University of Glasgow

School of Computing Science

Machine Learning& AI Case Study 3

Siyu Na,	2931267N
Xulong Yang,	2870692Y
Xinghan Cheng	2938618C

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfillment of the requirements of the Degree of Master of Science at the University of Glasgow

29 November 2023

1 Background

Spinal Cord Injury (SCI) affects about half of the patients with Central Neuropathic Pain (CNP), causing unexpected pain sensations like electric shocks or numbness[1]. There's no direct treatment, only preventive medications with strong side effects. Predicting who might develop this pain is crucial for targeted treatment, but current manual assessments are time-consuming and somewhat unreliable.

Brain EEG data might hold clues[2], but it's high-dimensional and challenging for classifiers to distinguish between SCI patients who later develop CNP and those who don't[3]. There's potential in using advanced machine learning techniques to uncover patterns in the EEG data that indicate the likelihood of CNP development. In that case, this could revolutionize early intervention strategies for SCI-related pain.

2 Introduction

This task is to devise a feature engineering strategy that, in combination with a classifier of the choice, optimizes prediction accuracy.

At the data acquisition stage, three feature selection methods are compared , including filtering methods, wrapper methods, and embedding methods. Each method with classifiers is combined, including SVM (SVM, SVM-Gaussian, SVM-Polynomial), KNN, Random Forest(RF), Naive Bayes, and Logistic Regression. By using several different classifiers, it is possible to see which model has the best fit in the context of this question.

In the validation phase, Leave-one-group-out cross-validation is used, which means all samples from a particular group or cluster are excluded from the training set in each fold, while the model is trained on the remaining data and tested on the excluded group[4]. In this case, nine of ten samples are used as prediction samples, and the remaining one is used as a validation sample. Besides, cross-validation is selected to optimize hyperparameter values for a better fit of the classifier. In this case, GridSearchCV is chosen to optimize the hyperparameter. GridSearchCV creates a grid of hyperparameter values to explore[5]. It specifies the hyperparameters to tune and their possible values. For each combination of hyperparameters in the grid, it conducts cross-validation (often k-fold cross-validation) on the training data to assess the model's performance.

Finally, each of the classifiers trained on the full set of features is evaluated for comparison, and final conclusions are drawn.

3 Methods

3.1 Feature Selection Methods：

The dataset for the given case study consists of EEG data from 18 participants. The data has been preprocessed and normalized, resulting in a large number of features. This high dimensionality can lead to computational inefficiencies and overfitting. The goal of feature selection is to identify a subset of features that minimizes error while maintaining predictive power. However, finding the optimal subset is considered an NP-hard problem, necessitating the use of efficient heuristic methods. This paper will explore three commonly used techniques: filtering, wrappers, and embedding. [6]

3.1.1 Filtering Methods

The filtering approach involves the use of surrogate measures to assess the relevance of individual features[7]. This method evaluates each feature independently, without considering their interactions. By applying statistical tests or correlation measures, irrelevant or redundant features can be identified and removed. This technique is computationally efficient but may overlook important feature combinations.

3.1.2 Wrapper Methods

The wrapper approach takes a more proactive approach by evaluating feature subsets using a predictive model. It involves building feature sets iteratively and assessing their performance using a chosen machine learning algorithm. This method considers feature interactions and can provide better results compared to filtering techniques[8]. However, it is computationally expensive due to the need for repeated model evaluations.

3.1.3 Embedding Methods

The embedding technique incorporates feature selection directly into the learning process of a predictive model. Methods like LASSO (Least Absolute Shrinkage and Selection Operator) use regularization to penalize irrelevant features, effectively forcing their coefficients to zero. This approach not only selects relevant features but also performs model training simultaneously. However, it may not be suitable for datasets with highly correlated features.

3.2 Classifiers Methods：

3.2.1 SVM

Support Vector Machine (SVM) is a powerful supervised learning algorithm commonly used for classification and regression problems. It is capable of effectively handling both linear and nonlinear data by constructing optimal hyperplanes or curves in high-dimensional spaces.

The core idea behind SVM is to find an optimal decision boundary that separates the sample points into different classes. SVM achieves effective classification by maximizing the margin between the decision boundary and the nearest sample points, known as support vectors.

In SVM, several methods and techniques are widely employed:

1. Linear SVM: Linear SVM is used for linearly separable datasets, where it constructs a linear hyperplane to separate the samples into different classes. The objective of this hyperplane is to maximize the distance between the support vectors and the boundary, thus achieving the best classification performance.

2. Nonlinear SVM: Nonlinear SVM is applied to datasets with nonlinear relationships. It utilizes kernel functions to map the data into higher-dimensional spaces, where it becomes linearly separable. Commonly used kernel functions include polynomial kernel and radial basis function (RBF) kernel.

3. Multiclass Classification: SVM can also be used for multiclass classification problems. One popular approach is the one-vs-rest method, which transforms the multiclass problem into multiple binary classification problems. The results of these binary classifiers are then combined to make the final classification decision.

4. Parameter Tuning: Parameter tuning plays a crucial role in the performance and generalization of SVM models. Common parameters include the regularization parameter C and the parameters of the kernel function. By appropriately adjusting these parameters, better classification results can be achieved.

3.2.2 KNN

KNN is a simple and versatile classification algorithm that is widely used in machine learning. It operates based on the principle that similar data points tend to belong to the same class. In KNN, the classification of a new data point is determined by the majority vote of its k nearest neighbors in the feature space. The value of k is a hyperparameter that needs to be specified. KNN can handle both binary and multi-class classification problems and can be easily implemented.

3.2.3 Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. Each tree in the random forest is built on a different subset of the training data and uses a random subset of features for splitting. The final prediction is made by aggregating the predictions of all the trees. Random Forest is known for its ability to handle high-dimensional and noisy data, as well as its resistance to overfitting. It is widely used for classification tasks and provides measures of feature importance.

3.2.4 Naive Bayes

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem with the assumption of independence between features. It is called "naive" because it assumes that all features are conditionally independent given the class label. Naive Bayes calculates the probability of a data point belonging to each class and selects the class with the highest probability as the prediction. It is computationally efficient and works well with high-dimensional data.

3.2.5 Logistic Regression

Logistic Regression is a widely used classification algorithm that models the relationship between the independent variables and the probability of a binary outcome. It estimates the parameters of a logistic function to predict the probability of a data point belonging to a specific class. Logistic Regression is relatively simple and interpretable, and it can handle both numerical and categorical features. Regularization techniques, such as L1 or L2 regularization, can be applied to prevent overfitting.

3.3 Validation Methods：

In order to evaluate the performance of the classifiers and determine the optimal hyper-parameter values, a common approach is to use cross-validation. Cross-validation is a technique that assesses the model's ability to generalize to unseen data by splitting the dataset into training and validation subsets. One widely used cross-validation method is leave-one-out cross-validation (LOOCV), where each data point is used as a validation sample once while the remaining samples are used for training[9].

3.3.1 Leave-One-Out Cross-Validation

Leave-One-Out Cross-Validation (LOOCV) is a cross-validation technique where the model is trained on all samples except one, which is used for validation. This process is repeated for each data point in the dataset, resulting in a set of accuracy, sensitivity, and specificity values for each classifier.

3.3.2 Baseline for Comparison

To establish a baseline for comparison, each classifier is evaluated on the full set of features without feature selection. This allows for an assessment of the classifiers' performance when all features are included in the model. By comparing the results of the classifiers trained without feature selection to those with feature selection, the impact of feature selection on the classifiers' performance can be determined.

