kashmala Akhtar**" Regn No. **00000365356** Titled: "**SVM-Based Classification of Microarrays Gene Expression Data**" accepted in partial fulfillment of the requirements for the award of **MS** degree.

**Examination Committee Members**1. Name: DR. MUHAMMAD ASIF FAROOQSignature: 2. Name: DR. SHAKEEL AHMEDSignature: Supervisor's Name: PROF. TAHIR MAHMOODSignature: 

  
Head of Department

30/7/2024  
Date

**COUNTERSIGNED**

Date: 31/7/2024

  
Dean/Principal

# Declaration

I, Kashmala Akhtar declare that this thesis titled “SVM-Based Classification of Microarrays Gene Expression Data” and the work presented in it are my own and has been generated by me as a result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a Master of Science degree at NUST
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at NUST or any other institution, this has been clearly stated
3. Where I have consulted the published work of others, this is always clearly attributed
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work
5. I have acknowledged all main sources of help
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself

---

Kashmala Akhtar,  
NUST00000365356SNS

# Copyright Notice

- Copyright in text of this thesis rests with the student author. Copies (by any process) either in full, or of extracts, may be made only in accordance with instructions given by the author and lodged in the Library of SNS, NUST. Details may be obtained by the Librarian. This page must form part of any such copies made. Further copies (by any process) may not be made without the permission (in writing) of the author.
- The ownership of any intellectual property rights which may be described in this thesis is vested in SNS, NUST, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of SNS, which will prescribe the terms and conditions of any such agreement.
- Further information on the conditions under which disclosures and exploitation may take place is available from the Library of SNS, NUST, Islamabad.

# Dedication

This thesis is dedicated to *my beloved husband*

# Abstract

Classifying microarray gene expression data is crucial due to its high-dimensional nature and its significant impact on disease diagnosis and personalized treatment strategies. Timely and accurate classification of gene expression data greatly influences treatment outcomes and patient survival rates. Traditionally, gene expression data analysis involves various statistical methods. However, with the emergence of advanced machine learning techniques, automated classification within these datasets becomes crucial. Present methodology typically involve SVM classifier with different kernel functions to classify diverse gene expression profiles. Nonetheless, the varied characteristics within gene expression data present notable classification challenges.

In our study, we introduce a comprehensive dataset comprising thousands of gene expression profiles from Leukemia cancer. Our approach involves proposing an optimal classification method by fine-tuning Support Vector Machine (SVM) parameters and selecting the most appropriate kernel functions. We utilize both standard and refined SVMs with various kernel functions, including linear, polynomial, radial basis function (RBF), and sigmoid, alongside penalized SVM models using  $L_1$ , Smoothly Clipped Absolute Deviation (SCAD), and  $SCAD + L_2$  penalties to improve classification performance.

Notably, our innovative approach, when applied to refined SVM with linear and polynomial kernels, achieves superior performance, with the  $L_1$  norm exhibiting the best classification accuracy among penalized models. This breakthrough marks a significant advancement in gene expression data classification literature, highlighting the potential of SVMs, particularly with linear and polynomial kernels combined with appropriate penalty terms, for precise and efficient disease classification.

# Acknowledgments

First and foremost, I express my profound gratitude to Allah Almighty (SWT) for His countless blessings. Following this, I am deeply grateful to my supervisor, Dr. Tahir Mehmood, for his invaluable guidance, encouragement, dedication, and patience. His support and constructive feedback were instrumental in bringing this thesis to fruition.

The authors extend their heartfelt thanks to the members of my evaluation committee and guidance committee, including Dr. Asif Farooq and Dr. Shakeel Ahmad, for their generous time, insightful suggestions and critical assessments. Their contributions have been essential to the development and refinement of this work.

Lastly, I owe a special debt of gratitude to my husband and family for their boundless love, kindness, and encouragement. Their prayers and constant support have been crucial in enabling me to achieve this significant milestone.ewpage



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Classification . . . . .	1
1.2	Support Vector Machine . . . . .	2
1.2.1	Types of Support Vector Machine . . . . .	2
1.2.2	Hyperplane And Support Vectors In SVM . . . . .	3
1.2.3	Why Support Vector Machine . . . . .	4
1.3	Applications Of Support Vector Machine . . . . .	4
1.4	Introduction Of Micro-arrays Gene Expression Data . . . . .	5
1.5	Objective Of Research Study . . . . .	8
1.6	Organization Of The Study . . . . .	8
<b>2</b>	<b>Literature Review</b>	<b>9</b>
<b>3</b>	<b>Methodology</b>	<b>11</b>
3.1	Data Acquisition . . . . .	11
3.2	Linearly Separable Binary Classification . . . . .	12
3.2.1	Lagrange Multipliers . . . . .	14
3.2.2	Dual Problem . . . . .	15
3.3	Linearly Non-Separable Data . . . . .	16
3.4	Kernels . . . . .	19
3.4.1	Non-Linear Data . . . . .	19

## CONTENTS

3.4.2	Linear Kernel . . . . .	19
3.4.3	Non-Linear Kernels . . . . .	20
3.4.4	Polynomial Kernels . . . . .	20
3.4.5	Radial Basis Function . . . . .	21
3.4.6	Sigmoid Kernel . . . . .	22
3.5	Penalized Support Vector Machine . . . . .	22
3.5.1	Ridge Penalty . . . . .	23
3.5.2	Least Absolute Shrinkage And Selection Operator Penalty . . . . .	24
3.5.3	Smoothly Clipped Absolute Deviation Penalty . . . . .	24
3.5.4	Elastic Smoothly Clipped Absolute Deviation Penalty . . . . .	25
3.6	Evaluation . . . . .	26
<b>4</b>	<b>Results And Discussion</b>	<b>30</b>
<b>5</b>	<b>Conclusion</b>	<b>45</b>
	<b>References</b>	<b>45</b>

# List of Figures

1.1	The figure illustrates hyperplane and support vectors . . . . .	3
1.2	Structure Of DNA . . . . .	6
3.1	Hyperplane through linearly separable data . . . . .	13
3.2	Enter Caption . . . . .	17
3.3	Hyperplane through Non-linearly separable data . . . . .	19
3.4	The figure illustrate confusion matrix . . . . .	27
4.1	This figure illustrates the train accuracy of different variants of SVM . . . . .	30
4.2	This figure illustrates the test accuracy of different variants of SVM . . . . .	31
4.3	This figure illustrates the train sensitivity of different variants of SVM . . . . .	32
4.4	This figure demonstrates test sensitivity of different variants of SVM . . . . .	33
4.5	This figure illustrate train specificity of different variants of SVM . . . . .	34
4.6	This figure illustrate the test specificity of different variants of SVM . . . . .	34
4.7	Post-hoc comparison for train standard SVM . . . . .	36
4.8	Post-Hoc comparison for test standard SVM . . . . .	37
4.9	Post-Hoc comparison for train refined SVM . . . . .	39
4.10	Post-Hoc comparison for test Refined SVM . . . . .	41
4.11	Post-Hoc comparison for train penalized SVM . . . . .	43
4.12	Post-Hoc comparison for test Penalized SVM . . . . .	44

# List of Tables

3.1	Details Of Gene Expression Data-set . . . . .	12
4.1	ANOVA Table for train accuracy for standard SVM . . . . .	35
4.2	ANOVA Table for train sensivity for standard SVM . . . . .	35
4.3	ANOVA Table for train specificity for standard SVM . . . . .	35
4.4	ANOVA Table for test accuracy for standard SVM . . . . .	36
4.5	ANOVA Table for test sensivity for standard SVM . . . . .	36
4.6	ANOVA Table for test specificity for standard SVM . . . . .	37
4.7	ANOVA Table for train accuracy for refined SVM . . . . .	38
4.8	ANOVA Table for train sensivity for refined SVM . . . . .	38
4.9	ANOVA Table for train specificity for refines SVM . . . . .	38
4.10	ANOVA Table for Test Accuracy for Refined SVM . . . . .	40
4.11	ANOVA Table for Test Sensivity for Refined SVM . . . . .	40
4.12	ANOVA Table for test specificity for refined SVM . . . . .	41
4.13	ANOVA Table for train accuracy for penalized SVM . . . . .	42
4.14	ANOVA Table for train sensivity for penalized SVM . . . . .	42
4.15	ANOVA Table for train specificity for penalized SVM . . . . .	43
4.16	ANOVA Table for test specificity for penalized SVM . . . . .	43

# List of Abbreviations and Symbols

## Abbreviations

<b>SVM</b>	Support Vector Machine
<b>DNA</b>	deoxyribonucleic acid
<b>A</b>	Adenine
<b>G</b>	Guanine
<b>C</b>	Cytosine
<b>T</b>	Thymine
<b>RNA</b>	ribonucleic acid
<b>mRNA</b>	messenger ribonucleic acid
<b>tRNA</b>	transfer ribonucleic acid
<b>rRNA</b>	ribosomal ribonucleic acid
<b>KKT</b>	Karush-Kuhn-Tucker
<b>RBF</b>	Radial Basis Function
<b>LASSO</b>	Least Absolute Shrinkage and Selection Operator
<b>SCAD</b>	Smoothly Clipped Absolute Deviation
<b>SQA</b>	Successive Quadratic Algorithm

LIST OF TABLES

<b>HSD</b>	Honest Significant Difference
<b>AML</b>	Acute myeloid leukaemia
<b>ALL</b>	Acute lymphocytic leukaemia
<b>IQR</b>	Interquartile Range
<b>LDA</b>	Linear Discriminant Analysis

# Introduction

## 1.1 Classification

Classification is a kind of statistical learning, they learn from previous experiences and the target variable is a category/digital variable that is used to categorize the input features. In classification, the dependent feature is the categorical variable, that is, the target variable is divided into a finite number of categories; the purpose of the classification algorithm is to model the connection between the feature vector and the appropriate category. The construction phase of a classification model includes several steps:

1. Data Preparation: The first is to handle the input data that will be classified with the help of the K-nearest neighbor algorithms. This will refer to the process of choosing the appropriate input variables as well as pre-processing of the dataset to the required format.
2. Model Learning: Subsequently, the algorithm is fine-tuned with the help of a labeled set, each record of which is associated with the required class label. Dwelling a little more on the algorithm training process, the training process involves learning of the relations between the independent features and the class labels.
3. Model Evaluation: After the model has been developed, the model is tested on a dataset that is different from the training dataset in order to determine how well the model works. This evaluation entails determining parameters such as accuracy, specificity as well as the sensitivity.

4. Model deployment: Once the model has been trained, then it is ready to use when predicting other unseen data.

## 1.2 Support Vector Machine

A Support Vector Machine (SVM) is considered a type of supervised statistical learning that is used primarily for classification purposes. Its fundamental principle involves identifying a hyperplane (a line or plane in a complex space) that optimally segregates the dataset into unique classes. This decision boundary is positioned to maximize the margin, defined as the separation space between the decision boundary and the nearest data values from each class. These critical data points, nearest to the decision boundary, are termed support vectors and play a pivotal role in delineating the decision boundary of the SVM. Subsequently, the classification of new data points becomes straightforward: they are projected onto the hyperplane, and their position relative to the hyperplane determines their class. [1]

### 1.2.1 Types of Support Vector Machine

There are two main types of Support Vector Machine (SVM):

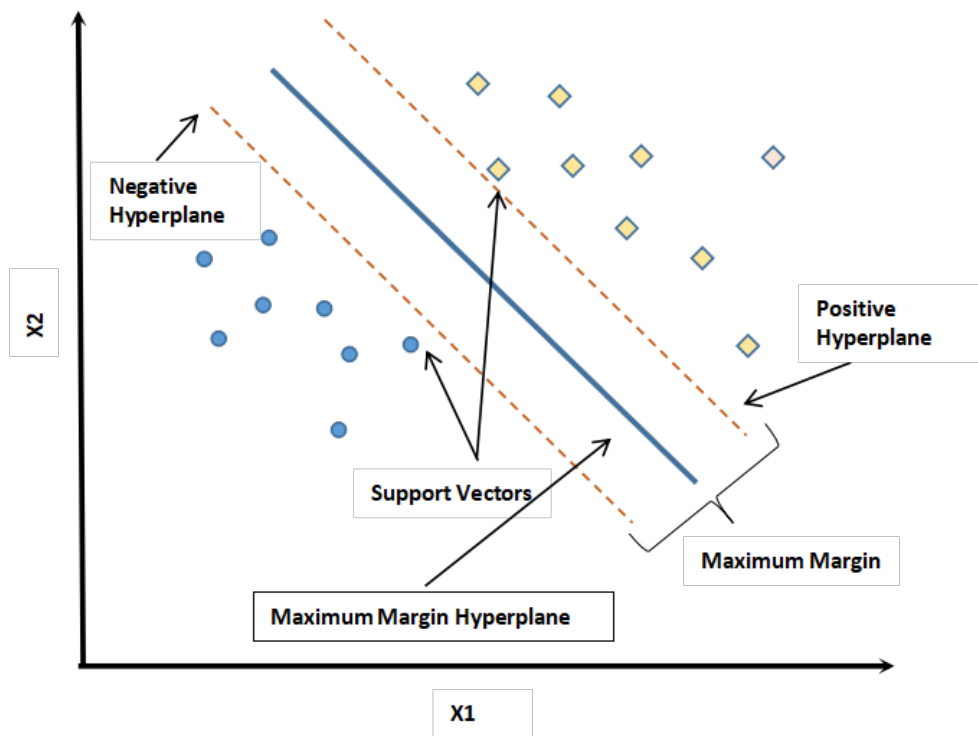
- Maximum margin classifier (Hard margin SVM): This type of SVM is used for problems with linearly separable data, where data points can be separated by a clear margin. Hard margin SVM is capable of identifying the maximum margins that exist between the points in the original space and create a hyperplane that will act as a separator for the data. However, this method can be sensitive to outliers and may not perform well on noisy or poorly scaled data.
- Soft Margin Classifier (Soft Margin SVM): This particular variant of SVM is tailored for datasets that aren't linearly separable, acknowledging the presence of misclassifications among data points. Unlike the hard margin SVM, the soft margin approach permits a degree of misclassification, prioritizing the maximization of the margin between data points. This adaptation enhances resilience against outliers and noisy data, ensuring a more robust classification model.



### 1.2.2 Hyperplane And Support Vectors In SVM

In Support Vector Machine, the hyperplane is a plane which provides decision boundary line between different classes of data values. The purpose is to locate the hyperplane with an optimum distance from the nearest data points in the provided dataset, where this distance is called the margin.

