

MLDL Practical 5

Name: Ishaan Khan
Class: D15C
Roll No: 29
Batch: B

Aim: To implement Support Vector Machine (SVM) for classification using a real-world dataset and evaluate model performance with hyperparameter tuning.

Dataset Source

Dataset Name: Titanic Survival Dataset

Platform: Kaggle

Dataset Link:

<https://www.kaggle.com/datasets/yasserh/titanic-dataset>

The Titanic dataset contains passenger information from the Titanic shipwreck and is used to predict whether a passenger survived.

Dataset Description

The Titanic dataset is a binary classification dataset.

Dataset Characteristics

- Number of instances: 891
- Number of features: 11
- Target variable: Survived
 - 1 → Survived
 - 0 → Did Not Survive

Feature Description

Important features include:

- Pclass – Passenger class (1, 2, 3)
- Sex – Gender of passenger
- Age – Age of passenger
- SibSp – Number of siblings/spouses aboard
- Parch – Number of parents/children aboard
- Fare – Ticket fare
- Embarked – Port of embarkation

Mathematical Formulation of SVM

Support Vector Machine (SVM) is a supervised learning algorithm that finds the optimal hyperplane to separate data points of different classes while maximizing the margin.

Linear Decision Function

$$f(x) = w^T x + b$$

Classification rule:

$$y = \text{sign}(w^T x + b)$$

Where:

- w = weight vector
- x = feature vector
- b = bias

Optimization Objective

SVM minimizes:

$$(1/2) \|w\|^2 + C \sum \xi_i$$

Subject to:

$$y_i (w^T x_i + b) \geq 1 - \xi_i$$

Where:

- C = Regularization parameter
- ξ_i = Slack variables (allow misclassification)

Kernel Function (RBF)

$$K(x, x') = \exp(-\gamma \|x - x'\|^2)$$

Where:

- γ controls the influence of a single training example

Algorithm Limitations

- Computationally expensive for large datasets

- Sensitive to hyperparameter selection
- Requires proper feature scaling
- Less interpretable compared to tree-based models

Methodology / Workflow

The experiment followed these steps:

1. Load dataset using KaggleHub
2. Handle missing values
3. Encode categorical variables
4. Drop irrelevant columns
5. Perform train-test split (80:20)
6. Apply feature scaling using StandardScaler
7. Train SVM classifier
8. Perform hyperparameter tuning using GridSearchCV
9. Evaluate model performance

Workflow Diagram

Dataset → Cleaning → Encoding → Scaling → SVM Training → Hyperparameter Tuning → Evaluation

Performance Analysis

The SVM model was evaluated using:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix
- ROC Curve
- AUC Score

After hyperparameter tuning, the SVM achieved strong classification performance. The confusion matrix showed reduced misclassification errors, and the ROC curve demonstrated good class separation capability.

Hyperparameter Tuning

Hyperparameter tuning was performed using GridSearchCV.