4 Design:

4.1 Preprocessing Of The Dataset：

Firstly, data will be read from the given CSV file based on the research data of existing cases. The data description of the CSV file is as follows:

- data.csv: This dataset is structured as a matrix with dimensions 180x432. The 180 rows correspond to the 18 participants, with each participant contributing data from 10 trials. The 432 columns represent measurements from 48 electrodes for 9 specific features. For example, the first 10 rows correspond to the data from the first participant, and the first 48 columns correspond to the measurements from the first electrode for the feature "alpha_ec."

- feature_names.csv: This file contains a one-dimensional vector with 432 entries. Each entry corresponds to the name and electrode number of a specific feature in the "data.csv" file. These feature names provide a description of the measurements taken from each electrode.

- labels.csv: This file consists of a one-dimensional vector with 180 entries. The vector assigns a category to each participant, indicating their respective group. Based on the background research review, the first 10 participants belong to the group of individuals who experience recurring symptoms (labeled as 0), while the remaining 8 participants belong to the control group without significant pain sensations (labeled as 1).

According to the categories in the "labels.csv" file, the data will be initially divided into two groups based on priority, in order to facilitate subsequent cross-validation comparisons.

The obtained data is visualized as shown in the following figure:

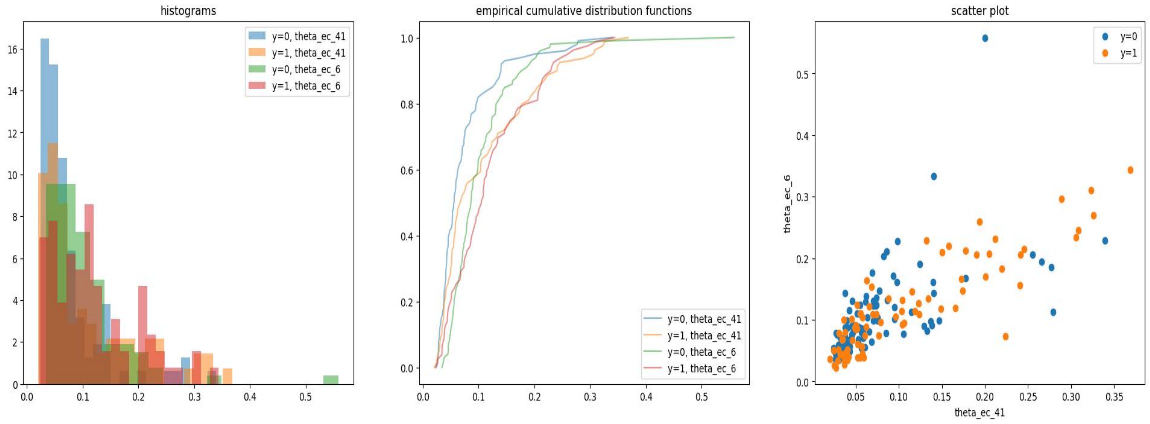

Figure 4.1 Samples of Data

4.2 Feature Selection

In this study, the three feature selection methods are filter, wrapper, and embedded methods. The purpose of these methods is to select the most informative subset of features from a given feature set.

For the filter method, at least one scoring function is utilized to measure the correlation between each feature and the target variable. By computing scores for each feature, they can be ranked, and the top-scoring ones are selected. In this study, the f_classif scoring function and the SelectKBest selector are employed to choose the top 10 features as the filtered subset.

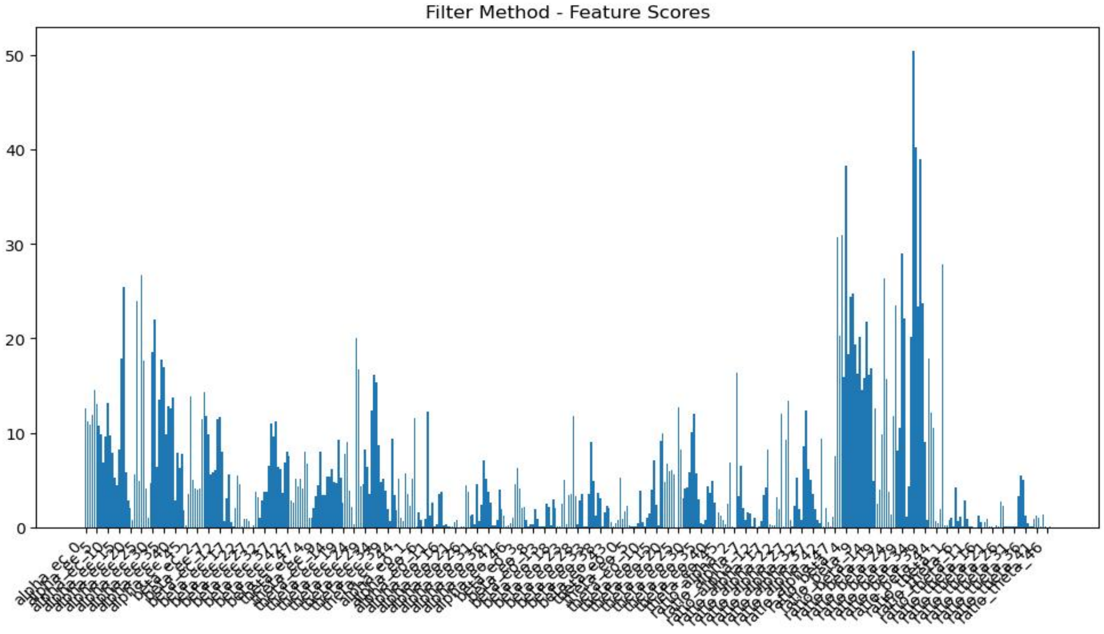

Figure 4.2 Feature Scores in Filter Method

In the wrapper method, techniques such as forward feature selection or backward feature elimination are employed. Forward feature selection starts with an empty feature set and gradually adds features until reaching a specified number or achieving a certain performance criterion. On the other hand, backward feature elimination begins with the complete feature set and progressively eliminates the least relevant features until reaching a specified number or achieving a certain performance criterion. Through these wrapper methods, the optimal feature subset that maximizes the performance of the classification or regression model can be identified.