The points in the close proximity to the hyperplane are called the support vectors. These support vectors define the position of the hyperplane and have significant importance in decision boundary creation. That is, the position of the hyperplane is fully dependent on the support vectors and any change with the support vectors will lead to a change in the hyperplane.



**Figure 1.1:** The figure illustrates hyperplane and support vectors

If it is possible to classify data with a hard margin classifier (Maximum margin classifier or SVM), then the hyperplane is chosen such that it has the maximum distance from the data points in the training data set which gives maximum generalization and no misclassifications due to outside noise. While in the case of a soft margin classifier (Soft margin SVM), the hyperplane formed allows for some amount of misclassification

making it ideal for use in non-linearly separable data.

### 1.2.3 Why Support Vector Machine

Support Vector Machines (SVMs) are popular because they can be used for a variety of tasks, including classification and regression, and often produce accurate results. Some reasons why SVMs are a good choice of algorithm include:

- SVMs can linear or non-linear boundaries, and this is in agreement of the kernel trick which enables the model to modify the input data values to a complex dimensional space where linear boundaries can be identified.
- Most of the SVMs are applicable in very high dimensions or when the number of attributes is much larger than the number of instances.
- SVMs are also memory efficient as they employ only a few training instances that are the support vectors to define the decision boundary.
- As with many machine learning algorithms, SVMs are general-purpose algorithms that can be used for a number of different tasks including text and image classification, bioinformatics, and face detection.
- While working in the high-dimensional space, the SVMs are significantly less sensitive to overfitting than the other models.

Overall, the SVM algorithm is a strong used tool for machine learning, and it's particularly useful when the data is large, high-dimensional and complex.

## 1.3 Applications Of Support Vector Machine

Support Vector Machines (SVMs) have a huge range of uses in different fields such as natural language processing, image classification, bio-informatics, and financial forecasting. Some specific examples include:

- Text classification: SVMs can categorize documents into various groups, such as distinguishing between spam and non-spam emails.

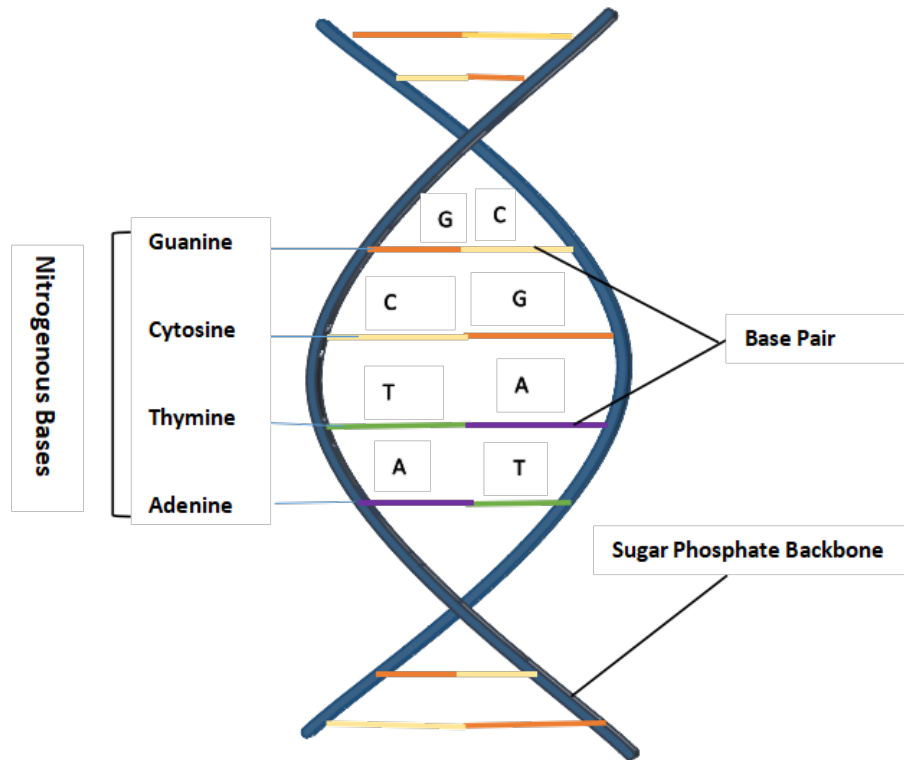
- Image classification: SVMs can be utilize to classify images into different categories, such as identifying objects in an image.
- Handwriting recognition: SVMs can be used to recognize handwriting in scanned documents.
- Bio-Informatics: SVMs can be used to predict the classification of proteins into different categories such as disease-causing or non-disease-causing.
- Financial forecasting: SVMs can be used to predict stock prices based on historical financial data.
- In medical field, SVMs can be used in cancer classification from biopsy images.

## 1.4 Introduction Of Micro-arrays Gene Expression Data

Cancer research is a significant area of investigation in the medical field. Accurately predicting various types of tumors can greatly improve treatment and reduce the toxicity for patients. Traditional cancer classification methods have relied on morphology and clinical characteristics, but they have limitations in their diagnostic ability. [2]. Experts suggest that treating tumors according to their pathogenic patterns could improve efficacy [3]. Furthermore, contemporary tumor classifications exhibit heterogeneity, characterized by molecularly distinct diseases that manifest diverse clinical trajectories. To enhance our comprehension of cancer classification, systematic methodologies employing comprehensive gene expression analysis. Gene expression levels contain valuable information for the prohibited and treatment of sickness, understanding the mechanisms of biological evolution, and discovering new drugs. With the emergence of microarray technology, it is now possible to monitor thousands of genes simultaneously, leading to the development of cancer classification methods based on gene expression data [4].

All living systems are based on cells as their fundamental working units, and the instructions that guide these cells are encoded in deoxyribonucleic acid (DNA). DNA is composed of four nucleotide, each containing a phosphate group, a deoxyribose sugar, and one of four nitrogen bases: adenine (A), guanine (G), cytosine (C), and thymine (T). These nitrogen bases form base pairs, with A always pairing with T and C always pairing with G, holding the two strands of the double helix structure together through

hydrogen bonds. The two strands of the DNA double helix are like chemical "mirror images" of each other, with each nucleotide on one strand corresponding to its complement on the other. An organism's entire collection of DNA, which encodes all of its genetic



**Figure 1.2:** Structure Of DNA

information, is referred to as its genome. The size of genomes can vary greatly; for example, the smallest known genome for a free-living organism (a bacterium) consists of approximately 600,000 DNA base pairs, whereas the human and mouse genomes consist of around 3 billion DNA base pairs. Except for mature red blood cells, Every cell in the human body possesses a full complement of DNA, constituting a complete genome. The genome isn't a single continuous sequence, but rather a collection of genes that are organized in a specific manner.

Here's a rewritten version of your text that avoids plagiarism while retaining the original meaning:

—

Gene expression involves the process of converting the DNA sequence of a gene into

RNA. The extent of gene expression is directly proportional to the quantity of RNA molecules produced within a cell, which in turn affects the synthesis of specific proteins. Distinct patterns of gene expression emerge in various biological contexts—such as embryonic development, cellular differentiation, and physiological responses under normal conditions—which provide insights into a gene’s activity in specific biochemical environments. Alterations in gene expression are often observed in diseases, including cancer, and can be linked to mutations in genes that control the cell cycle, apoptosis, genomic stability, and other critical factors.

Recent advancements have shown that DNA microarray technology is effective for analyzing gene expression patterns, aiding in understanding gene functions and detecting cancer. The cDNA microarray technique, a modern advancement over traditional probe hybridization methods, enables simultaneous analysis of thousands of genes, allowing comprehensive recording of gene states. A cDNA microarray consists of numerous distinct DNA sequences, each represented as a high-density spot on a glass slide. During experiments to compare gene expression levels under different conditions, each data point reflects the ratio of expression levels for a specific gene. Specifically, if  $m$  represents the number of genes on a microarray chip, the experimental results provide a set of ratios where the numerator is the gene expression level under changing conditions, and the denominator is the expression level in a reference condition.

The classification of gene expression data has garnered significant attention from researchers in fields such as statistics, machine learning, and data science. Various classification algorithms have been developed over time, with recent studies focusing on cancer classification using gene expression profiles. Differences in gene expression patterns are linked to various cancer types. Classification methods, ranging from traditional nearest neighbor approaches to advanced support vector machines (SVMs), have been explored. No single classifier outperforms all others universally; some are suited for binary classification, while others are more versatile. Additionally, many gene classification algorithms prioritize accuracy but may overlook the computational demands, which can be substantial given the nature of gene expression data.

This survey aims to provide a detailed examination of cancer classification challenges to develop more effective and efficient classification algorithms. [5]

## 1.5 Objective Of Research Study

Followings are the main objectives of research:

- Understanding how SVM works with kernels and with different penalties.
- Using the SVM to classify gene expression data.
- Comparison of different variants of SVM.

## 1.6 Organization Of The Study

The research begins by providing a brief overview of classification using the support vector machine algorithm, microarray gene expression data. Section 2 presents a review of the literature. In Section 3, the mathematical description of linearly separable and non-linearly separable data, kernels, and penalized SVM is discussed. Chapter 4 is dedicated to the results and discussion of all the proposed methods. In addition, in Chapter 5 we provide a conclusion of the research work.

## CHAPTER 2

# Literature Review

The study by Peng Guan et al. aimed to incorporate previous knowledge into the analysis of the lung cancer gene expression database to improve the accuracy of the cancer classification. The researchers used a machine-based classification method for support vectors and a public gene expression data set to test their modified strategy. The results illustrated that the incorporation of previous knowledge improved the classification accuracy from 98.86% to 100% in the training set and the accuracy of the test set improved from 98.51% to 99.06%. Additionally, the standard deviations of the modified method significantly decreased from 0.26% to 0% in the training set and from 3.04% to 2.10% in the test set[6]. The paper by Guyon et al. presents a method to select informative genes for cancer classification using SVM. The authors applied their method to microarray datasets of cancer gene expression profiles and found that their method outperformed other gene selection methods in terms of classification accuracy and reduced the number of genes required for analysis. The selected genes were found to be significantly associated with the respective cancers and had potential diagnostic and therapeutic implications. However, further validation on larger and diverse datasets is needed to confirm the robustness and generalizability of the proposed method [7]. The paper by H. Zhang et al. presents a method for gene selection using SVM with a non-convex penalty function. The authors applied their method to two microarray data sets and found that it outperformed other gene selection methods in terms of classification accuracy and reduced the number of genes selected. The selected genes were significantly associated with the respective cancers and had potential diagnostic and therapeutic implications. The authors further evaluated the robustness and computa-

tional efficiency of their method and found consistent and efficient results. Overall, the study demonstrates the potential of utilizing SVM with a non-convex constraint for gene identification in microarray data analysis with clinical implications for disease diagnosis and treatment.[8]

The study proposed a method for detecting and classifying melanoma skin cancer using support vector machines (SVM) and different texture features extracted from images of skin lesions. The proposed SVM-based method achieved high accuracy, sensitivity, and specificity in both datasets used for evaluation, demonstrating the potential of using SVM for skin cancer detection and classification. The findings showed that the introduced SVM-based method achieved high accuracy, sensitivity, and specificity in both data sets. Specifically, on the UCSF dataset, the proposed method achieved an accuracy of 96.89%, a sensitivity of 96.19%, and a specificity of 98.00%. In the ISIC data set, the proposed method achieved an accuracy of 83.43%, a sensitivity of 82.73%, and a specificity of 85.13%. The study demonstrates the potential of using SVM for skin cancer detection and classification. The proposed method achieved high precision and can help improve the early detection and diagnosis of melanoma skin cancer, which can have significant clinical implications for patient outcomes. However, it is vital to note that the capability of the method can vary depending on the specific data set and characteristics used for the analysis[9] Gopinath and Shanthi (2013) developed a support vector machine (SVM)-diagnostic system aimed at detecting thyroid cancer using statistical texture characteristics. Their study focused on leveraging advanced machine learning techniques to improve the precision and efficiency of thyroid cancer detection. By analyzing statistical texture features extracted from medical images, the SVM model demonstrated promising results in differentiating between cancerous and non-cancerous thyroid tissues. This research contributes to ongoing efforts to develop more effective and precise diagnostic tools for thyroid cancer, which may improve patient outcomes through early detection and treatment intervention.[10]



# Methodology

This study provides an overview of the data set used and describes our approach to integrating support vector machines and its variants into the classification task. The methodology section delves into the steps required for SVM creation and training, with the selected dataset playing a pivotal role in our research foundation.

## 3.1 Data Acquisition

In this thesis, the gene expression of the leukemia cancer data set was classified into its two groups, acute myeloid (or myelogenous) leukemia (AML) and acute lymphocytic (or lymphoblastic) leukemia (ALL). We present the results of our framework that would demonstrate the comparison between different versions of support vector machine. The programming and analysis of SVM is processed in R software.

The data set was taken to classify leukemia using microarray gene expression obtained from the link

<https://file.biomedcentral.com/suppl/10.1186/s12859-015-0681-7> 72 samples were included in the data used to categorize the data for this thesis. There were 47 labels labeled "1" indicating ALL type and 25 with "2" indicating AML type. Each sample originally measured 6817 probes set, but we removed genes not present in at least one sample. So our data set contains 72 samples and 5,147 genes. The objective is to distinguish between ALL and AML samples.

In the context of SVM modeling, the use of the kernlab package in the R programming language introduces the concept of refined SVMs. Unlike traditional SVM implementa-

tions, refined SVMs undergo an enhanced modeling process characterized by meticulous parameter tuning and optimization techniques. This refinement process, facilitated by the flexibility of the kernlab package, enables practitioners to explore a diverse range of kernel functions and hyperparameter configurations tailored to the specific characteristics of the dataset. Through careful selection of kernel functions and optimization of hyper-parameters, refined SVMs aim to capture intricate patterns within the data while mitigating overfitting and enhancing generalization performance. The incorporation of advanced techniques such as cross-validation further ensures robust model evaluation and validation. By referring to SVM models developed using kernlab as "refined SVMs," researchers acknowledge the sophisticated modeling approach undertaken to maximize predictive performance and extract meaningful insights from the data. This distinction underscores the rigor and depth of the SVM modeling methodology used in the study, improving the credibility and validity of the research findings.

