Comparative Analysis of ML Algorithms

Submitted by: GAUTAM KRISHNA
Registration No: 12214473

Program: Bachelor of Technology in Computer Science & Engineering
Section: K22MK
Course Code: INT234
Under the Guidance of Mr. Vikas Mangotra (UID: 31488)
Discipline of CSE/IT

Lovely School of Computer Science & Engineering

Lovely Professional University, Phagwara

CERTIFICATE

This is to certify that Gautam Krishna bearing Registration no. 12214473 has completed INT234 project titled, “COMPARATIVE ANALYSIS OF ML ALGORITHMS” under guidance and supervision of Mr. Vikas Mangotra. To the best of my knowledge, the present work is the result of his original development, effort and study.

Signature:

Name of the Supervisor: Vikas Mangotra

Designation of the Supervisor: Assistant Professor

School of Computer Science and Engineering

Lovely Professional University Phagwara, Punjab.

Date: 17.11.2024

DECLARATION

I, Gautam Krishna, student of Bachelor of Technology in Computer Science and Engineering under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 17.11.2024

Signature: Gautam Krishna

Registration No: 12214473

Name of the student: Gautam Krishna

ACKNOWLEDGEMENT

I would like to express my profound gratitude to Mr. Vikas Mangotra sir of the School of Computer Science & Engineering department, Lovely Professional University, for their contributions to the completion of my project titled “Comparative Analysis of ML algorithms”. I would like to express my special thanks to him, for his time and efforts throughout the year. His useful advice and suggestions were helpful to me during the project’s completion. In this aspect, I am eternally grateful to him. I would like to acknowledge that this project was completed entirely by me and not by anyone else.

TABLE OF CONTENTS

INTRODUCTION

The Sonar dataset is a famous dataset in the field of machine learning and data science, consisting of sonar signals reflected off a metal cylinder and a rock. The challenge is to classify these sonar signals into two categories: "rock" and "metal." The dataset includes 60 continuous features, each representing a particular characteristic of the sonar signal.

In this project, we applied multiple machine learning algorithms to this classification problem to identify the most accurate and effective model. These algorithms include K-Nearest Neighbors (KNN), Decision Trees, Support Vector Machines (SVM), and Naive Bayes. The analysis of this dataset is crucial as it simulates real-world problems, such as sonar systems used in marine exploration, where classifying objects based on reflected signals is essential.

The overall goal was to analyze how well each model could predict the class of a sonar signal and to understand the strengths and weaknesses of each algorithm in dealing with real-world data. The project also aimed to explore how preprocessing steps, like normalization and feature scaling, affected model performance.

Here are the four machine learning algorithms applied in the analysis:

1. **K-Nearest Neighbors (KNN)**: A non-parametric method used for classification based on feature similarity. The algorithm classifies a new data point by looking at the most common class among the nearest data points.
2. **Decision Tree**: A supervised learning algorithm that splits data into subsets using a tree-like model of decisions based on feature values. It is highly interpretable and easy to understand.
3. **Support Vector Machine (SVM)**: A classification algorithm that finds the optimal separating hyperplane between classes by maximizing the margin between data points of different classes.
4. **Naive Bayes**: A probabilistic classifier based on Bayes' theorem, which assumes that the features are independent, making it computationally efficient even with a large number of feature

K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is an instance-based learning algorithm, meaning it makes predictions based on the closest data points in the feature space. To classify a new data point, KNN identifies the *k* nearest neighbors and assigns the class that is most frequent among them. The value of *k* is critical to the model's performance: a small *k* may lead to a model that is too sensitive to noise, while a large *k* could oversimplify the model, making it less responsive to patterns in the data.

Decision Tree

A Decision Tree is a flowchart-like model where internal nodes represent features, branches represent decision rules, and leaf nodes represent outcomes or classes. Starting from the root node, the tree splits the data based on features that maximize a specific criterion, such as Information Gain or Gini Impurity. This process continues until a stopping condition or maximum tree depth is reached. Decision Trees are highly interpretable, as they clearly show the decision-making process. However, they can suffer from overfitting if not properly pruned, capturing noise in the data as part of the model.

Support Vector Machine (SVM)

Support Vector Machine (SVM) is a classification algorithm that aims to find the optimal hyperplane that maximizes the margin between classes in the feature space. For linearly separable data, SVM constructs a straight hyperplane. For non-linear data, SVM uses kernel functions (such as linear or radial basis function) to project the data into a higher-dimensional space, where the data can be more easily separated. The margin maximization makes SVM robust to minor variations and noise in the data. While SVM with a linear kernel works well for linearly separable datasets, exploring non-linear kernels could improve performance when dealing with complex, non-linear data patterns.

Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem, assuming that all features are conditionally independent given the target class. It calculates the probability of each class given the features and selects the class with the highest probability. This makes Naive Bayes efficient and well-suited for datasets with noise or small sample sizes. While it performed reasonably well in this analysis, it is often outperformed by other models in more complex datasets due to its simplifying assumption of feature independence, which may not always hold true in practice.

Summary of the Algorithms

Algorithm	Strengths	Limitations
KNN	Simple to understand and effective on small datasets	Sensitive to feature scaling and value of k
Decision Tree	Easy interpretability and visualizable	Tends to overfit without pruning
SVM	Robust with good margin maximization	Computationally intensive, sensitive to feature scaling
Naive Bayes	Efficient, works well with noisy data	Assumes feature independence, limiting performance

Scope of Analysis

This analysis explores the ability of four supervised machine learning algorithms (KNN, Decision Trees, SVM, and Naive Bayes) to classify sonar signals as either rock or metal. Each algorithm is implemented and evaluated on the Sonar dataset to compare their effectiveness and robustness in classification tasks.

The key objectives of this project are as follows:

- **Model Performance Evaluation**: To compare the performance of KNN, Decision Trees, SVM, and Naive Bayes based on several key metrics such as accuracy, precision, recall, F1 score, and error rate. These metrics provide a comprehensive understanding of how well each model predicts the correct class and how they balance true positives, false positives, and false negatives.

- **Feature Scaling and Preprocessing**: To analyze the importance of preprocessing steps, particularly feature scaling. Algorithms like KNN and SVM are sensitive to the scale of the data, making it crucial to normalize the features to a consistent range (e.g., 0-1) for better performance.

- **Model Comparison**: To compare the strengths and weaknesses of each model and determine the most appropriate model for classifying sonar signals. The decision-making process involves assessing the model's accuracy, its computational complexity, and its interpretability.

- **Real-World Application**: By evaluating the effectiveness of different algorithms on this dataset, the project also lays the groundwork for applying these techniques in real-world applications, such as underwater exploration, where sonar data classification plays a pivotal role.

OBJECTIVES

Evaluate and Compare the Performance of ML Algorithms

To implement and evaluate four machine learning algorithms (KNN, Decision Tree, SVM, and Naive Bayes) on the Sonar dataset and compare their performance in terms of accuracy and classification metrics.

- **Optimize and Interpret Model Results**

To optimize hyperparameters of each algorithm and analyze their interpretability, particularly the Decision Tree, to determine the most effective model for sonar data classification.

ANALYSIS OF DATASET

The Sonar dataset consists of 208 instances, each representing a sonar signal with 60 features. These features represent various signal characteristics measured at different intervals. The goal is to predict whether a sonar signal corresponds to a rock or a metal cylinder.

Dataset Structure:

- **Features**: 60 continuous numerical features describing the sonar signal's reflection characteristics.
- **Target Variable**: The class label, which can either be "Rock" or "Metal." This is a binary classification task.

Exploratory Data Analysis (EDA): Before applying machine learning models, we first conducted an EDA to understand the structure and distribution of the data. This helped identify key characteristics of the dataset and potential issues that could impact model performance.

- **Class Distribution**: The dataset is relatively balanced with an almost equal distribution of 'rock' and 'metal' labels, which makes it suitable for classification tasks without severe class imbalance.
- **Feature Correlation**: We analyzed the correlation between different features to identify which ones are most relevant for classification. Features with high correlation may cause multicollinearity and could affect some algorithms.
- **Missing Data**: The dataset does not contain any missing values, so we did not need to perform imputation.

Data Preprocessing:

- **Normalization**: We normalized the data to a range of 0 to 1 using Min-Max scaling to ensure that all features contribute equally to the classification process, particularly for algorithms like KNN and SVM.
- **Train-Test Split**: The dataset was divided into a training set (80% of the data) and a test set (20% of the data). This allowed us to train the models on a portion of the data and evaluate them on unseen data to check for generalization.