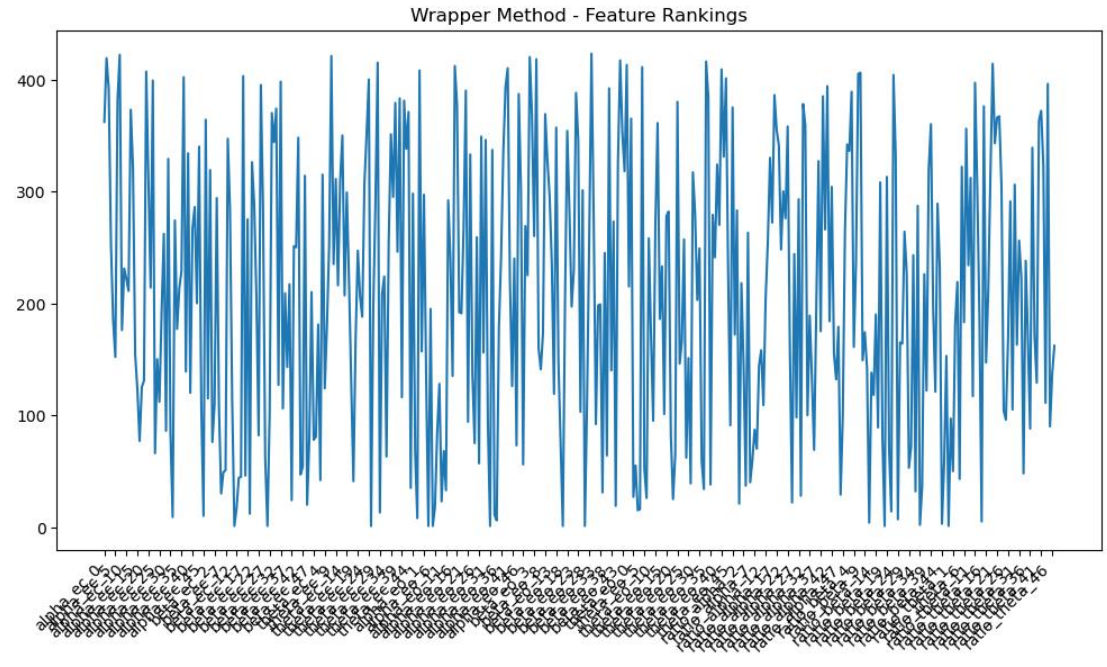

Figure 4.3 Feature Rankings in Wrapper Method

The embedded method is a technique that combines feature selection with the model training process. In this study, L1 regularization is utilized to select the most informative features during the model training. By employing a linear support vector classifier (LinearSVC) with L1 regularization, the final feature subset can be obtained by selecting the features with non-zero coefficients.

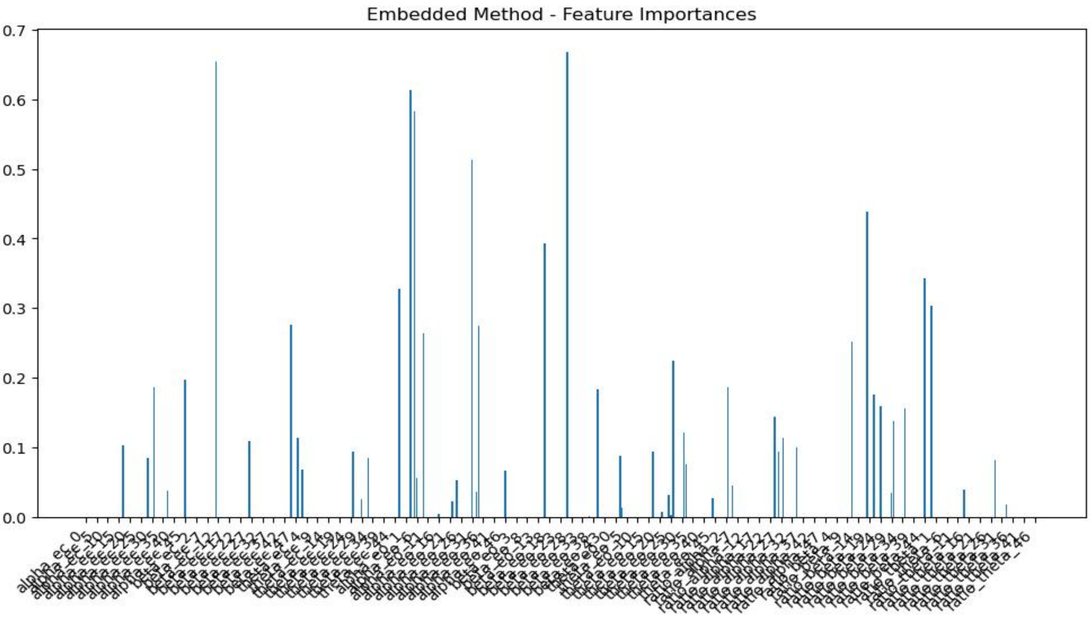

Figure 4.4 Feature Importances in Embedded Method

By comparing these feature selection methods, it is important to evaluate their performance and effectiveness in selecting the best feature subset. This could help to determine the most suitable feature selection method for our study and provide insights into the relationship between the selected feature subset and model performance. The ultimate outcome is that all three methods yield a subset of 10 features.

4.3 Classifier

The problem in this case study is a binary issue and the next step involves determining the classification strategy based on the selected feature values. Considering the classifier methods been chosen previously, comparing these classifiers horizontally is necessary to assess their strengths and weaknesses initially. Then adopting the same strategy to obtain the optimal parameters is in consideration. The number of features obtained in the previous steps is consistently 10, which significantly reduces the complexity for the classifiers. The table below illustrates the seven selected classifiers (SVM accounting for three categories[10]).

Classifier	Advantage	Disadvantage
SVM (RBF/Gaussian /Polynomial)	1. can use different kernel 2. better in prevent overfitting 3. suitable for high-dimensional data.	1. long training time 2. requires proper selection of kernel functions
KNN	1. easy to understand 2. better in non-linear problems. 3. real-time predictions	1. bad in imbalanced datasets 2. high computational cost in high-dimensional datasets
Random Forest	1. better in large-scale datasets 2. handle non-linear problems 3. Robust to missing data and imbalanced datasets.	1. bad in linear problems 2. prone to overfitting with noisy data
Naive Bayes	1. simple and fast, easy 2. better in texting problems 3. handle multi-class classification problems	1. needs feature independence 2. bad in non-linear problems
Logistic Regression	1. fast computation, suitable for large-scale datasets 2. better in linear problems 3. model complexity can be controlled through regularization.	1. bad in non-linear problems 2. may be affected by noisy data

Table 4.5 Selected Classifiers

For the adopted strategy, training and prediction for each model will be done sequentially based on feature values. Each model is trained in a relative manner according to the following procedure:

- Define a feature selector that utilizes the three different feature selection methods mentioned earlier: filtered, wrapped, and embedded. Each selector employs a distinct set of features for the selection process.

- Construct a multi-step pipeline which is needed to GridSearch methods
- Define hyperparameter grids
- Utilize GridSearchCV for hyperparameter optimization
- Perform grid search training on the scaled input data and target variable, aiming to discover the optimal hyperparameter combination.
- Retrieve the estimator of the best model

4.3.1 Pipeline

A Pipeline is a tool for seamlessly linking multiple data processing steps to construct an integrated data processing workflow. Through a Pipeline, it is quite easy to combine various steps such as feature scaling, feature selection, and model training in a sequential manner, creating a continuous data processing flow.[11] The primary reason for using a Pipeline is its ability to streamline workflows, enhance code readability, and improve maintainability. By chaining the steps together, Experimental results can be easily reproduced, avoiding data leakage, and aiding hyperparameter optimization.

Firstly, multiple data processing steps are defined, including the selection of a standard scaler, utilization of three types of previously processed feature selection as the "features" parameter, and the choice of different classifiers based on the model. These steps are then combined sequentially to form a Pipeline. Each step takes the output of the previous one as input and performs the corresponding data processing.