Data set	Total samples	No. of Genes	Class Label	Class-wise Sample
Leukemia	72	5148	ALL	47
			AML	25

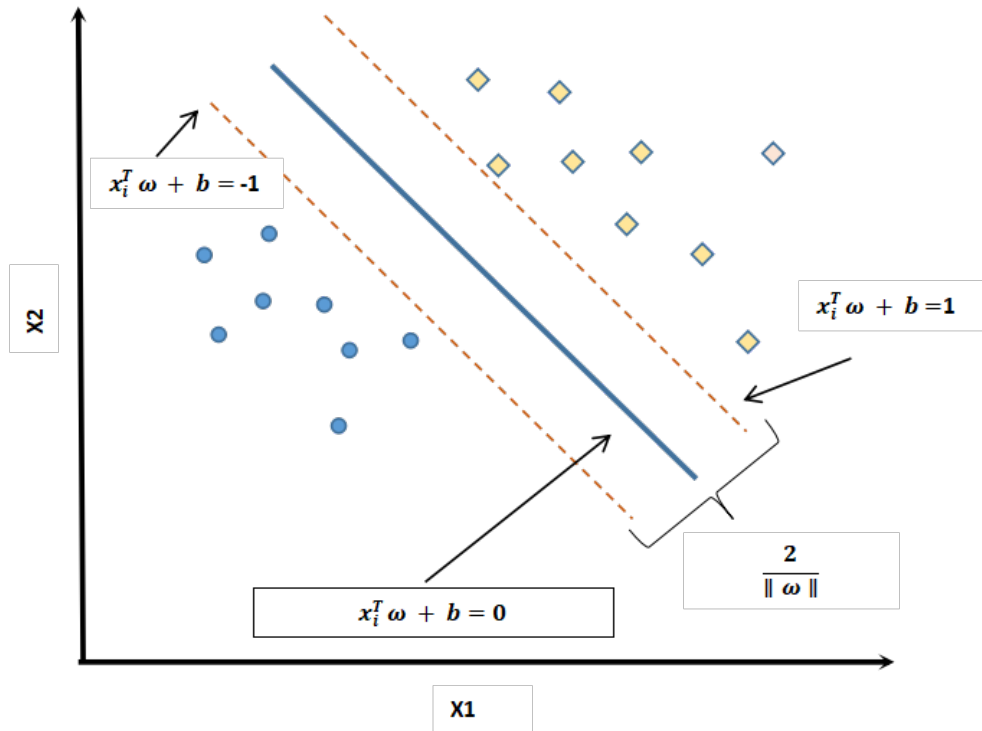
**Table 3.1:** Details Of Gene Expression Data-set

## 3.2 Linearly Separable Binary Classification

Suppose training data includes of  $L$  points, where each point is represented by an input vector  $u_k$  with  $D$  attributes and is classified as either  $v_k = -1$  or  $v_k = 1$ . The data is assumed to be linearly separable, meaning that a boundary (a line in 2D and a hyperplane in higher dimensions) can be drawn to separate the two classes. In other words, the dataset is in form of  $u_k, v_k$  where  $i = 1, \dots, L$ ,  $v_k$  is either  $-1$  or  $1$ , and  $u_k$  is a  $D$ -dimensional vector in the real space. The hyperplane that separates the two classes can be represented by an equation of the form,

$$u_k^T \psi + b = 0, \quad k = 1, \dots, L \quad (3.2.1)$$

In this equation, The vector  $\psi$  is orthogonal to the hyperplane, and  $b$  represents the distance from the hyperplane to the origin along the normal vector  $\psi$ . We have to find a good value of  $\psi$  and  $b$  such that we maximize its margin



**Figure 3.1:** Hyperplane through linearly separable data

From the figure 3.1 lets, we have two predictors  $x_1$  and  $x_2$ , these are also known as feature vectors. The diamonds are classified as class +1 and blue circles are marked as class -1. Now in the spirit of SVM we want to create a maximum margin classifier which means we want to choose a decision boundary (that middle black line is the decision boundary) such that the space around the decision boundary which is called the margin as big as possible. Equation 3.2.1 is known as equation of hyperplane and can be written as

$$\psi_1.u_1 + \psi_2.u_2 + \dots + \psi_p.u_p + b = 0$$

Where  $b$  is the intercept and  $\psi_1, \psi_2, \dots, \psi_p$  are the coefficients, as we are dealing with 2-D our hyperplane is a straight line. So  $\psi$  and  $b$  are only two coefficients of concern. We have to find good values of  $\psi$  and  $b$  such that it maximize the margin.

If  $\vec{u}$  is on the decision boundary, then it must follow the equation of hyperplane  $u_k^T \psi + b = 0$  to find how many units we have to walk in the  $\psi$  direction in order to reach the other line  $u_k^T * \psi + b = 1$ . First, make a unit vector in the direction of  $\psi$ . If we are at our

vector  $\vec{u}$  and we walk  $k$  unit in the direction of  $\psi$  then we will be at  $u_k^T * \psi + b = 1$ . So

$$\vec{\psi} \cdot \left( \vec{u} + k \frac{\vec{\psi}}{\|\vec{\psi}\|} \right) + b = 1$$

$$\vec{\psi} \cdot \vec{u} + k \frac{\vec{\psi} \cdot \vec{\psi}}{\|\vec{\psi}\|} + b = 1$$

$$k = \frac{\|\vec{\psi}\|}{\vec{\psi} \cdot \vec{\psi}}$$

$$k = \frac{1}{\|\vec{\psi}\|}$$

So, margin size is  $k = \frac{2}{\|\vec{\psi}\|}$ . So, to maximize the margin, all we have to do is minimize the magnitude of  $\psi$ .

Here we have a couple of constraints:

$$\text{If } v_k = 1 \text{ then } u_k^T \psi + b \geq 1 \quad (3.2.2)$$

$$\text{If } v_k = -1 \text{ then } u_k^T \psi + b \leq 1.$$

From above two constraints we can write  $v_k(u_k^T \psi + b) \geq 1 \forall k = 1, 2, \dots, L$  So optimization problem will become

$$\min_{w,b} \|\psi\| \quad \text{Subject to } v_k(u_k^T \psi + b) \geq 1 \quad \forall k = 1, \dots, L$$

Minimizing  $\|\psi\|$  is equal to minimizing,  $\frac{1}{2}\|\vec{\psi}\|^2$  which make above equation a quadratic programming optimization. So we have

$$\min_{\psi,b} \frac{1}{2}\|\vec{\psi}\|^2 \quad \text{Subject to } v_k(u_k^T \psi + b) - 1 \geq 0 \quad \forall k = 1, \dots, L \quad (3.2.3)$$

And the equation for support vector will be  $v_k(u_k^T \psi + b) = 1$

### 3.2.1 Lagrange Multipliers

The general Lagrangian function can be written as:

$$L(\psi, b, \alpha_k) = \frac{1}{2} \psi^t \psi - \sum_{i=1}^L \alpha_k (v_k (u_k^T \psi + b) - 1) \quad \forall k = 1, \dots, L$$

$$L(\psi, b, \alpha_k) = \frac{1}{2} \psi^t \psi - \sum_{i=1}^L \alpha_k v_k u_k^T \psi - \sum_{i=1}^L v_k \alpha_k b + \sum_{i=1}^L \alpha_k$$

where  $\alpha'_k$ s are the Lagrange multipliers.

The objective of optimization is to determine the values of  $\psi$ ,  $b$ , and  $\alpha$  that maximize the margin while adhering to the constraints  $\alpha_k \geq 0$  and  $v_k(u_k^T \psi + b) \geq 1$ . This is achieved by differentiating the Lagrangian  $L$  from  $\psi$  and  $b$  and setting the derivatives to zero.

$$\frac{\partial L}{\partial \psi} = 0$$

$$\psi = \sum_{i=1}^L \alpha_k u_k^T v_k$$

Similarly,

$$\frac{\partial L}{\partial b} = 0$$

$$\alpha^T v = 0$$

The only samples in our data that are contributing to the definition of the margin itself are the support vectors and since  $\psi$  is the part of the definition of margin itself the only vectors in our data that are allowed to contribute to the definition of  $w$  can be support vectors themselves. For any  $u_k$  that are not support vectors, their  $\alpha_k$  must be equal to zero, which means that the Lagrangian follows the KKT (Karush-Kuhn-Tucker) condition. So, Lagrangian become

$$L(\alpha_k) = \sum_{k=1}^L \alpha_k - \frac{1}{2} \sum_{k,j} \alpha_k \alpha_j u_k^T v_k v_j, \quad \alpha^T v = 0, \quad \forall k$$

So because of KKT conditions the Lagrangian can be written in the form of Dual function.

### 3.2.2 Dual Problem

The dual problem is:

$$\max_{\alpha_k \geq 0} [\min_{\psi, b} L(\psi, b, \alpha)]$$

$$\max_{\alpha_k \geq 0} \left[ \sum_{k=1}^L \alpha_k - \frac{1}{2} \sum_{k,j} \alpha_k \alpha_j u_k^T v_k v_j, \quad \alpha^T v = 0, \quad \forall k \right]$$

And this can be reduced to:

$$\min_{\alpha_k \geq 0} \left[ \frac{1}{2} \sum_{k,j} \alpha_k \alpha_j u_k^T u_j v_k v_j - \sum_{k=1}^L \alpha_k, \quad \forall k \right]$$

We know that for all non-support vectors,  $\alpha_k$ 's will be zero from the KKT Conditions. We just need to calculate the inner product between input vector  $x$  and support vectors

because many terms of  $\alpha_k v_k \langle u_k, u_j \rangle$  will be zero. Efficiency in prediction is one of the advantages that dual problems provide us.

Additionally, the only action on the input  $u$  for prediction is to compute the inner product of  $u$  and the support vectors  $\langle u_k, u_j \rangle$ . As a result, it enables us to enlarge the feature space using kernel tricks without explicitly accessing higher-dimensional feature space. Another significant advantage of the dual form is that we may enlarge the feature space without paying extra for it to do so during training and prediction.

### 3.3 Linearly Non-Separable Data

For the incomplete linearly separable data, it becomes quite a challenge to obtain an appropriate boundary for the two types of data. In this situation, one frequently uses a support vector machine (SVM) and performs the task of soft-margin classification. A soft-margin SVM allows some misclassifications while still finding the best hyperplane that separates the data. This is done by adding a slack variable  $\zeta_k$  to the equation 3.2.2,  $\forall k = 1, 2, \dots, L$

$$\text{If } v_k = 1 \text{ then } u_k^T \psi + b \geq +1 - \zeta_k \quad (3.3.1)$$

$$\text{If } v_k = -1 \text{ then } u_k^T \psi + b \geq -1 + \zeta_k \quad (3.3.2)$$

$$\zeta_k \geq 0 \quad \forall k \quad (3.3.3)$$

Which can be combined into

$$v_k(u_k^T \psi + b) \geq +1 - \zeta_k \quad \text{where } \forall k \quad (3.3.4)$$

In the soft margin Support Vector Machine (SVM), there is a penalty imposed on data points that are located on the wrong side of the margin boundary, and this penalty increases as the distance from the boundary increases. Our objective is to reduce the number of misclassification, so objective function 3.2.3 from recent will be:

$$\min \frac{1}{2} \|\psi\|^2 + C \sum_{k=1}^L \zeta_k \quad \text{s.t.} \quad v_k(u_k^T \psi + b) - 1 + \zeta_k \geq 0 \quad \forall k \quad (3.3.5)$$

where  $C$  is regularization parameter which controls the size of margin and slack variable penalty. Reformulate as a Lagrangian where we have to minimize with respect to  $\psi$ ,  $b$ ,  $\zeta_k$  and maximize the  $\alpha_k$ . That is,

$$L = \frac{1}{2} \|\psi\|^2 + C \sum_{k=1}^L \zeta_k - \sum_{k=1}^L \alpha_k [v_k(u_k^T \psi + b) - 1 + \zeta_k] - \sum_{k=1}^L \mu_k \zeta_k \quad (3.3.6)$$

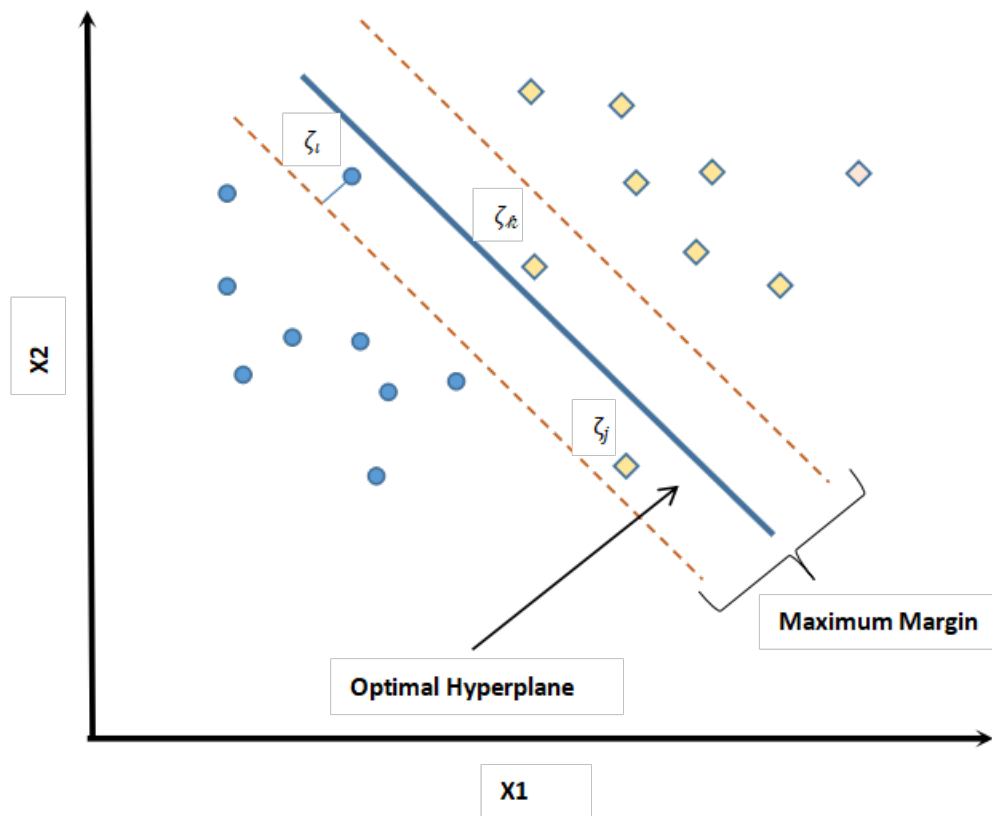


Figure 3.2: Enter Caption

Differentiation with respect to  $\psi$ ,  $b$ ,  $\zeta_k$  and equate it to zero.

$$\frac{\partial L}{\partial \psi} = 0 \quad \psi = \sum_{k=1}^L \alpha_k v_k u_k \quad (3.3.7)$$

$$\frac{\partial L}{\partial b} = 0 \quad \alpha^T v = 0 \quad (3.3.8)$$

$$\frac{\partial L}{\partial \zeta} = 0 \quad C = \alpha_k + \mu_k \quad (3.3.9)$$

So equation 3.3.6 will become

$$\begin{aligned} L = & \frac{1}{2} \sum_{k,j} \alpha_k \alpha_j v_k v_j u_k^T u_j + \sum_{k=1}^L \alpha_k \zeta_k + \sum_{k=1}^L \mu_k \zeta_k - \sum_{k,j} \alpha_k \alpha_j v_k v_j u_k^T u_j \\ & - \sum_{k=1}^L \alpha_k v_k b + \sum_{k=1}^L \alpha_k - \sum_{k=1}^L \alpha_k \zeta_k - \sum_{k=1}^L \mu_k \zeta_k \end{aligned} \quad (3.3.10)$$

$$L = -\frac{1}{2} \sum_{k,j} \alpha_k \alpha_j v_k v_j u_k^T u_j + \sum_{k=1}^L \alpha_k \quad \text{And} \quad \sum_{k=1}^L \alpha_k v_k = 0 \quad (3.3.11)$$

The reduced dual optimization problem is

$$\min_{\alpha_k \geq 0} \left[ \frac{1}{2} \sum_{k,j} \alpha_k \alpha_j v_k v_j u_k^T u_j - \sum_{k=1}^L \alpha_k \right] \quad (3.3.12)$$

There is no slack variables  $\zeta_k$  appears in quadratic problem function.