Comparison of Algorithms

We implemented four machine learning algorithms and evaluated their performance on the Sonar dataset. The models were compared based on their ability to accurately classify sonar signals as either "rock" or "metal." The following evaluation metrics were used:

- **Accuracy**: The proportion of correctly classified instances out of all instances.
- **Precision**: The proportion of correctly identified positive instances out of all predicted positive instances.
- **Recall**: The proportion of correctly identified positive instances out of all actual positive instances.
- **F1 Score**: The harmonic mean of precision and recall, providing a balanced view of the model's performance.
- **Error Rate**: The proportion of incorrect predictions out of all predictions.

Performance Overview:

- **K-Nearest Neighbors (KNN)**:
  KNN performed well, showing high accuracy. However, its performance was sensitive to noisy data and outliers. This is because KNN relies on the proximity of data points, which can be affected by the presence of irrelevant or redundant features. We experimented with different values of 'k' (number of nearest neighbors) to find the optimal parameter.
- **Decision Tree**:
  The Decision Tree model was easy to interpret, but it showed a tendency to overfit the training data. Overfitting occurs when the tree becomes too complex, capturing noise rather than the underlying patterns in the data. To address this, we applied pruning techniques to reduce the tree size and improve generalization.
- **Support Vector Machine (SVM)**:
  The SVM model showed the highest performance overall. It achieved a high level of accuracy, precision, and recall. SVM is particularly well-suited for classification tasks with clear margins between classes and can handle high-dimensional data effectively.
- **Naive Bayes**:
  The Naive Bayes algorithm performed reasonably well but was less accurate compared to SVM. This is expected because Naive Bayes assumes feature independence, which may not hold true in complex datasets like Sonar, where features can exhibit dependencies.

```
# Load required libraries
library(class)
library(rpart)
library(rpart.plot)
library(e1071)
library(mlbench)

# Load and preprocess Sonar dataset
data(Sonar)
sonar_data <- Sonar
sonar_data$Class <- factor(sonar_data$Class, levels = c("R", "M"))
normalized_data <- scale(sonar_data[, -61])  # Exclude Class column
normalized_data <- as.data.frame(normalized_data)
normalized_data$Class <- sonar_data$Class

# Data splitting
set.seed(123)
split <- sample(2, nrow(normalized_data), replace = TRUE, prob = c(0.8, 0.2))
train_data <- normalized_data[split == 1, ]
test_data <- normalized_data[split == 2, ]

# Model function to compute accuracy
model_accuracy <- function(model, train_data, test_data) {
  pred <- predict(model, test_data, type = "class")
  confusion_matrix <- table(test_data$Class, pred)
  accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
  return(accuracy)
}

# KNN
knn_predicted <- knn(train = train_data[, -61], test = test_data[, -61],
                     cl = train_data$Class, k = 5)
knn_accuracy <- mean(knn_predicted == test_data$Class)

# Decision Tree
tree_model <- rpart(Class ~ ., data = train_data, method = "class")
rpart.plot(tree_model)
tree_accuracy <- model_tree_accuracy <- model_tree_accuracy <-
model_accuracy(tree_model, train_data, test_data)
```