4.3.2 Hyperparameter

Hyperparameters are adjustable parameters in machine learning models. Unlike the model's inherent parameters, hyperparameters cannot be learned directly from the training data and need to be manually set. Therefore, parameter grids are employed as a tool for hyperparameter optimization. By defining a parameter grid, the hyperparameters to be searched and their possible value ranges are specified[12]. Each model has its own set of parameters, as shown in the table below:

Classifier	Parameter
SVM	classifier__C, classifier__kernel, classifier__degree, classifier__gamma
KNN	classifier__n_neighbors, classifier__weights
Random Forest	classifier__n_estimators, classifier__max_depth
Naive Bayes	classifier__priors
Logistic Regression	classifier__C, classifier__penalty

Table 4.6 Parameters in Each Classifier

GridSearchCV is a technique used for hyperparameter optimization. It is based on a grid search and cross-validation approach, systematically exploring different combinations in the hyperparameter space and evaluating the performance of each combination through cross-validation. In the previous step, after obtaining the key-values for hyperparameters, GridSearchCV is used as the search tool. Param Grid is provided as input, and it automatically searches and evaluates all possible hyperparameter combinations, identifying the optimal combination. This process helps alleviate the burden of manual hyperparameter tuning. [13]

4.4 Best Paramaters For Classifiers

The final optimal parameters obtained through grid search are as follows:

Classifier	Parameter
SVM(RBF)	{'classifier__C': 10, 'classifier__kernel': 'rbf', 'features__filtered__k': 10}
SVM(Poly)	{'classifier__C': 10, 'classifier__degree': 2, 'features__filtered__k': 10}
SVM(Gaussian)	{'classifier__C': 10, 'classifier__gamma': 0.1, 'features__filtered__k': 10}
KNN	{'classifier__n_neighbors': 11, 'classifier__weights': 'uniform', 'features__filtered__k': 10}
Random Forest	{'classifier__max_depth': 10, 'classifier__n_estimators': 200, 'features__filtered__k': 10}
Naive Bayes	{'classifier__priors': None, 'features__filtered__k': 10}
Logistic Regression	{'classifier__C': 1, 'classifier__penalty': 'l2', 'features__filtered__k': 10}

Table 4.7 Hyper-Parameters in Each Classifier

5 Result

5.1 Performance Metrics

The quantification of their basic performance is shown by these metrics[14]:

- Accuracy: Accuracy measures the proportion of correctly predicted samples by the model, indicating the overall correctness of the model's classification.
- Precision: Precision is the proportion of true positive samples among all samples predicted as positive by the model, measuring the accuracy of the model when predicting positive instances.
- Recall: Recall is the proportion of true positive samples among all actual positive instances, measuring the model's ability to correctly identify positive instances.
- F1 Score: The F1 Score is the weighted harmonic mean of precision and recall, providing a comprehensive metric that combines the accuracy and recall of the model. It offers an overall assessment of the model's performance.

The corresponding graphical representation is as follows:

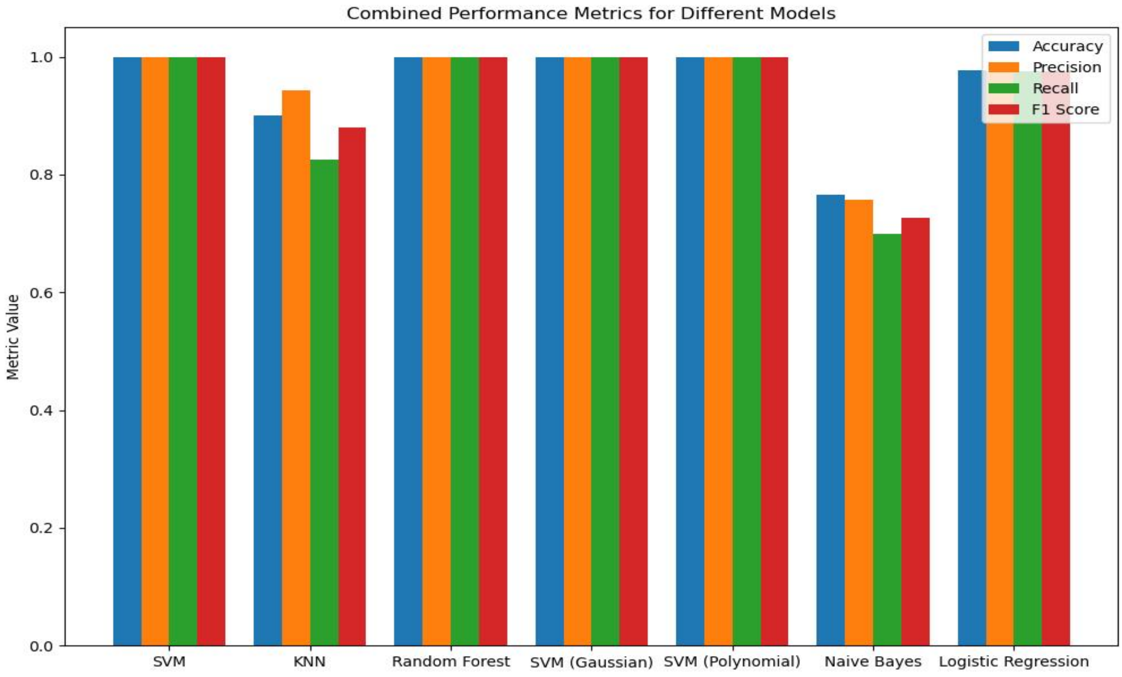

Figure 5.1 Performance Metrics for Different Models

5.2 Confusion Matrix

The Confusion Matrix is a tabular representation used to evaluate the performance of a classification model[15]. It is based on the correspondence between predicted results and true labels, categorizing samples into four different categories: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