In case of SVM without noise,

$$\begin{aligned} \alpha_k & \geq 0, \quad i = 1, \dots, L \\ \sum_{k=1}^L \alpha_k v_k & = 0 \end{aligned} \quad (3.3.13)$$

In case of SVM with noise,

$$\begin{aligned} 0 & \leq \alpha_k \leq C, \quad i = 1, \dots, L \\ \sum_{k=1}^L \alpha_k v_k & = 0 \end{aligned} \quad (3.3.14)$$

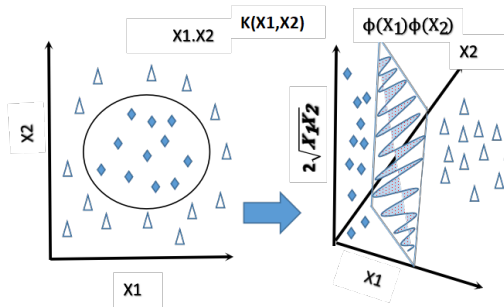
In a dual formulation of the support vector machine (SVM), the objective function remains unchanged, but the constraints may differ depending on whether noise is taken into account. If there is noise, the value of SVM  $\alpha_k$  lies between 0 and C. The SVM classifier that incorporates noise is often referred to as a soft SVM due to its lack of a rigid decision boundary separating the different classes of data points. Soft margin classification is characterized by the identification of support vectors. The support vectors are the data points  $u_k$  that have non-zero Lagrangian multipliers  $\alpha_k$  associated with them.[11]



### 3.4 Kernels

#### 3.4.1 Non-Linear Data

When dealing with non-linearly separable data, SVM employs the use of kernels. A kernel is a mathematical function that maps data from its original space to a higher-dimensional feature space, where it may become linearly separable. The mapping is done in such a way that the dot product between two data points in the feature space can be calculated, which is used to determine the similarity between them. The graph on the left depicts two attributes,  $(x_1, x_2)$  where the data points represented inside the circle are one class and the data points represented outside the circle are another class. It is not possible to separate these data points into two distinct classes using a line. However, if the same set of data points are transformed into a three-dimensional feature space  $(x_1^2, x_2^2, \sqrt{2x_1x_2})$ , the two classes become linearly separable as shown in the figure 3.3.



**Figure 3.3:** Hyperplane through Non-linearly separable data

The KKT conditions are the same for the non-linear case as it is for linearly non-separable data. The quadratic problem with kernels function will be:

$$\min_{\alpha_k \geq 0} \left[ \frac{1}{2} \sum_{k,j} \alpha_k \alpha_j v_k y_j K(\vec{u}_k, \vec{u}_j) - \sum_{k=1}^L \alpha_k \right] \quad (3.4.1)$$

#### 3.4.2 Linear Kernel

The linear kernel is a simple mathematical function used in support vector machine (SVM) for performing binary classification. The main idea behind the linear kernel is to transform the input data into a higher-dimensional space in which the data points

become linearly separable.

The linear kernel transforms the input data by computing the dot product of the input features. If the input data is linearly separable in the original feature space, then it will also be linearly separable in the transformed space after the dot product. The linear kernel can be written mathematically as follows:

$$K(x, y) = u_k \cdot u_j + C$$

where  $u_k$  and  $u_j$  are the input feature vectors,  $\langle \cdot, \cdot \rangle$  is the dot product, and  $C$  is the regularization parameter. The effect of increasing hyperparameter  $C$  is to make the margin tighter, so that fewer support vectors will be needed to define the hyperplane. Cost determines the penalty for misclassification, with higher values of cost leading to a more strict margin between classes. A high value of cost can lead to over-fitting, while a low value can lead to under-fitting.

The linear kernel is a popular choice for many SVM applications because it has a simple mathematical form and is computationally efficient. However, it is not suitable for handling non-linearly separable data, and more complex kernels such as polynomial, radial basis function (RBF), and sigmoid kernels should be used in such cases.

### 3.4.3 Non-Linear Kernels

Non-linear kernels are a type of kernel function used in support vector machines (SVMs) to handle non-linearly separable data. Unlike linear kernels, which only consider linear combinations of the input features, non-linear kernels can capture complex relationships between the input features.

Examples of commonly used non-linear kernels include the radial basis function (RBF) kernel, the polynomial kernel, and the sigmoid kernel.

### 3.4.4 Polynomial Kernels

The polynomial kernel is defined as the dot product of two input vectors raised to a power " $d$ ". This helps to transform the input data into a higher dimensional space where a linear boundary can be used to separate the classes. Polynomial kernels can

model complex relationships in the input data, but they can also lead to over-fitting if the degree "d" is set too high.

For d-degree of the polynomial kernel

$$K(u_k, u_j) = (\vec{u}_k \cdot \vec{u}_j + c)^d \quad (3.4.2)$$

d value can be used to control the complexity of the model, the higher the d value more complex is model and cause over-fitting and the lower the d value simple the model is and can cause underfitting. The c value is used to control the trade-off between the fit of the training model and the size of the margin. The higher the C value, the lower the training error which results in overfitting. And lower the C value cause high training error but result in underfitting. In a polynomial kernel, the input data is recommended to be scaled concerning a feature before being applied to the Kernel function.

### 3.4.5 Radial Basis Function

In SVM, the radial basis kernel (also known as the RBF kernel or Gaussian kernel) is a popular choice of kernel function used to transform input data into a higher-dimensional space. The radial basis kernel is a highly efficient and complex method that combines multiple polynomial kernels, each with different degrees, to transform input data into a higher-dimensional space that can be separated by a hyperplane. The radial basis kernel operates by projecting input data into a high-dimensional space, achieved by computing the dot product and squares of all the features in the dataset. This higher-dimensional representation is then used for classification, following the same underlying principle as SVMs. It is defined as:

$$K(u_k, u_j) = \exp\left(-\frac{\|u_k - u_j\|^2}{2\sigma^2}\right) \quad (3.4.3)$$

where  $u_k$  and  $u_j$  are input vectors,  $\|u_k - u_j\|$  is the Euclidean distance between the two vectors, and  $\sigma$  is a hyperparameter that controls the width of the Gaussian distribution.

If,  $\gamma = \frac{1}{2\sigma^2}$  then,

$$K(u_k, u_j) = \exp(-\gamma\|u_k - u_j\|^2) \quad (3.4.4)$$

The RBF kernel is particularly useful when the decision boundary between classes is highly nonlinear, as it can capture complex relationships between input features. However, it is also prone to overfitting if the  $\gamma$  parameter is set too high.

### 3.4.6 Sigmoid Kernel

Mathematically, the sigmoid kernel is defined as:

$$K(u_k, u_j) = \tanh(\sigma * u_k * u_j + c)$$

,

where  $u_k$  and  $u_j$  are the input data vectors, and  $\sigma$  and  $c$  are kernel parameters. Let us break down the mathematical representation of the sigmoid kernel:

$u_k$  and  $u_j$  are input data vectors that represent the features of the data points you are working with. They are usually represented as column vectors:

$$u_k = [u_{k1}, u_{k2}, \dots, u_{kD}]^T$$

$$u_j = [u_{j1}, u_{j2}, \dots, u_{jD}]^T$$

Here,  $D$  represents the number of features in your data set.  $\sigma$  is a parameter of the sigmoid kernel that controls the non-linearity of the mapping. It is a positive real number.  $c$  is another parameter of the sigmoid kernel that adjusts the classification threshold. It is also a real number.  $\tanh z$  The hyperbolic tangent function ( $\tanh$ ) is an activation function commonly used in machine learning. It maps the input value  $z$  to a value between  $-1$  and  $1$ . Mathematically, it is defined as:

$$\tanh z = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)}$$

$K(u_k, u_j)$  represents the value of the sigmoid kernel function for the input vectors  $u_k$  and  $u_j$ . Calculate the product of dots between the transformed feature vectors and apply the hyperbolic tangent function. The dot product  $(u_k \cdot u_j)$  calculates the sum of the element-wise multiplication of the corresponding elements of  $u_k$  and  $u_j$ .

## 3.5 Penalized Support Vector Machine

Considering the optimization problem with the objective function in the form of "*loss + penalty*" as given in equation 3.5.1

$$\min_{b, \psi} \sum_{k=1}^L [1 - v_k f(u_k)]_+ + \frac{\gamma}{2} \|\psi\|^2 \quad (3.5.1)$$

where  $\sum$  denotes the sum over all training examples,  $[1 - v_k f(u_k)]_+$  represents the positive part (hinge loss),  $\frac{\gamma}{2} * \|\psi\|^2$  is the penalty term with  $\gamma$  being the regularization parameter (inverse of  $C$ ).

It can be shown that the solution to this optimization problem, with  $\gamma = 1/C$ , is equivalent to the solution of the equation 3.3.5 that corresponds to the formulation of SVM with the hinge loss and  $L_2$ norm penalty [12].

$$\min_{b, \psi} \frac{1}{n} l(v_k, f(u_k)) + \text{pen}_\gamma(\psi) \quad (3.5.2)$$

In this equation, the loss component is represented by the aggregate of hinge loss functions  $l(v_k, f(u_k))$ , which is defined as  $\max(1 - v_k f(u_k), 0)$ . Each sample  $u_k$ , with  $i$  ranging from 1 to  $L$ , contributes to the loss function. The hinge loss penalizes misclassifications and encourages correct classification with a margin of at least 1.

The penalty term in equation 3.5.2 is denoted as  $\text{pen}_\gamma(\psi)$ ,  $\psi$  represents the hyperplane coefficients of the SVM model. The penalty component can take different forms depending on the specific regularization method used. The choice of penalty term,  $\text{pen}_\gamma(\psi)$ , determines the regularization approach employed in SVM. Common penalty terms include the  $L_1$ norm penalty, the  $L_2$ norm penalty, and variations such as SCAD penalties. These penalty terms control the model sophistication, encourage sparsity, and prevent over-fitting by constraining the weight coefficients.

### 3.5.1 Ridge Penalty

The penalty term utilized in the standard SVM formulation involves the use of the  $L_2$  norm, commonly referred to as the ridge penalty.

$$\text{pen}_\gamma(\psi) = \gamma \|\psi\|_2^2 = \gamma \sum_{j=1}^p \psi_j^2$$

This penalty is used to regulate the variance of the coefficients by shrinking them. However, it is worth mentioning that the ridge penalty does not promote sparsity by shrinking the coefficients to zero. Consequently, it does not facilitate feature selection as all coefficients retain their significance within the model.

The objective function can be formulated using Loss + Penalty criteria [12]:

$$\min_{b, \psi} \sum_{k=1}^L (\max(0, 1 - v_k(\psi * u_k + b))) + \frac{\gamma}{2} (\|\psi\|_2)^2 \quad (3.5.3)$$

where  $\gamma$  is the regularization parameter that controls the strength of the regularization. The focus of this formulation is on regularization, which is important even when there are enough variables (such as gene expression arrays) to ensure separation. Instead of relying on observations on the boundary to dictate a maximum margin separator with a small value of  $\gamma$ , a more regularized solution can be chosen that involves more observations. Additionally, this formulation allows for a range of flexible, non-linear generalizations. [13]

### 3.5.2 Least Absolute Shrinkage And Selection Operator Penalty

Tibshirani [14] originally proposed the utilization of a  $L_1$  penalization function in generalized linear models, which led to the growth of the LASSO technique. LASSO enables parameter estimation while incorporating constraints, and achieves sparsity by reducing certain coefficients to zero. Bradley [15] subsequently adapted the  $L_1$  regularization concept to SVMs, allowing for automatic feature selection by encouraging specific hyperplane coefficients to shrink towards zero. This adaptation extends the capabilities of SVMs, enabling them to identify and prioritize the most relevant features.

$$pen_{\gamma}(\psi) = \gamma \|\psi\|_1 = \gamma \sum_{j=1}^p |\psi|_j$$

Due to the uniqueness of the  $L_1$  penalty function, the  $L_1$  SVM exhibits the advantageous property of automatically selecting features by shrinking the hyperplane coefficients to zero.

Nevertheless, there are two limitations associated with the  $L_1$  norm penalty. Firstly, the number of selected variables is restricted by the number of available values. Secondly, in cases where there are correlated features, the  $L_1$  norm penalty often selects only one feature from the correlated group while disregarding the others.