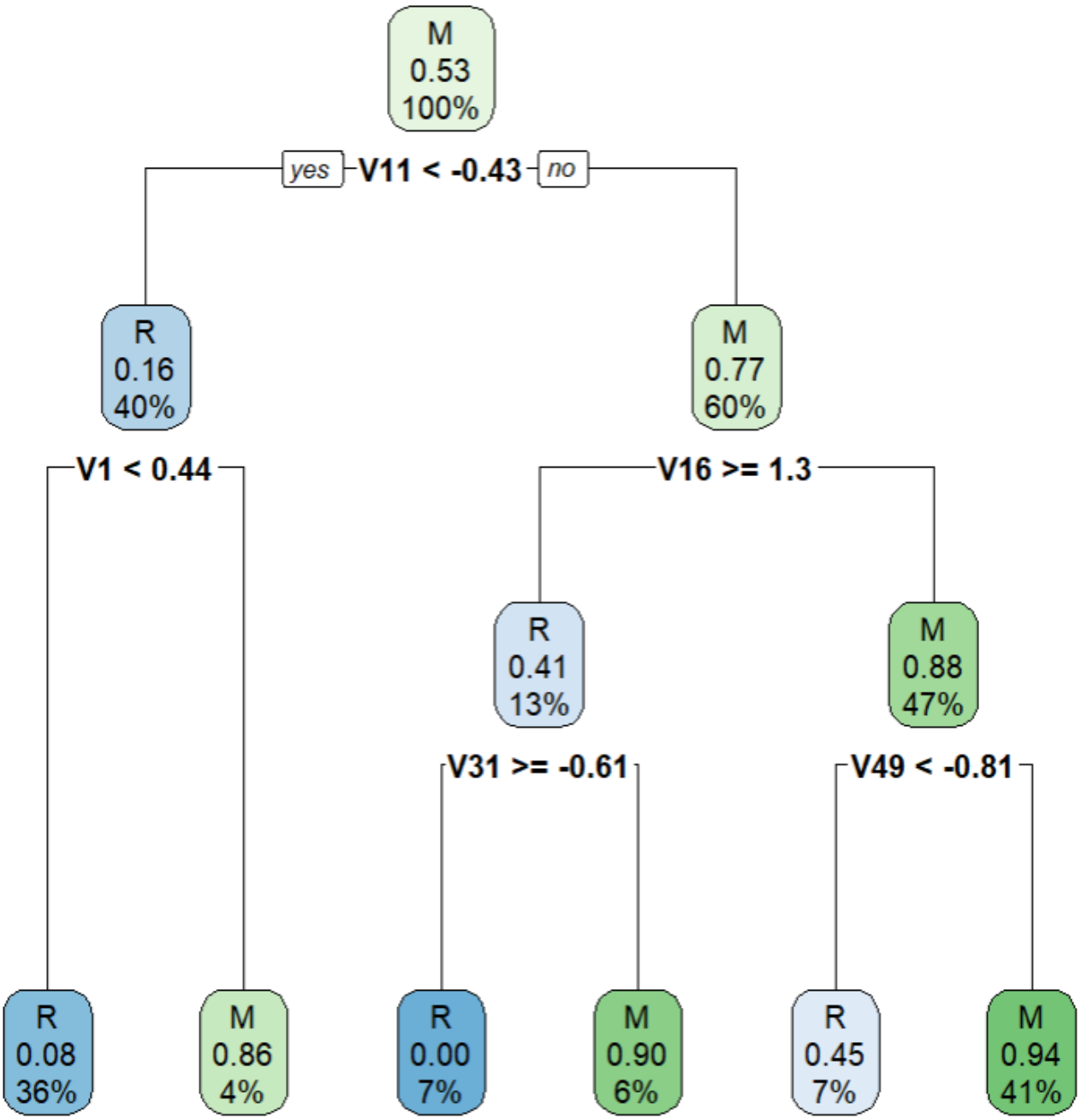
```
# SVM
svm_model <- svm(Class ~ ., data = train_data, kernel = "linear")
svm_accuracy <- model_accuracy(svm_model, train_data, test_data)

# Naive Bayes
nb_model <- naiveBayes(Class ~ ., data = train_data)
nb_accuracy <- model_accuracy(nb_model, train_data, test_data)

# Print all accuracies
accuracies <- c(KNN = knn_accuracy, DecisionTree = tree_accuracy, SVM = svm_accuracy,
NaiveBayes = nb_accuracy)
print(accuracies)

# Find and print the algorithm with the highest accuracy
best_algorithm <- names(accuracies)[which.max(accuracies)]
cat("The algorithm with the highest accuracy is:", best_algorithm, "with an accuracy of",
round(max(accuracies), 4))
```

Results for Decision Tree Algorithm



Accuracies obtained

```
> print(accuracies)
        KNN DecisionTree          SVM   NaiveBayes
  0.8205128    0.6410256    0.6923077    0.4871795
>
```

Final Output

```
> best_algorithm ← names(accuracies)[which.max(accuracies)]
> cat("The algorithm with the highest accuracy is:", best_algorithm, "with an
round(max(accuracies), 4))
The algorithm with the highest accuracy is: KNN with an accuracy of 0.8205
```

Link to Files : Click Here

CONCLUSION

After applying and comparing the four machine learning algorithms (KNN, Decision Tree, SVM, and Naive Bayes) on the Sonar dataset, we found that:

- **SVM** provided the best overall performance, achieving the highest accuracy, precision, and recall.
- **KNN** showed strong performance but was more sensitive to noise in the data.
- **Decision Tree** was interpretable but suffered from overfitting, which required pruning for optimal performance.
- **Naive Bayes** was computationally efficient but lagged in accuracy compared to the other models.

This analysis highlights the importance of selecting the right algorithm for specific problems and the need for proper data pre-processing to enhance model performance. In real-world scenarios, SVM might be the preferred choice for its high performance, but Decision Trees and KNN are also viable options depending on the context and data characteristics.