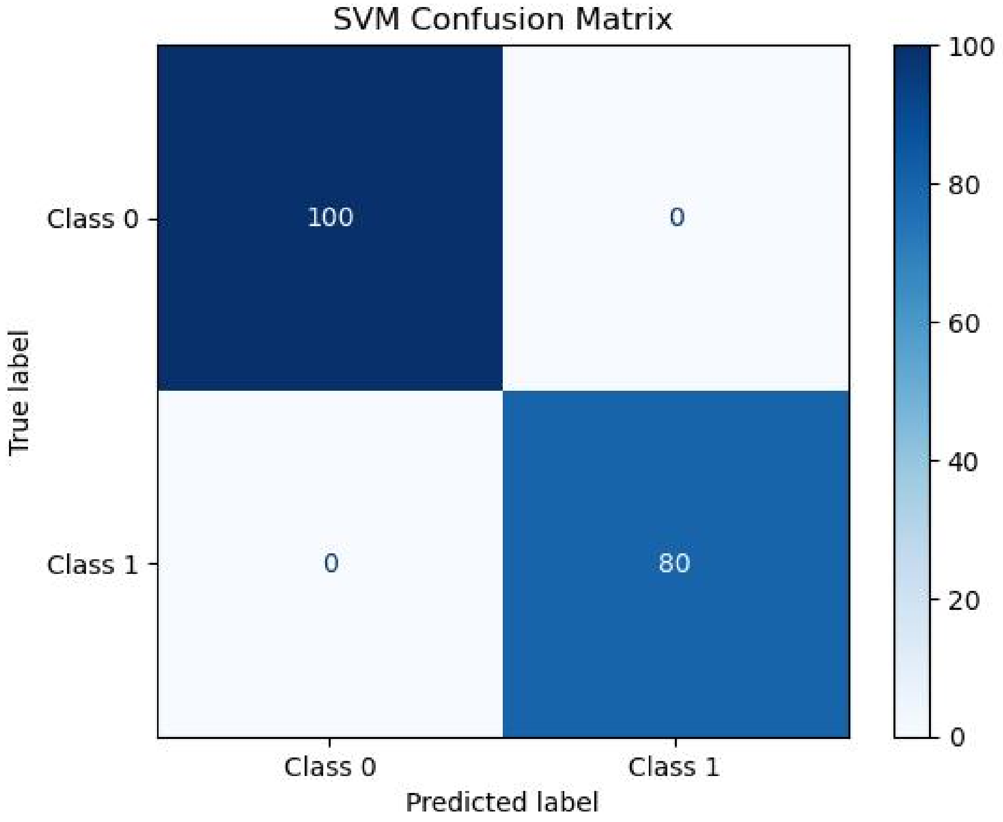

Figure 5.2 Confusion Matrix of SVM

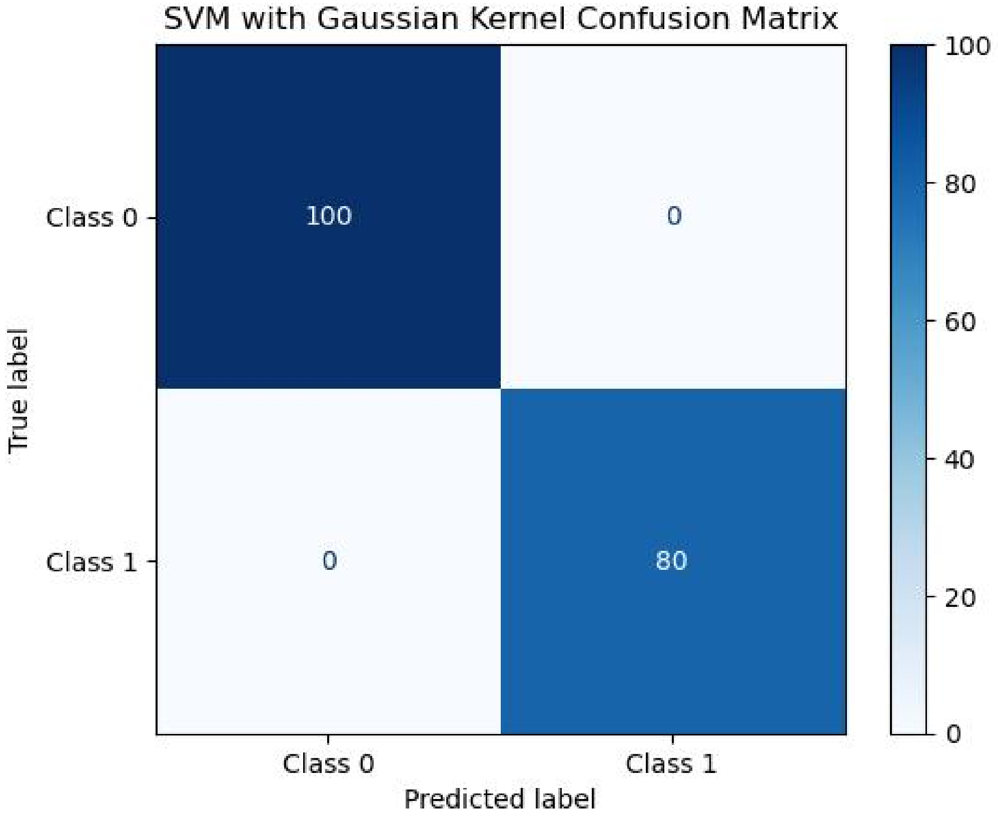

Figure 5.3 Confusion Matrix of SVM(Gaussian Kernel)

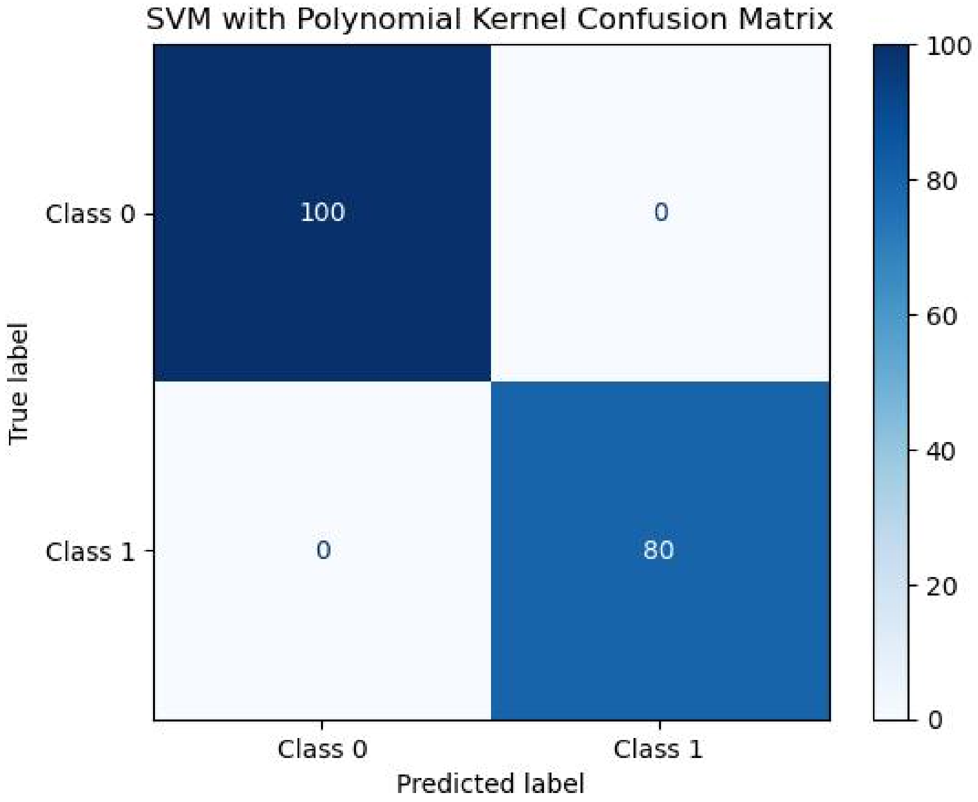

Figure 5.4 Confusion Matrix of SVM(Polynomial Kernel)