### 3.5.3 Smoothly Clipped Absolute Deviation Penalty

While the  $L_1$  penalty in linear regression models can generate sparse solutions, it can also introduce bias in the estimates when dealing with large coefficients. To address this issue, Fan and Li [16] introduced the smoothly clipped absolute deviation (SCAD) penalty. The SCAD penalty not only promotes sparsity by setting small estimates to zero, but also mitigates the bias problem associated with the  $L_1$  penalty. Furthermore,

the SCAD penalty allows for nearly unbiased estimates of large coefficients and maintains the continuity of the model with respect to the data. Mathematically, the SCAD penalty can be expressed as follows.

$$p_{SCAD(\gamma)}(\psi) = \begin{cases} \gamma|\psi| & \text{if } |\psi| \leq \gamma, \\ \frac{-(|\psi|^2 - 2a\gamma|\psi| + \gamma^2)}{2(a-1)}, & \text{if } \gamma < |\psi| \leq a\gamma \\ \frac{(a+1)\gamma^2}{2}, & \text{if } |\psi| > a\gamma \end{cases}$$

By setting the tuning parameters  $a > 2$  and  $\gamma > 0$ , the function in the above equation becomes a quadratic spline function with two knots located at  $\gamma$  and  $a\gamma$ . Apart from its singularity at the origin, the function  $p_{SCAD(\gamma)}(\psi)$  possesses a first-order derivative that remains continuous. As a result, we introduce the SCAD SVM model as

$$\min_{\psi, b} \frac{1}{n} \sum_{i=1}^n [1 - v_k(b + \psi f(u_k))]_+ + \sum_{j=1}^p p_{\gamma}(|\psi_j|) \quad (3.5.4)$$

The parameter  $\gamma$  plays a crucial role in striking a balance between data fitting and model simplicity. When  $\gamma$  is too small, the learning process tends to overly fit the training data, resulting in a classifier that lacks sparsity. On the other hand, when  $\gamma$  is excessively large, the resulting classifier may be highly sparse but lacks discriminative power. Thus, the choice of  $\gamma$  needs to be carefully considered, achieving an optimal trade-off between these competing objectives.

### 3.5.4 Elastic Smoothly Clipped Absolute Deviation Penalty

In their study, Fan and Li [16] highlighted the benefits of the SCAD penalty compared to the  $L_1$  penalty. Nevertheless, when dealing with non-sparse data, employing the SCAD penalty alone can result in excessively strict feature selection. To address the limitations of the SCAD penalty for non-sparse data, we propose a hybrid approach that combines the SCAD and  $L_2$  penalties. This new penalty term takes the form of an integrated  $SCAD + L_2$  penalty, aiming to leverage the advantages of both penalties.

$$p_{\gamma}(\psi) = \sum_{j=1}^p p_{SCAD(\gamma_1)}(\psi_j) + \gamma_2 \|\psi\|_2^2 \quad (3.5.5)$$

The Elastic SCAD method incorporates tuning parameters  $\gamma_1$  and  $\gamma_2$ , both of which are non-negative. Our expectation is that the Elastic SCAD approach will enhance the

performance of the SCAD method when dealing with less sparse data. Due to the inherent characteristics of the SCAD and  $L_2$  penalties, we anticipate that the Elastic SCAD method will demonstrate strong predictive accuracy for both sparse and non-sparse data. Furthermore, it can be demonstrated that the combined penalty employed by Elastic SCAD exhibits desirable properties such as sparsity, continuity, and asymptotic normality. Specifically, as the tuning parameter for the  $L_2$  penalty approaches zero, the Elastic SCAD method demonstrates asymptotic normality and sparsity. These properties contribute to the oracle property, as defined by Fan and Li [16].

The Elastic SCAD SVM has a form

$$\min_{\psi, b} \frac{1}{n} \sum_{i=1}^n [1 - v_k(b + \psi f(u_k))]_+ + \sum_{j=1}^p p_{\gamma_1}(|\psi_j|) + \gamma_2 \|\psi\|_2^2 \quad (3.5.6)$$

In traditional approaches, quadratic programming and linear programming methods are commonly used to solve standard SVM and  $L_1$  SVM problems. However, these methods face challenges when applied to the SCAD SVM problem described in equation 3.5.6. The non-differentiability of the hinge loss function at zero and the non-convexity of the SCAD penalty in terms of the weight vector  $w$  can cause standard optimization packages to fail. To address these challenges, we propose an iterative algorithm to efficiently solve the SCAD SVM problem. Our approach utilizes the Successive Quadratic Algorithm (SQA), which is an extension of Newton's method for unconstrained optimization. The SQA iteratively minimizes a quadratic approximation of the problem, allowing us to find a step away from the current point towards the optimal solution [17].

## 3.6 Evaluation

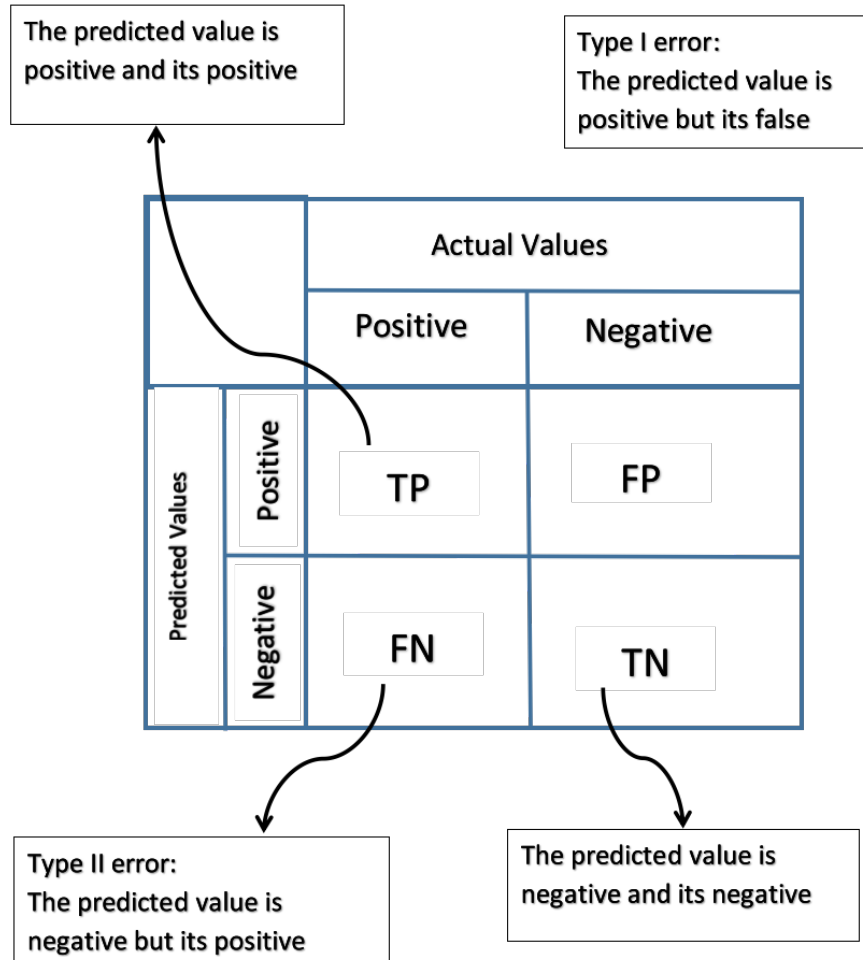
The evaluation of classification models is an essential step in assessing the efficiency of an algorithm on new data. The assessment helps to find how good the algorithm can generalize to new data and make accurate predictions. Here are some common evaluation techniques for classification models.

- **Cross-validation:** This strategy include splitting of the dataset into  $m$  subsets or folds. The algorithm is trained and assess  $m$  times, with each fold serving as the test set once while the remaining folds are used for training. The performance results are then averaged to provide a more reliable estimate of the effectiveness



of the model.

- Confusion matrix: Specificity used to inspect the effectiveness of binary classification models, specifically targeting the negative class. It represents the proportion of accurately identified negative instances relative to the total number of actual negative cases in the dataset.



**Figure 3.4:** The figure illustrate confusion matrix

- Specificity is a performance metric used to assess binary classification models, with a focus on the negative class. It quantifies the fraction of true negative instances that are correctly identified by the model relative to the total number of actual negative instances in the dataset.

Mathematically, specificity is calculated as:

$$\textit{Specificity} = \frac{\textit{True Negatives}}{(\textit{True Negatives} + \textit{False Positives})}$$

where true negatives are the instances that belong to the negative class and are correctly classified as negative, and false positives are the instances that belong to the negative class but are incorrectly classified as positive.

- Sensitivity is a metric used to evaluate the performance of a binary classification model, particularly when the positive class is of interest. Sensitivity measures the proportion of correctly identified positive instances out of all actual positive instances in the data set.

Mathematically, sensitivity is calculated as:

$$\textit{Sensitivity} = \frac{\textit{True Positives}}{(\textit{True Positives} + \textit{False Negatives})}$$

true positives are the instances that belong to the positive class and are correctly classified as positive, and false negatives are those that belong to the positive class but are incorrectly classified as negative.

- Accuracy is a metric that is used to evaluate the performance of a classification model. It measures the proportion of correctly classified instances out of all the instances in the data sets. In other words, it is the ratio of the number of correct predictions to the total number of predictions.

Mathematically, accuracy is calculated as follows:

$$\textit{Accuracy} = \frac{(\textit{TP} + \textit{TN})}{(\textit{TP} + \textit{TN} + \textit{FP} + \textit{FN})}$$

where true positives (TP) are instances correctly identified as belonging to the positive class. True negatives (TN) are instances correctly identified as belonging to the negative class. False positives (FP) are negative class instances incorrectly classified as positive, and false negatives (FN) are positive class instances incorrectly classified as negative.

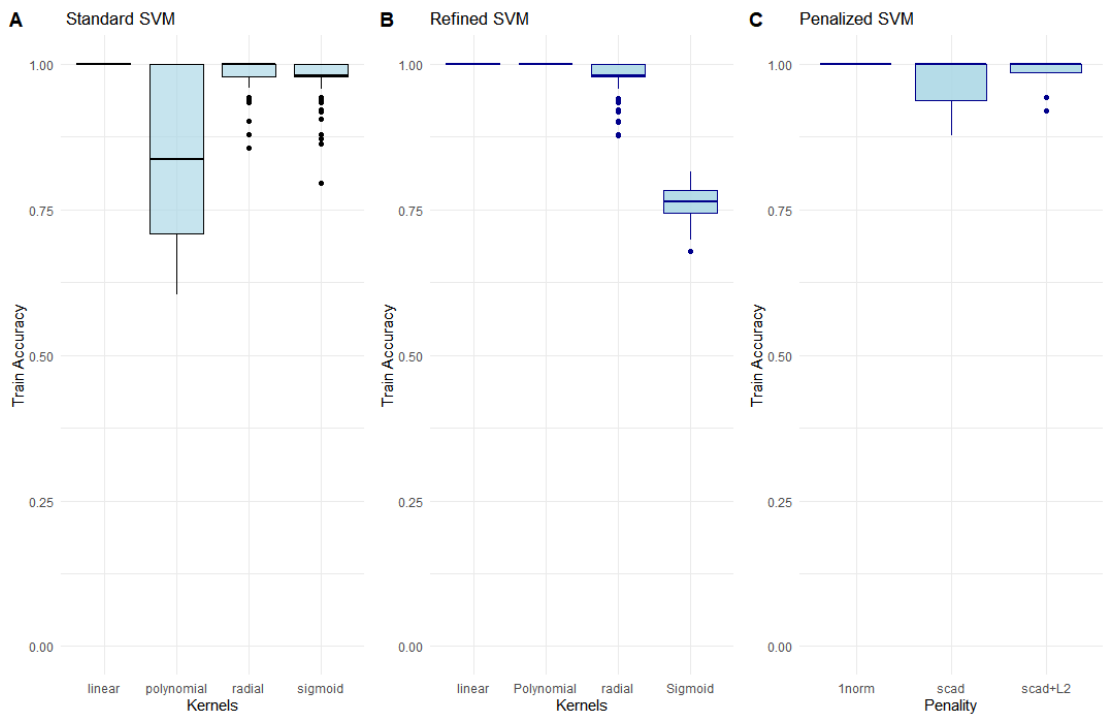
- Post hoc analysis, derived from the Latin term "post hoc" meaning "after this," involves examining the outcomes of experimental data. Typically, such analyses consider the family-wise error rate, which refers to the probability of encountering

at least one Type I error within a set or family of comparisons. One commonly employed post hoc analysis method is the Honest Significant Difference (HSD) test, which is highly popular. The HSD test adjusts the test statistic when conducting pairwise comparisons between two groups. These comparisons provide an estimation of the disparity between the groups and offer a confidence interval for the estimate. We interpret this confidence interval similarly to how we interpret the confidence interval in a standard independent samples t-test: if the interval encompasses 0.00, it indicates that the groups are not significantly different, whereas if it does not include 0.00, it suggests that there is a significant difference between the groups.

# Results And Discussion

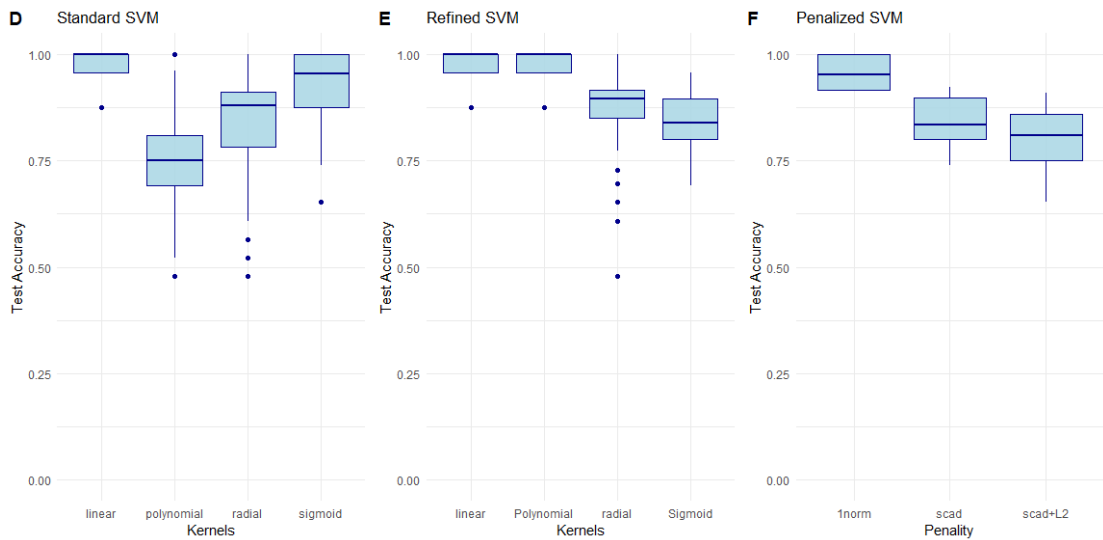
In this section, we provide an overview of our analysis conducted on the gene expression data from the Leukemia dataset using various versions of the Support Vector Machine (SVM) model. The data set includes 72 samples with 5147 characteristics and is split into a training set 80% and a testing set 20%.