Parameters Tuned

- C: [0.1, 1, 10, 100]
- Kernel: Linear, RBF
- Gamma: scale, auto

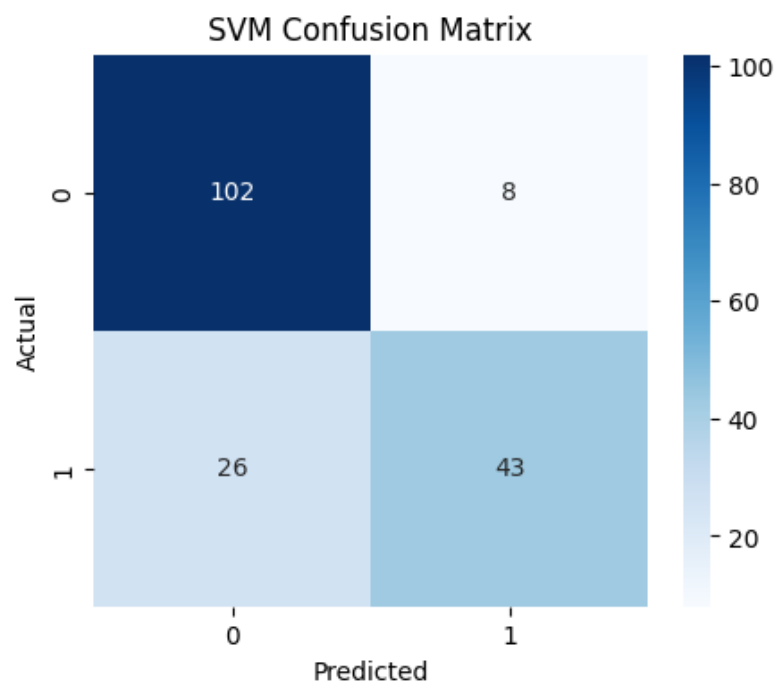
Impact of Tuning

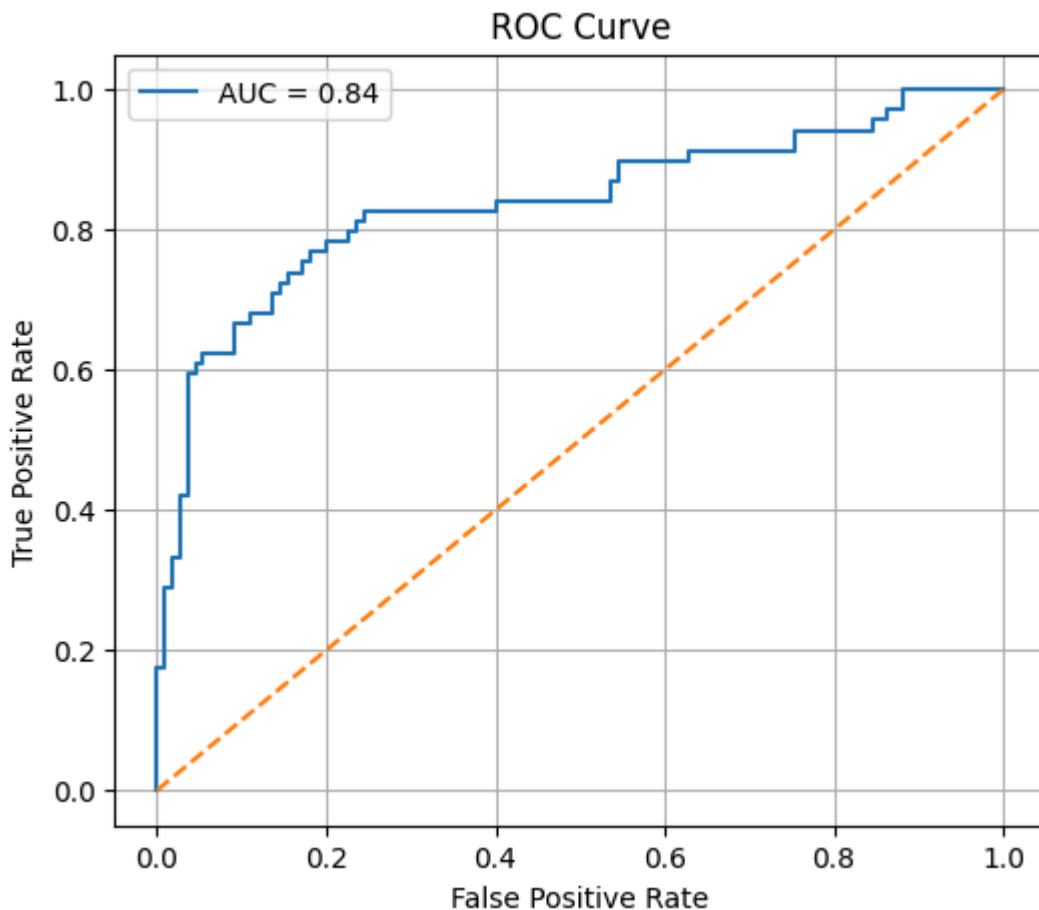
- Low C → Higher bias, possible underfitting
- High C → Lower bias, possible overfitting
- RBF kernel → Captures non-linear patterns
- Optimal gamma improves decision boundary flexibility

Tuning significantly improved classification accuracy and generalization.

Output

=== SVM Performance ===					
Accuracy: 0.8100558659217877					
	precision	recall	f1-score	support	
0	0.80	0.93	0.86	110	
1	0.84	0.62	0.72	69	
accuracy			0.81	179	
macro avg	0.82	0.78	0.79	179	
weighted avg	0.81	0.81	0.80	179	





Conclusion

In this experiment, Support Vector Machine was successfully implemented on the Titanic survival dataset. After proper preprocessing and hyperparameter tuning, SVM achieved high predictive performance.

This experiment highlights:

- The importance of feature scaling
- The impact of kernel selection
- The necessity of hyperparameter tuning
- The effectiveness of SVM in real-world classification tasks

SVM is a powerful classification algorithm and performs especially well when the data is moderately complex and properly preprocessed.