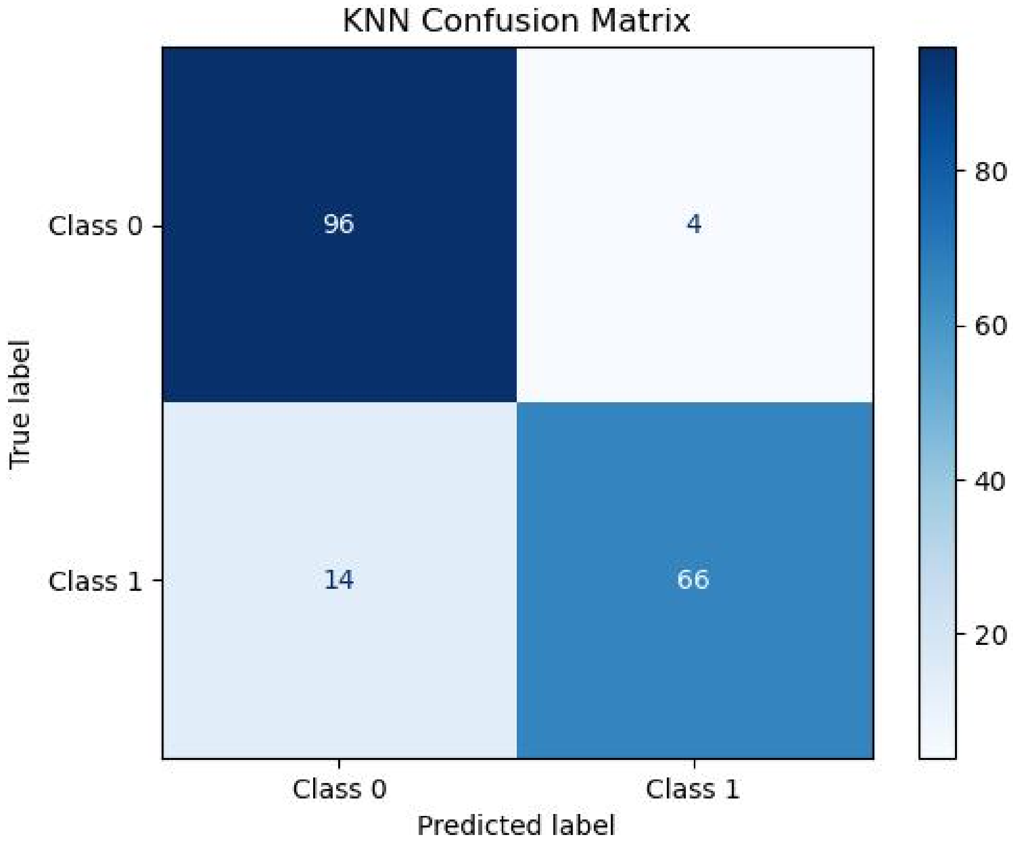

Figure 5.5 Confusion Matrix of KNN

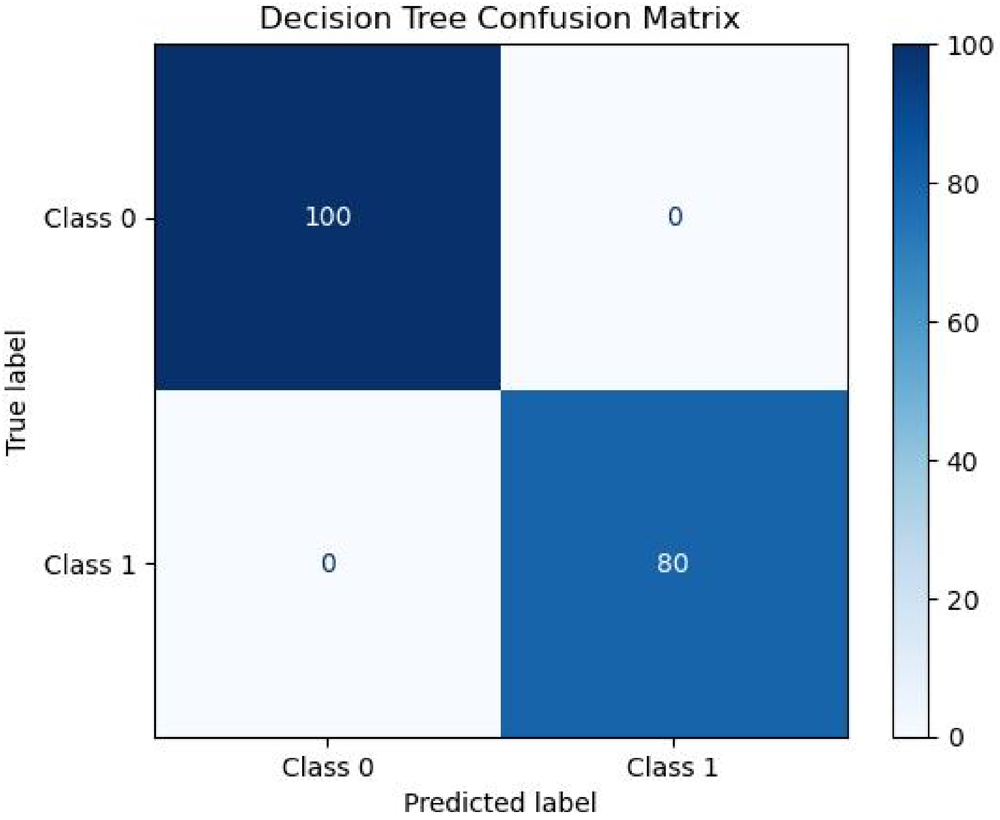

Figure 5.6 Confusion Matrix of Random Forest

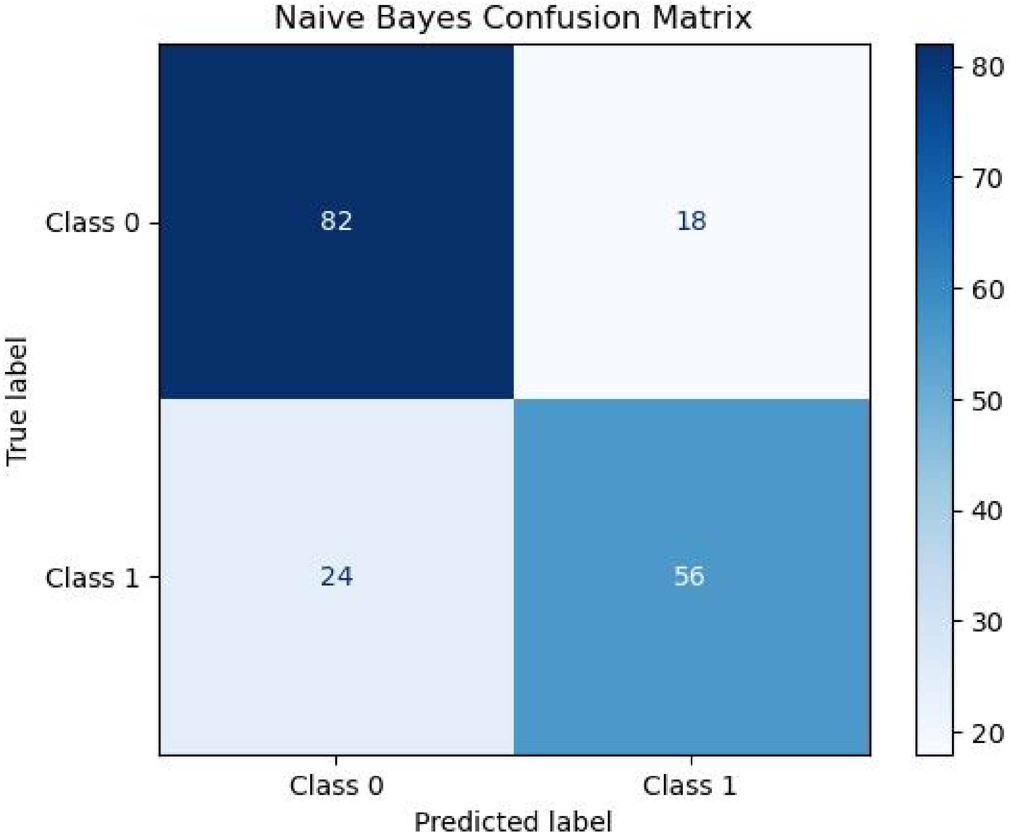

Figure 5.7 Confusion Matrix of Naive Bayes

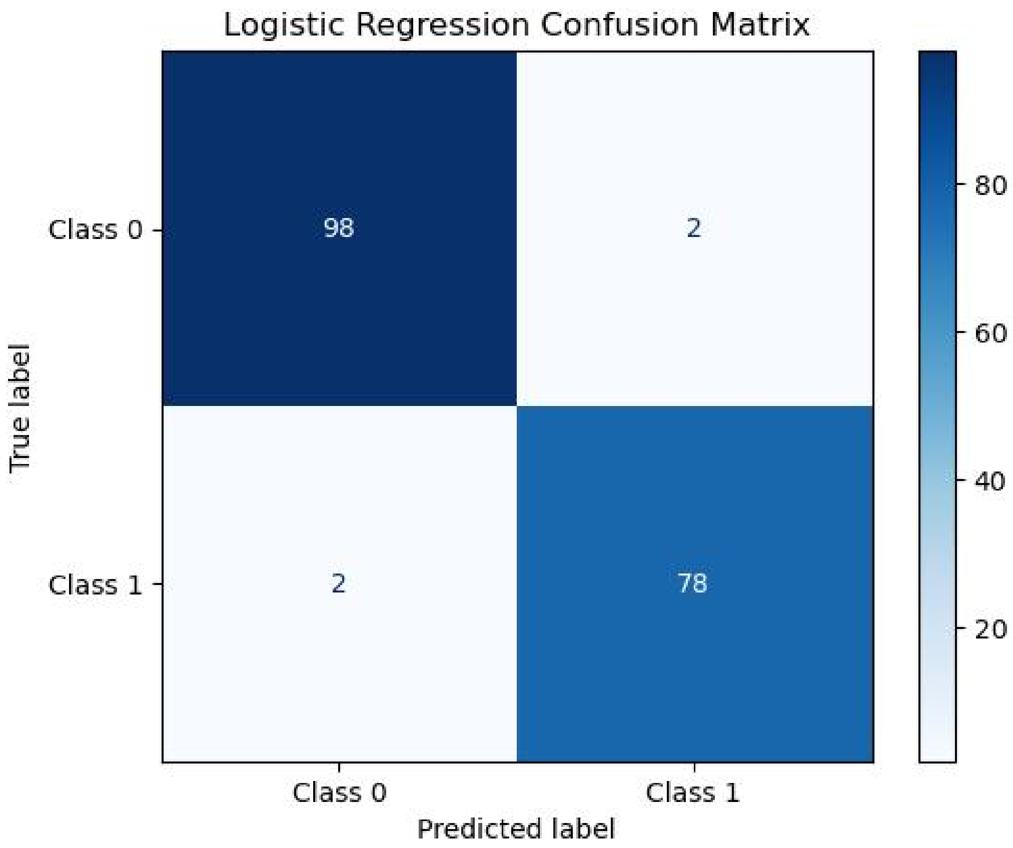

Figure 5.8 Confusion Matrix of Logistic Regression

5.3 Average Accuracy

The average accuracy of the model is shown in the table below:

Classifier	Average Accuracy
SVM(RBF)	0.9500000000000001
SVM(Poly)	0.9500000000000001
SVM(Gaussian)	0.9111111111111112
KNN	0.8611111111111112
Random Forest	0.8444444444444446
Naive Bayes	0.7555555555555556
Logistic Regression	0.7555555555555556

Table 5.9 Average Accuracy of Different Classifiers

5.4 Baseline

In the subsequent analysis, a baseline comparison was conducted using a classifier model. The test data was used for label prediction, and the baseline accuracy was calculated as a metric for evaluating the model's performance. The results are as follows:

Classifier	Baseline Accuracy
SVM(RBF)	0.8611111111111112
SVM(Poly)	0.6666666666666666
SVM(Gaussian)	0.8611111111111112
KNN	0.8888888888888888