Three variants of SVM, ie standard SVM, refined SVM and penalized SVM, are applied on the train and test data set for Leukemia cancer. The performance achieved is shown in the following boxplots.



**Figure 4.1:** This figure illustrates the train accuracy of different variants of SVM

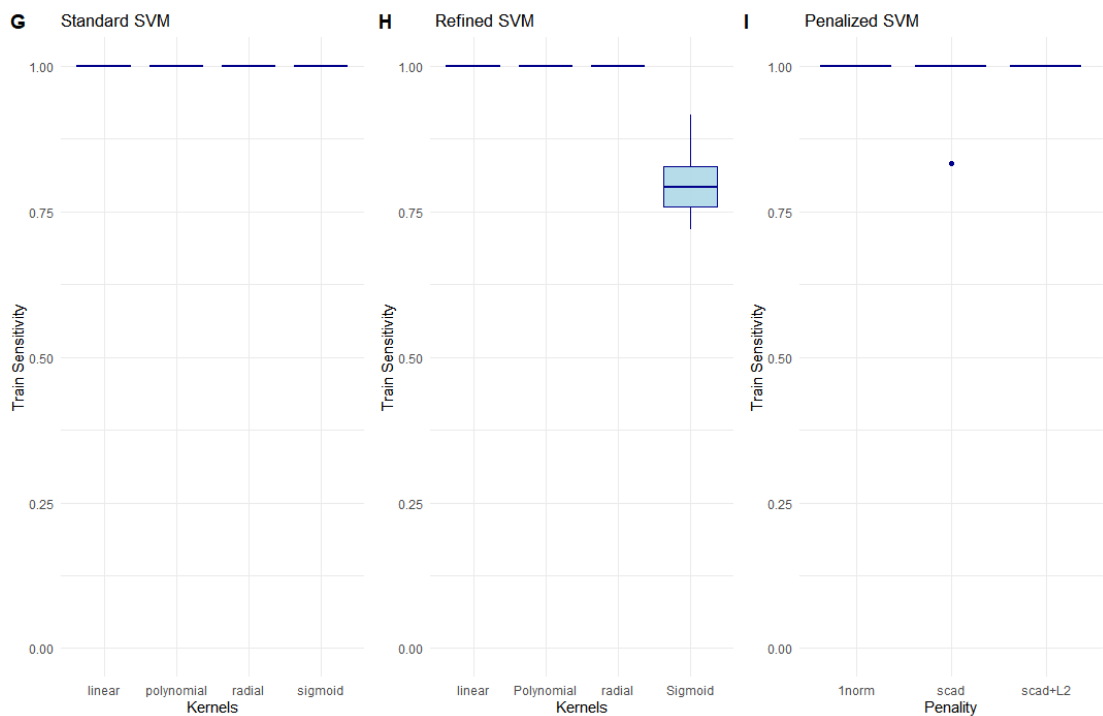
The train accuracy analysis depicted in Figure 4.1 highlights distinct trends among different SVM kernels and regularization methods. Notably, the linear kernel, employed in both standard and refined SVM, as well as the polynomial kernel utilized in refined SVM, demonstrate superior performance with consistently high mean accuracy scores. This suggests their reliability and efficacy across various datasets. Conversely, the radial kernel employed in standard SVM, along with regularization methods such as *scad* and *scad* +  $L_2$  penalties, exhibit mean accuracy scores of 1 but display notable variations, implying less stable performance. Furthermore, the sigmoid kernel employed in refined SVM demonstrates the lowest mean accuracy, indicating potential challenges in classification accuracy with this particular configuration. These findings suggest that the linear kernel in both standard and refined SVM and polynomial kernel in refined in both standard and refined SVM may be more suitable for classification tasks requiring high accuracy and stability.



**Figure 4.2:** This figure illustrates the test accuracy of different variants of SVM

The analysis of various SVM kernels, as depicted in Figure 4.2, underscores significant variation in their test accuracy performance. Among them, the standard linear kernel, refined linear, and polynomial kernels exhibit commendable consistency, boasting a mean accuracy of 1. This signifies robust classification capabilities across dataset, with only a single outlier observed at 0.875. On the other hand, employing the  $L_1$  penalty yields a mean accuracy of 0.9, albeit with some variability. The SCAD (Smoothly Clipped Absolute Deviation) penalty demonstrates a mean test accuracy of 0.8, also with no-

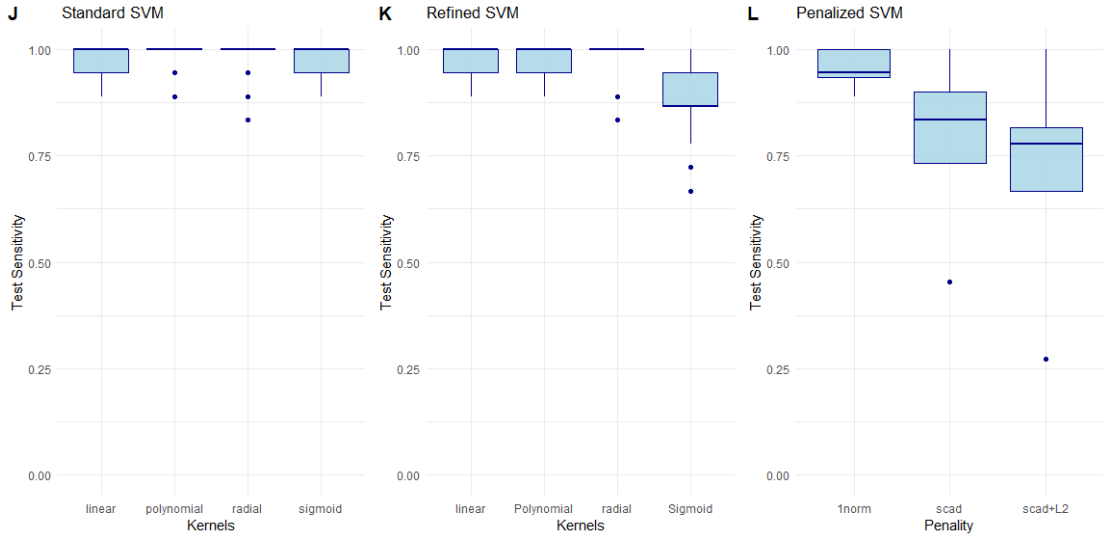
ticeable variation. Incorporating both SCAD and  $L_2$  penalty results in a mean accuracy of 0.79, indicating a less consistent behavior compared to other kernels. Similarly, while the standard polynomial kernel showcases a mean accuracy of 0.75 with variation and one outlier, the sigmoid kernel achieves a mean accuracy of 0.95, albeit with variation and an outlier. In contrast, both the polynomial and radial kernels exhibit lower mean accuracy of 0.75 and 0.875, respectively, with multiple outliers at various accuracy levels, suggesting less stable performance. The sigmoid kernel stands out with the highest mean accuracy of 0.95, indicating robust classification performance, supported by only one outlier at 0.63. These findings suggest that the linear and sigmoid kernels may be more suitable for classification tasks requiring high accuracy and stability, while the polynomial and radial kernels may exhibit more variable performance.



**Figure 4.3:** This figure illustrates the train sensitivity of different variants of SVM

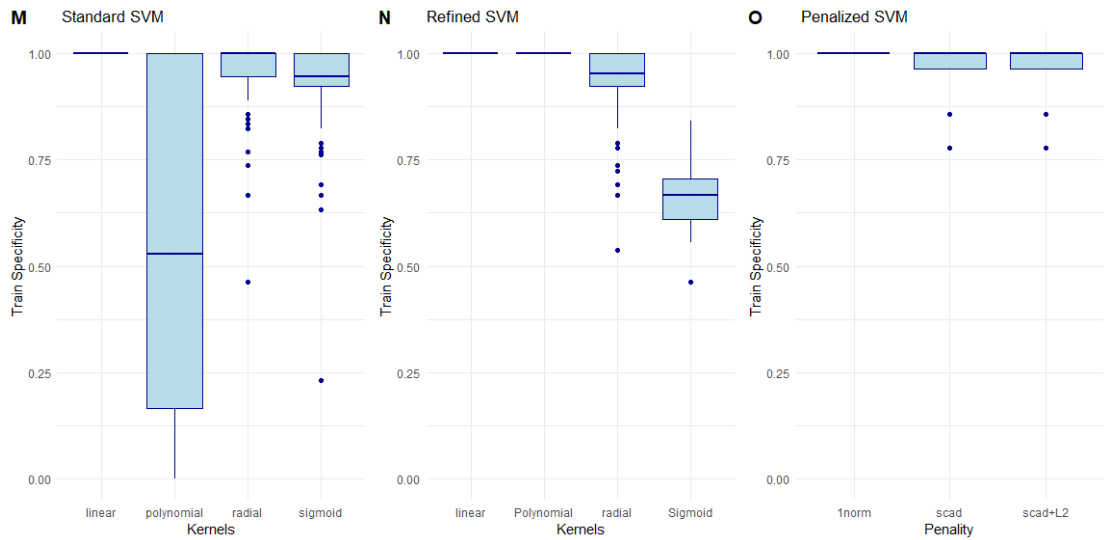
Figure 4.3 demonstrate the train sensitivity of variants of SVM which is pretty good for all of kernels and penalized SVM except a sigmoid kernel in refined SVM. A sensitivity rate of 0.8 for the sigmoid kernel suggests that it accurately identifies approximately 80% positive instances, reflecting its effectiveness.

The test sensitivity results depicted in Figure 4.4 illustrate predominantly strong performance across various SVM kernels, with sensitivity scores mostly reaching 1, signifying



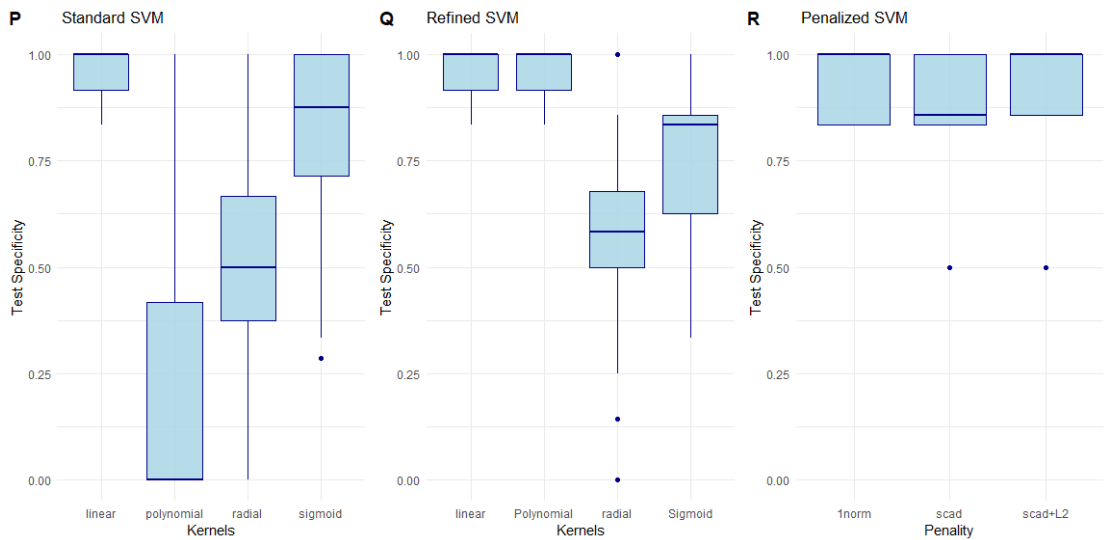
**Figure 4.4:** This figure demonstrates test sensitivity of different variants of SVM

excellent capability in correctly identifying positive instances. However, an exception is observed with the refined SVM utilizing the sigmoid kernel, suggesting potential challenges in correctly identifying positives in this specific configuration. Outliers in sensitivity are noted in several kernels, including standard radial, polynomial, and refined radial, as well as the sigmoid kernels, indicating instances where sensitivity deviates notably from the norm across datasets. Notably, the refined sigmoid kernel exhibits a median sensitivity of 0.875, indicating generally strong performance, albeit with some variability and outliers. In contrast, penalized models like SCAD and  $SCAD + L_2$  demonstrate slightly lower sensitivity scores of 0.83 and 0.8, respectively, while the  $L_1$  penalty model shows a mean test sensitivity of 0.9 with some variation, suggesting that while penalized models perform adequately, they may not consistently match the sensitivity levels achieved by certain other kernels. The specificity analysis reveals significant differences among SVM kernels, particularly between standard and refined SVM setups. Kernels like standard linear, refined linear, polynomial, and  $L_1$  exhibit perfect train specificity, indicating accurate identification of true negatives. However, standard radial, SCAD, and  $SCAD + L_2$  SVMs show median train specificity at 1 with some variability and outliers, suggesting challenges in discerning negatives. Standard sigmoid and radial kernels maintain specificity around 0.9 with variability and outliers, while the refined sigmoid kernel demonstrates lower specificity at 0.7, alongside notable variability and outliers. Notably, standard polynomial SVM displays the weakest specificity. Penalized SVM models generally maintain high specificity. These results emphasize the importance of



**Figure 4.5:** This figure illustrate train specificity of different variants of SVM

selecting appropriate kernels and refined strategies in SVM-based classification tasks.



**Figure 4.6:** This figure illustrate the test specificity of different variants of SVM

The specificity analysis unveils diverse performance patterns across different SVM models and kernel functions. Standard linear kernel, refined linear and polynomial kernel and  $L_1$  and  $scd + L_2$  displays a median test specificity of 1 with some variation, indicating perfect classification of negative instances. Standard polynomial SVM, however, struggles with a median specificity of 0, suggesting a high rate of false positives. Radial SVM exhibits moderate specificity, while sigmoid SVM demonstrates strong performance with a median specificity of 0.875. Refined SVM models generally enhance specificity, espe-



cially in linear and polynomial kernels. Penalized SVM methods vary in specificity, with  $L_1$  regularization showing perfect performance but SCAD and  $SCAD+L_2$  regularization exhibiting slight deviations and outliers, highlighting potential areas for improvement.

**Table 4.1:** ANOVA Table for train accuracy for standard SVM

	df	ss	ms	F	Pr(>F)
Kernels	3	14.12	4.705	930.9	$< 2e - 16^{***}$
Residuals	5116	25.86	0.005		
Signifi: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1					

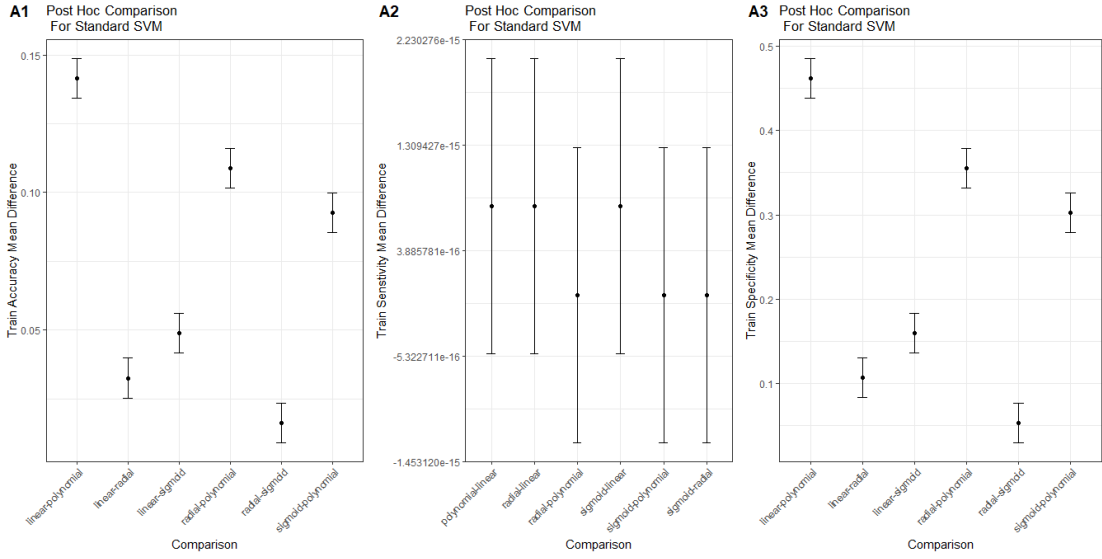
**Table 4.2:** ANOVA Table for train sensitivity for standard SVM

	df	ss	ms	F	Pr(> F)
Kernels	3	$5.000 \times 10^{-28}$	$1.602 \times 10^{-28}$	1	0.392
Residuals	5116	$8.195 \times 10^{-25}$	$1.602 \times 10^{-28}$		

**Table 4.3:** ANOVA Table for train specificity for standard SVM

	df	ss	ms	F	Pr(>F)
Kernels	3	40.52	13.507	26344	$< 2 \times 10^{-16^{***}}$
Residuals	3836	1.97	0.001		
Signifi: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1					

The conducted series of ANOVA tests aimed at assessing the impact of different SVM kernels on model performance, focusing on train accuracy, sensitivity, and specificity. The results revealed compelling insights pertinent to our research inquiry. In terms of train accuracy, the analysis unveiled a statistically significant effect of SVM kernels on model performance, indicating that the choice of kernel significantly influences accuracy outcomes. Subsequent post hoc tests illuminated specific differences between pairs of kernels, providing granular insights into their relative performance. Notably, non-overlapping confidence intervals between kernels underscored significant disparities in accuracy. However, the analysis pertaining to train sensitivity did not yield statistically significant results, suggesting that the variation in SVM kernels did not significantly affect sensitivity metrics. Conversely, the examination of train specificity echoed the findings observed in accuracy, highlighting a notable influence of SVM kernels on model



**Figure 4.7:** Post-hoc comparison for train standard SVM

performance. The post hoc assessments further delineated specific differences between kernel pairs, reinforcing the significance of kernel selection in optimizing model specificity. Visual representations of these findings serve as pivotal contributions to our thesis, offering a comprehensive understanding of the comparative performance of SVM kernels across different performance metrics. These insights inform strategic decisions in selecting the most efficacious kernel for classification tasks, thereby enriching the scholarly discourse within our research domain.

**Table 4.4:** ANOVA Table for test accuracy for standard SVM

	df	ss	ms	F	Pr(>F)
Kernels	3	95.52	31.84	2416	$< 2 \times 10^{-16}***$
Residuals	5116	67.41	0.01		
Signifi: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1					

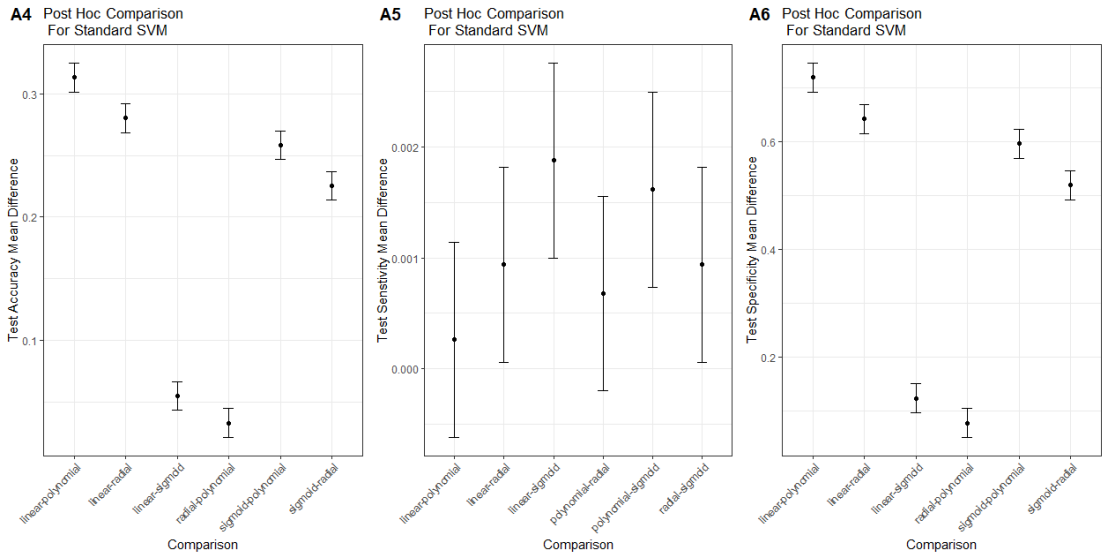
**Table 4.5:** ANOVA Table for test sensitivity for standard SVM

	df	ss	ms	F	Pr(>F)
Kernels	3	0.0027	0.0008967	12.01	$7.87 \times 10^{-8}***$
Residuals	5116	0.3821	0.0000747		
Signifi: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1					

**Table 4.6:** ANOVA Table for test specificity for standard SVM

	df	ss	ms	F	Pr(>F)
Kernels	3	503.1	167.69	2425	$< 2 \times 10^{-16}***$
Residuals	5116	353.7	0.07		

Signifi: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘.’ 1



**Figure 4.8:** Post-Hoc comparison for test standard SVM

The ANOVA tests conducted on test accuracy, sensitivity, and specificity with respect to different SVM kernels yielded significant findings. For test accuracy, the analysis revealed a highly significant effect of SVM kernels on model performance ( $F(3, 5116) = 2416, p < 0.001$ ). Subsequent post hoc tests identified specific differences between pairs of kernels, illustrating mean differences in test accuracy along with 95% confidence intervals. Regarding test sensitivity, the ANOVA analysis demonstrated a significant effect of SVM kernels on model performance ( $F(3, 5116) = 12.01, p < 0.001$ ). Post hoc tests unveiled specific differences between kernel pairs, elucidating mean differences in test sensitivity and corresponding confidence intervals. Similarly, the analysis concerning test specificity showcased a highly significant effect of SVM kernels on model performance ( $F(3, 5116) = 2425, p < 0.001$ ). Post hoc tests delineated specific differences between pairs of kernels, depicting mean differences in test specificity along with 95% confidence intervals. The graphical representations of these findings provide a visual understanding of the comparative performance of SVM kernels across different evaluation metrics,

guiding informed decisions in kernel selection for classification tasks. These insights contribute significantly to the advancement of knowledge in the field and inform strategic decisions in machine learning model development.

**Table 4.7:** ANOVA Table for train accuracy for refined SVM

	df	ss	ms	F	Pr(>F)	
Kernels	3	40.52	13.507	26344	$< 2 \times 10^{-16***}$	
Residuals	3836	1.97	0.001			
Signifi:	0 ‘***’	0.001 ‘**’	0.01 ‘*’	0.05 ‘.’	0.1 ‘ ’	1

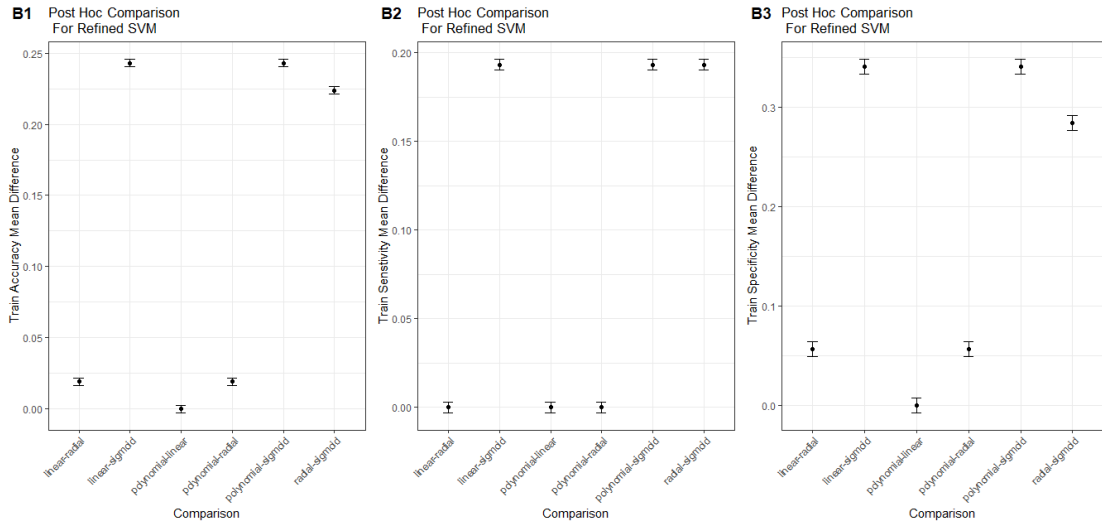
**Table 4.8:** ANOVA Table for train sensitivity for refined SVM

	df	ss	ms	F	Pr(>F)	
Kernels	3	26.90	8.966	13486	$< 2 \times 10^{-16***}$	
Residuals	3836	2.55	0.001			
Signifi:	0 ‘***’	0.001 ‘**’	0.01 ‘*’	0.05 ‘.’	0.1 ‘ ’	1

**Table 4.9:** ANOVA Table for train specificity for refined SVM

	df	ss	ms	F	Pr(>F)	
Kernels	3	76.52	25.506	6570	$< 2 \times 10^{-16***}$	
Residuals	3836	14.89	0.004			
Signifi:	0 ‘***’	0.001 ‘**’	0.01 ‘*’	0.05 ‘.’	0.1 ‘ ’	1

Figure 4.9 illustrate the ANOVA results for train accuracy showed a significant effect of kernel type ( $F(3, 3836) = 26344, p < 0.001$ ), indicating that the choice of kernel significantly impacts the accuracy of the SVM model. The subsequent Tukey HSD post-hoc test revealed specific pairwise differences between the kernels. The plot (B1) illustrates these differences, showing the mean difference in train accuracy for each pair of kernels, along with 95% confidence intervals. The small confidence intervals suggest high precision in our estimates, indicating robust and reliable differences in accuracy between the kernels. Similarly, the ANOVA results for train sensitivity also demonstrated a significant effect of kernel type ( $F(3, 3836) = 13486, p < 0.001$ ). The Tukey HSD post-hoc test identified specific pairs of kernels with significant differences in sensitivity. The corresponding plot (B2) depicts these differences, with mean differences and confidence



**Figure 4.9:** Post-Hoc comparison for train refined SVM

intervals indicating the variability. The smaller differences and confidence intervals suggest that while there are significant differences, they might not be large in magnitude, yet they are consistent and statistically meaningful. For train specificity, the ANOVA results again confirmed a significant effect of kernel type ( $F(3, 3836) = 6570, p < 0.001$ ). The post-hoc comparisons provided detailed insights into which kernel pairs differ significantly in specificity. The plot (B3) presents these findings, showing mean differences and confidence intervals. The results show small but significant differences in specificity among kernels, emphasizing the importance of kernel selection for optimal model performance. The analyses collectively underscore the critical role of kernel selection in SVM performance. Each kernel type affects train accuracy, sensitivity, and specificity differently, with statistically significant differences observed in each measure. Although some of these differences are small, their consistency and significance highlight that even minor variations in kernel choice can lead to meaningful changes in model performance. This is particularly crucial in applications requiring high precision, such as medical diagnostics or anomaly detection, where optimizing every aspect of model performance can significantly impact outcomes.