Random Forest	0.7777777777777778
Naive Bayes	0.6944444444444444
Logistic Regression	0.9166666666666666

Table 5.10 Baseline Accuracy of Different Classifiers

5.5 Others

Other evaluation methods, such as Receiver Operating Characteristic (ROC) curve, provide a graphical representation for assessing the performance of binary classification models[16]. By plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) at different classification thresholds, the ROC curve illustrates the trade-off between sensitivity and specificity.

Typically, a completely random prediction would result in a curve close to the diagonal line (y=x). The closer the curve is to the top-left corner, the better the model's performance, and vice versa. The area under the ROC curve (AUC-ROC) is commonly used as a measure of the model's overall predictive accuracy.

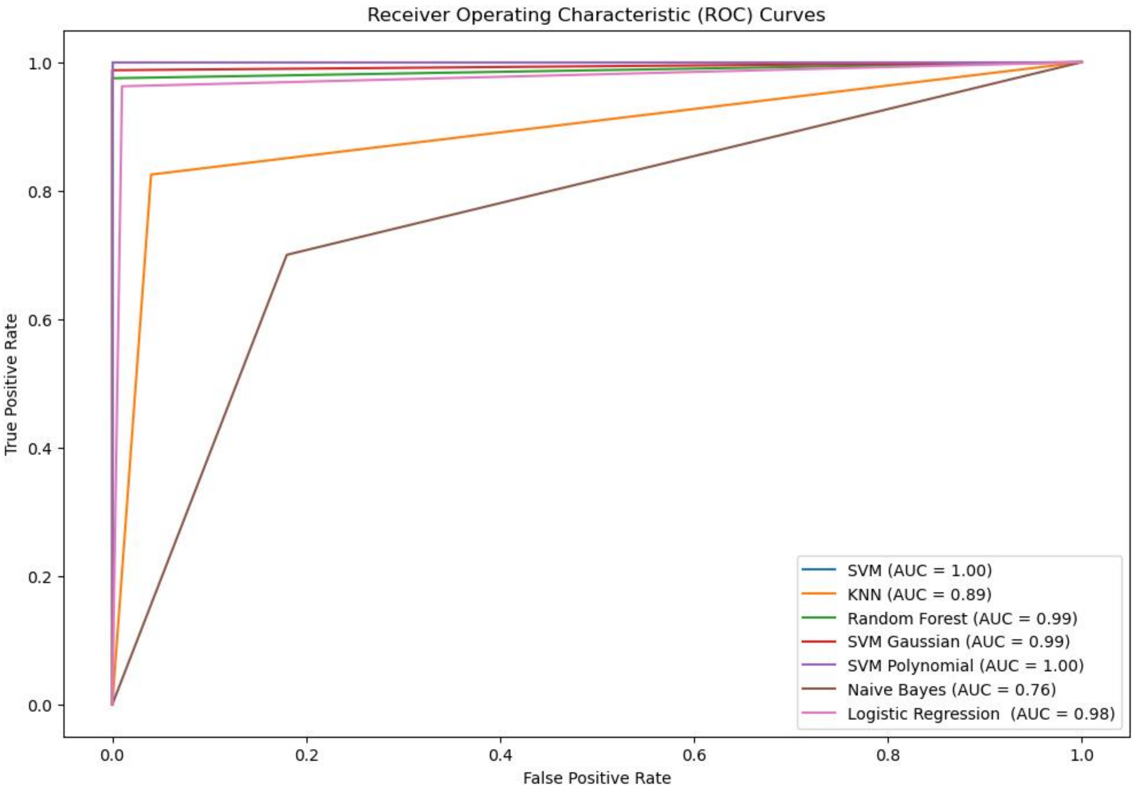

Figure 5.11 ROC Curves of Different Classifiers

Additionally, differences can be assessed using alternative metrics such as MSE (Mean Squared Error), RMSE (Root Mean Squared Error), and MAE (Mean Absolute Error), as depicted in the figure below[17].

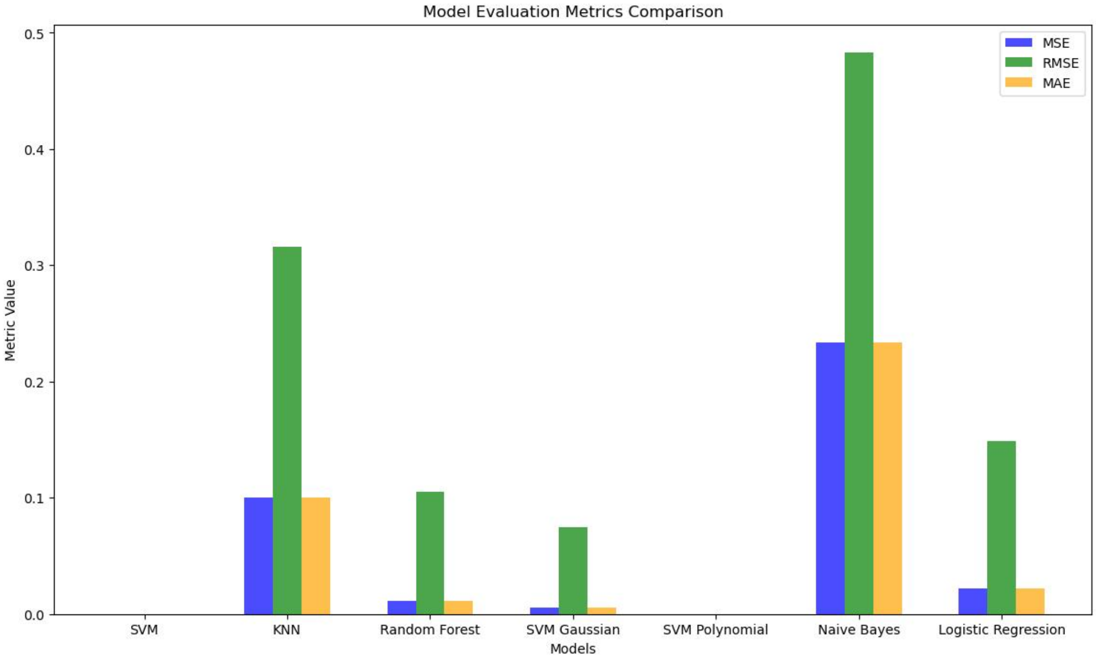

Figure 5.12 Error Bar Chart of Different Classifiers

MSE measures the average squared difference between the predicted and actual values. It provides a measure of the overall magnitude of errors, where higher values indicate larger errors.

RMSE is the square root of MSE and provides a more interpretable metric in the original units of the target variable. It represents the average magnitude of the prediction error and is useful for comparing models or understanding the typical size of errors.

MAE measures the average absolute difference between the predicted and actual values. It provides a more robust measure that is not affected by outliers.

6 Disscussion

Based on the analysis of the results of the models mentioned above, overall, Logistic Regression, KNN, and SVM with linear kernel perform better on average compared to the other three models, which aligns with the general judgment pattern for binary analysis problems.

Nonlinear models such as Gaussian models are inferior to other SVM kernels, indicating the impact of different kernel functions on SVM performance. Naive Bayes ranks last in all indicators, indicating its suboptimal performance in linear problem analysis, maybe GA-evolved Bayesian would be better[18]. Similarly, the Random Forest model did not achieve significant results and, according to our runtime testing, it had the longest training time among all models, second only to SVM and KNN. Given the inadequacy in addressing optimization problems, the answers may vary under a sufficiently long scenario[19].

SVM performed well in this validation, with showing outstanding performance in average prediction accuracy and a low error rate advantage. It did not lag behind other models in any of the evaluation metrics. On the other hand, KNN performed well in the baseline accuracy. Besides, Logistic Regression also showed notable performance in the baseline accuracy and was the fastest model in terms of training speed. Although its average accuracy is not high, it closely approached SVM in terms of ROC and confusion matrix metrics.

In conclusion, the SVM model performed the best, while KNN and Logistic Regression each had their advantages in different aspects.

Reference

[1] Yue Cao, Yuying Chen, Michael DeVivo; Lifetime Direct Costs After Spinal Cord Injury. Top Spinal Cord Inj Rehabil 1 April 2011; 16 (4): 10–16.

[2] Kirschstein T, Köhling R. What is the Source of the EEG? Clinical EEG and Neuroscience. 2009;40(3):146-149. doi:10.1177/155005940904000305

[3] Nunez P.L. ,Srinivasan R. Electric Fields of the Brain: The Neurophysics of EEG., Oxford University Press, 2006

[4] A. Stevens, I. Miralles and B. v. Wesemael, "Soil Organic Carbon Predictions by Airborne Imaging Spectroscopy: Comparing Cross-Validation and Validation," Soil Science Society of America Journal, vol. 76, (6), pp. 2174-2183, 2012.

[5] Z. M. Alhakeem et al, "Prediction of Ecofriendly Concrete Compressive Strength Using Gradient Boosting Regression Tree Combined with GridSearchCV Hyperparameter-Optimization Techniques," Materials, vol. 15, (21), pp. 7432, 2022.

[6] Jović, A., Brkić, K., & Bogunović, N. (2015, May). A review of feature selection methods with applications. In 2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO) (pp. 1200-1205). Ieee.

[7] Tang, J., Alelyani, S. and Liu, H., 2014. Feature selection for classification: A review. Data classification: Algorithms and applications, p.37.

[8] Chandrashekar, Girish, and Ferat Sahin. "A survey on feature selection methods." Computers & Electrical Engineering 40.1 (2014): 16-28.

[9] Kukreja, Sunil L., Johan Löfberg, and Martin J. Brenner. "A least absolute shrinkage and selection operator (LASSO) for nonlinear system identification." IFAC proceedings volumes 39.1 (2006): 814-819.

[10] Ö. Karal, "Performance comparison of different kernel functions in SVM for different k value in k-fold cross-validation," 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Istanbul, Turkey, 2020, pp. 1-5

[11] Skrzeszewski, L. (2018). Hyper-parameter tuning with Pipelines. Medium. https://medium.com/@kocur4d/hyper-parameter-tuning-with-pipelines-5310aff069d6

[12] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011. https://scikit-learn.org/stable/about.html#citing-scikit-learn

[13] Verma, A. (2023). GridSearchCV in scikit-learn: A Comprehensive Guide. DEV Community.https://dev.to/anurag629/gridsearchcv-in-scikit-learn-a-comprehensive-guide-2a72

[14] N. W. S. Wardhani, M. Y. Rochayani, A. Iriany, A. D. Sulistyono and P. Lestantyo, "Cross-validation Metrics for Evaluating Classification Performance on Imbalanced Data," 2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA), Tangerang, Indonesia, 2019, pp. 14-18

[15] B. P. Salmon, W. Kleynhans, C. P. Schwegmann and J. C. Olivier, "Proper comparison among methods using a confusion matrix," 2015 IEEE International

Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 2015, pp. 3057-3060

[16] Fawcett T. An introduction to ROC analysis[J]. Pattern recognition letters, 2006, 27(8): 861-874.

[17] Chai T, Draxler R R. Root mean square error (RMSE) or mean absolute error (MAE)[J]. Geoscientific model development discussions, 2014, 7(1): 1525-1534.

[18] Wiggins, M., Saad, A., Litt, B., & Vachtsevanos, G. (2008). Evolving a Bayesian classifier for ECG-based age classification in medical applications. Applied soft computing, 8(1), 599-608.

[19] Luz, E. J. da S., Nunes, T. M., de Albuquerque, V. H. C., Papa, J. P., & Menotti, D. (2013). ECG arrhythmia classification based on optimum-path forest. Expert Systems with Applications, 40(9), 3561–3573.