The ANOVA and post hoc analysis of the refined SVM models revealed significant differences in performance metrics—test accuracy, sensitivity, and specificity—across different kernel functions. The ANOVA results indicated a highly significant effect of the kernel type on these metrics ( $p < 2e - 16$  for all). Post hoc comparisons showed that the radial kernel generally outperforms the sigmoid, linear, and polynomial ker-

**Table 4.10:** ANOVA Table for Test Accuracy for Refined SVM

	<b>df</b>	<b>ss</b>	<b>ms</b>	<b>F</b>	<b>Pr(&gt;F)</b>
Kernels	3	13.78	4.595	932.3	$< 2 \times 10^{-16}***$
Residuals	3836	18.91	0.005		

Signifi: 0 '\*\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**Table 4.11:** ANOVA Table for Test Sensitivity for Refined SVM

	<b>df</b>	<b>ss</b>	<b>ms</b>	<b>F</b>	<b>Pr(&gt;F)</b>
Kernels	3	7.289	2.4296	884.8	$< 2 \times 10^{-16}***$
Residuals	3836	10.533	0.0027		

Signifi: 0 '\*\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

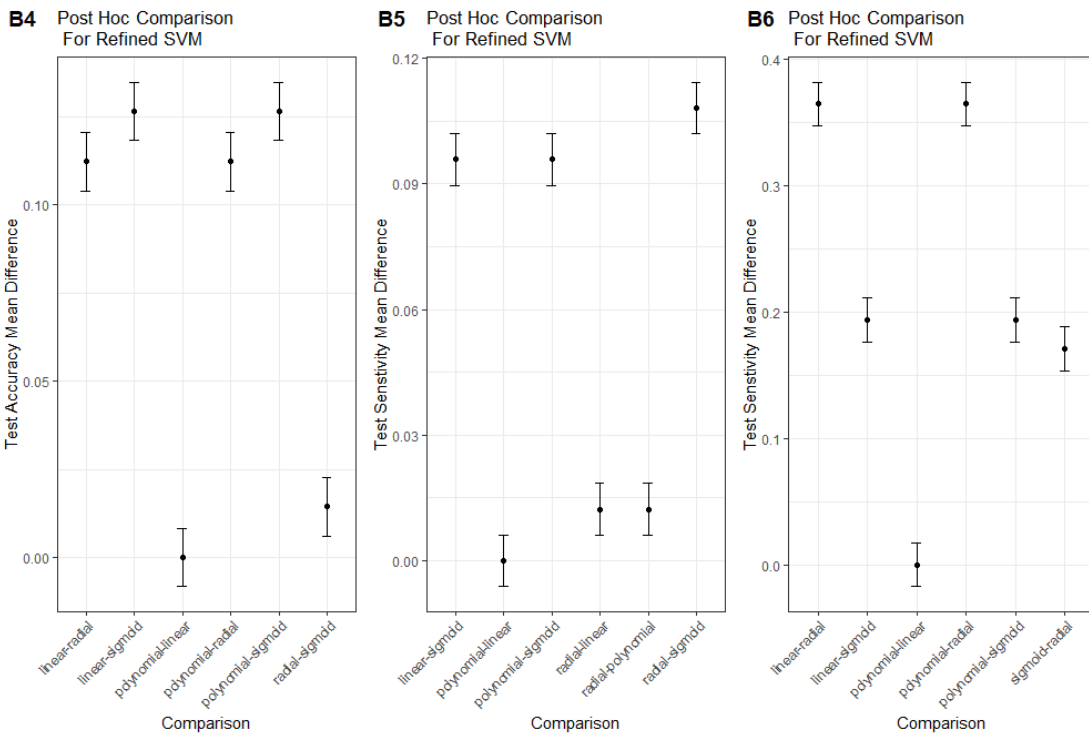
nels in terms of sensitivity and specificity, with substantial differences observed (e.g., radial-sigmoid comparison in specificity showing a difference of 0.171). The linear and polynomial kernels displayed similar performance, particularly in accuracy and specificity. These findings highlight the importance of kernel selection in SVM modeling, suggesting that the radial kernel may be preferable for achieving higher sensitivity and specificity, while the choice between linear and polynomial kernels may be less critical due to their comparable performance. Visual plots further corroborate these results, providing a clear depiction of mean differences and confidence intervals, emphasizing the significant impact of kernel choice on SVM performance.

The analysis conducted on penalized SVM models revealed notable insights into their performance across various penalty types. ANOVA tests unveiled statistically significant differences in train accuracy, where the  $L_1$  penalty outperformed SCAD, as evidenced by a significant F and a p-value below the 0.05 threshold. However, while differences in train sensitivity and specificity were also observed, they did not reach the same level of significance, as indicated by higher p-values. Further exploration through post hoc comparisons provided additional clarity. For train accuracy, the confidence intervals indicated significant differences between penalties, aligning with the ANOVA findings. Conversely, for train sensitivity and specificity, the confidence intervals passing through zero suggested non-significant differences between penalty types, indicating that the choice of penalty may have a less pronounced impact on these metrics. These results un-

**Table 4.12:** ANOVA Table for test specificity for refined SVM

	df	ss	ms	F	Pr(>F)
Kernels	3	88.91	29.635	1357	$< 2 \times 10^{-16}***$
Residuals	3836	83.78	0.022		

Signifi: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘.’ 1



**Figure 4.10:** Post-Hoc comparison for test Refined SVM

underscore the nuanced influence of penalty selection on model performance and highlight the need for careful consideration when optimizing SVM models for specific objectives.

The analysis conducted on penalized SVM models aimed to investigate their performance concerning test accuracy, sensitivity, and specificity. In the ANOVA analysis for test accuracy, a statistically significant difference was observed across penalization methods ( $df = 2, F = 28.87, p < 0.001$ ). Subsequent post-hoc Tukey tests revealed significant differences between the " $L_1$ " and both " $scad + L_2$ " and " $L_1 - scad$ " methods ( $p < 0.001$ ), indicating varying performance levels among the penalization techniques. Similarly, in the analysis of test sensitivity, a significant difference was found among the penalization methods ( $df = 2, F = 10.36, p = 0.000145$ ). Post-hoc comparisons showed significant differences between " $L_1$ " and both " $scad + L_2$ " and " $L_1 - scad$ " methods

**Table 4.13:** ANOVA Table for train accuracy for penalized SVM

	<b>df</b>	<b>ss</b>	<b>ms</b>	<b>F</b>	<b>Pr(&gt;F)</b>
Penalities	2	0.01216	0.006081	5.771	0.00522**
Residuals	57	0.06006	0.001054		

Signifi: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

**Table 4.14:** ANOVA Table for train sensitivity for penalized SVM

	<b>df</b>	<b>ss</b>	<b>ms</b>	<b>F</b>	<b>Pr(&gt;F)</b>
Penalities	2	0.00833	0.004167	3.353	0.042*
Residuals	57	0.07083	0.001243		

Signifi: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

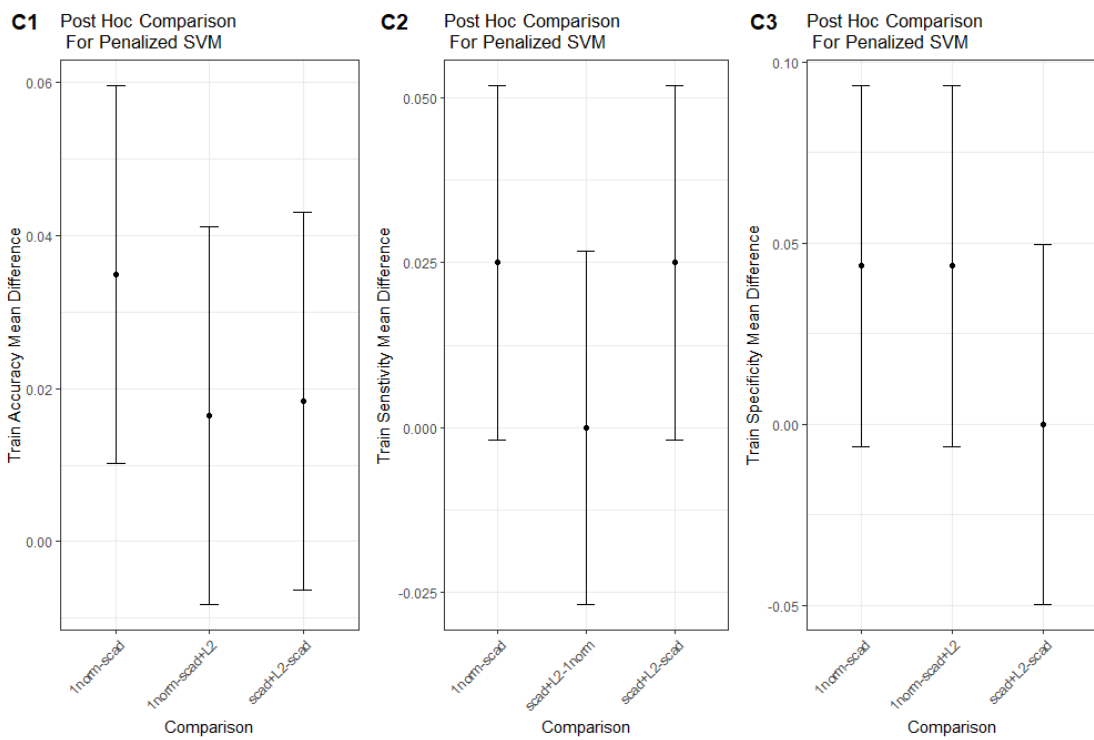
( $p < 0.01$ ). However, for test specificity, no significant difference was observed among the penalization methods ( $df = 2, F = 1.716, p = 0.189$ ). These findings suggest that while penalization methods significantly impact test accuracy and sensitivity, they do not have a significant effect on test specificity. This nuanced understanding of penalized SVM model performance can inform practitioners in selecting appropriate penalization strategies based on specific evaluation criteria.



**Table 4.15:** ANOVA Table for train specificity for penalized SVM

	df	ss	ms	F	Pr(>F)
Penalties	2	0.02541	0.012703	2.97	0.0593
Residuals	57	0.24376	0.004277		

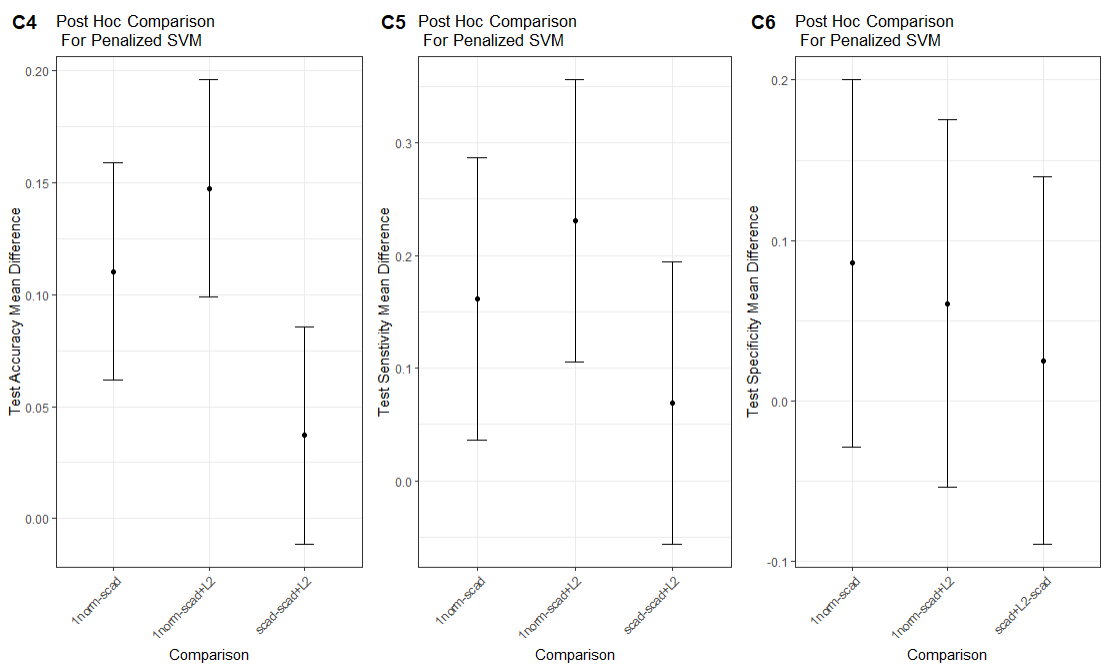
Signifi: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1



**Figure 4.11:** Post-Hoc comparison for train penalized SVM

**Table 4.16:** ANOVA Table for test specificity for penalized SVM

	df	ss	ms	F	Pr(>F)
Penalties	2	0.0777	0.03886	1.716	0.189
Residuals	57	1.2906	0.02264		



**Figure 4.12:** Post-Hoc comparison for test Penalized SVM

# Conclusion

The comprehensive analysis of various Support Vector Machine (SVM) variants applied to the Leukemia dataset has provided valuable insights into their performance across different evaluation metrics. Rigorous experimentation and statistical analyses revealed the significant impact of different SVM kernels and regularization techniques on key performance indicators such as accuracy, sensitivity, and specificity. Kernel selection emerged as crucial, with the linear kernel consistently demonstrating robust accuracy and sensitivity, while the polynomial kernel performed well in refined setups with some variability. Conversely, the radial kernel exhibited higher variability and instability, particularly in specificity, and the sigmoid kernel showed promising but inconsistent results. Penalized SVM models offered nuanced insights, where  $L_1$  penalties showed superior accuracy, though their effect on sensitivity and specificity was less pronounced. This study enhances the understanding of SVM-based classification methodologies, particularly for classification of cancer using gene expression data, and can guide the selection of appropriate SVM variants and kernels for specific tasks. Future research should explore ensemble techniques or hybrid models combining SVM with other machine learning algorithms to potentially enhance performance, and investigate the generalizability of these findings across different datasets and domains to ensure robustness and scalability in real-world applications.

# References

- [1] Taiwo Oladipupo Ayodele. Types of machine learning algorithms. *New advances in machine learning*, 3:19–48, 2010.
- [2] Francisco Azuaje. Interpretation of genome expression patterns: computational challenges and opportunities. *IEEE engineering in medicine and biology magazine: the quarterly magazine of the Engineering in Medicine & Biology Society*, 19(6): 119, 2000.
- [3] Joseph DeRisi, Lolita Penland, ML Bittner, PS Meltzer, M Ray, Y Chen, YA Su, and JM Trent. Use of a cdna microarray to analyse gene expression. *Nat. genet.*, 14:457–460, 1996.
- [4] Todd R Golub, Donna K Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P Mesirov, Hilary Coller, Mignon L Loh, James R Downing, Mark A Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537, 1999.
- [5] Mark Schena, Dari Shalon, Ronald W Davis, and Patrick O Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467–470, 1995.
- [6] Peng Guan, Desheng Huang, Miao He, and Baosen Zhou. Lung cancer gene expression database analysis incorporating prior knowledge with support vector machine-based classification method. *Journal of experimental & clinical cancer research*, 28(1):1–7, 2009.
- [7] Isabelle Guyon, Jason Weston, Susan Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3): 389–422, 2002.

## REFERENCES

- [8] Heping Zhang, Jae-Woo Ahn, Xiao Lin, and Chihyun Park. Gene selection using support vector machines with non-convex penalty. *Bioinformatics*, 22(2):206–214, 2006.
- [9] H. Alquran, I. A. Qasmieh, A. M. Alqudah, and A. S. Al-Adwan. The melanoma skin cancer detection and classification using support vector machine. In *2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 1–5. IEEE, 2017.
- [10] B Gopinath and N Shanthi. Support vector machine based diagnostic system for thyroid cancer using statistical texture features. *Asian Pacific Journal of Cancer Prevention*, 14(1):97–102, 2013.
- [11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.
- [12] Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5(Oct):1391–1415, 2004.
- [13] S. V. N. Vishwanathan, Nicol N. Schraudolph, and Alexander J. Smola. Step size adaptation in reproducing kernel hilbert space. *Journal of Machine Learning Research*, 7(40):1107–1133, 2006.
- [14] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.
- [15] Paul S. Bradley and Olvi L. Mangasarian. Feature selection via concave minimization and support vector machines. In *ICML*, volume 98, pages 82–90, 1998.
- [16] Anestis Antoniadis. Wavelets in statistics: A review. *Journal of the American Statistical Association*, 92(438):160–175, 1997.
- [17] Jorge J More and Stephen J Wright. *Optimization software guide*. SIAM, 1993.