Evaluating Classification Algorithms

HYEONJUN JUN
Department of Mathematics
University of California San Diego
La Jolla, CA 92037
hjun@ucsd.edu

December 11, 2024

Abstract

In this experiment, the following three classification algorithms are evaluated: Support Vector Machines (SVM), Decision Trees, and k-Nearest Neighbors (KNN) using three different datasets from University of California Irvine: Adult Income, Credit Approval, and Bank Marketing. These data sets are represented in terms of data type, class imbalance, and varying data sizes. In this experiment, explore the impact of different train test cases (20-80, 50-50, 80-20). The results will provide us with the strengths and limitations of each algorithm and will give us an improved idea of the data analysis task.

1 Introduction

Machine learning plays an important role in the modern technological environment, and its application scope is expanding in various fields. The successful implementation of these machine learning models is possible through accurate performance evaluation.

Evaluating the performance of a model is important for predicting how well the model will perform in a real environment. Therefore, it is essential to apply an accurate and reliable evaluation method during the model development process.

In this experiment, we will introduce various methods for the depth analysis of Support Vector Machine(SVM), Decision Trees, and K-nearest neighbor(KNN), evaluating the performance of machine learning models and explore the advantages and disadvantages of each method. Through this, you can develop a better model and understand how to effectively apply it to solving real problems.

Performance evaluation of machine learning models is an important process for increasing the reliability of the model and developing a model that can be applied to solving real problems.

2 Method

The following steps outline the approach:There are several metrics used to evaluate the performance of machine learning models. The most basic metrics are Accuracy, Precision and Recall Score. In this experiment, we will focus on Accuracy. Accuracy represents the proportion of samples that the model correctly predicted, and evaluates how accurately the model predicted among the entire data. However, if the data is imbalanced, it is difficult to accurately evaluate the performance of the model with Accuracy alone. Therefore, Area Under the Curve (AUC) will also be used for evaluation. and the experimental setup uses three distinct datasets and evaluates three machine learning classifiers (SVM, Decision Trees, and KNN) across different train-test partitions.

2.1 Data Sets

Adult: This dataset from the UCI Machine Learning Repository containing demographic, work, and income information of individuals. The dataset is loaded using Pandas and cleaned by dropping missing values. The features include age, education, marital status, occupation, and more, while the target variable is whether the income is above or below a certain threshold. Categorical variables are encoded using Label Encoding, and all features are standardized using Standard Scaler.

Credit Approval: This dataset from the UCI repository containing financial attributes related to credit approval decisions. Similar preprocessing steps were applied: missing values were dropped, features were selected, and target encoding was used for categorical variables. The features include credit-related attributes like age, income, debt, and more. Standard scaling was applied to the features.

Bank Marketing: This dataset from the UCI repository focused on marketing campaigns, where the target is whether an individual subscribed to a term deposit. The preprocessing steps included dropping missing values and encoding categorical features. Standard scaling was applied to numerical features to ensure uniformity across the datasets.

2.2 Classifiers

Three classifiers were selected for evaluation:

Support Vector Machine(SVM) is particularly powerful in high-dimensional data and can also solve nonlinear classification problems using kernel tricks. As an advantage, it provides high classification accuracy compared to other algorithms due to the margin optimization principle. However, the performance of the model is greatly affected by the parameter settings.

Decision trees are tree-based model used to classify data or perform regression analysis. As an advantage, unlike linear models, they are effective for nonlinear data. However, they lack stability because the model can change significantly even with small changes in data, K-Nearest Neighbors (KNN) is an algorithm that is not greatly affected by noise in the learning data. Therefore, if the number of data is large, it can show quite effective performance. However, it has the disadvantage that the number of optimal neighbors and which distance calculation measure is suitable for analysis are unclear, so the user must arbitrarily select them according to the characteristics of each data, In addition, the calculation time is long.

3 Experiment

3.1 Experimental Setup

To systematically compare the performance of the classifiers, the following methodology was used:

Cross-Validation: Each dataset was divided into three stratified. This technique ensures that the class distribution in each fold is representative of the entire dataset, preventing any bias during model evaluation. Each fold consisted of training and testing sets with a specific proportion of the original dataset:

20-80 Split: 20 of the data was used for training, and 80 for testing. This configuration was used to test the models with smaller training datasets to simulate scenarios where model training data is limited.

50-50 Split: 50 of the data was allocated for training, and 50 for testing. This split was used to evaluate the performance of the models when training and testing datasets were of equal size, allowing for a balanced comparison.

80-20 Split: 80 of the data was used for training, and 20 for testing. This configuration simulated a scenario where the model has ample data for training and less data for testing, assessing the models' ability to generalize from large training sets.

Each dataset was partitioned in a similar manner to ensure consistency across different datasets and to minimize the influence of dataset-specific characteristics.

3.2 Performance Metrics

Used two performance data Accuracy and Area Under the Curve. Accuracy: The proportion of correctly classified instances out of the total instances in the test set. It provides a straightforward measure of how well the model performed across all test cases. AUC (Area Under the Curve): This metric was used to evaluate the model's ability to distinguish between classes, especially in imbalanced datasets. A higher AUC indicates better model performance in classifying instances and distinguishing between classes.

3.3 Results

The results analysis revealed that SVM consistently outperformed Decision Trees and KNN across datasets in terms of accuracy and AUC, particularly in scenarios with abundant training data (80-20 split), from the data table (Classifier Performance Across Datasets).

Table 1: Classifier Performance Across Datasets

Dataset	**Classifier**	**Train Size**	**Accuracy**	**AUC**
Adult	SVM	20%	0.8393	0.8799
	SVM	50%	0.8451	0.8886
	SVM	80%	0.8471	0.8906
	Decision Tree	20%	0.8068	0.7378
	Decision Tree	50%	0.8115	0.7464
	Decision Tree	80%	0.8076	0.7414
	KNN	20%	0.8147	0.8311
	KNN	50%	0.8234	0.8423
	KNN	80%	0.8272	0.8473
Credit Approval	SVM	20%	0.8484	0.9112
	SVM	50%	0.8591	0.9242
	SVM	80%	0.8576	0.9207
	Decision Tree	20%	0.8162	0.8112
	Decision Tree	50%	0.8147	0.8121
	Decision Tree	80%	0.8224	0.8197
	KNN	20%	0.7932	0.8569
	KNN	50%	0.8346	0.8891
	KNN	80%	0.8498	0.9048
Bank Marketing	SVM	20%	0.8964	0.8547
	SVM	50%	0.8971	0.8514
	SVM	80%	0.8981	0.8504
	Decision Tree	20%	0.8657	0.6839
	Decision Tree	50%	0.8697	0.6930
	Decision Tree	80%	0.8733	0.7015
	KNN	20%	0.8888	0.7841
	KNN	50%	0.8903	0.8007
	KNN	80%	0.8915	0.8077

This allows SVM to find an optimal separating hyperplane even when the data is complex, making it particularly effective when there is enough training data available (80-20 split). However, SVM's performance is heavily influenced by the choice of hyperparameters, such as the kernel type and regularization parameter, which require careful tuning to avoid overfitting and ensure robust generalization.

Decision Trees, on the other hand, showed moderate performance in the 50-50 and 80-20 splits but struggled with overfitting, especially with smaller datasets (20-80 split). This is due to the inherent complexity of Decision Trees, which can easily adapt to noise and minor fluctuations in data. This sensitivity leads to overfitting when the training set is small, where the tree might memorize specific instances rather than capturing general patterns. The results highlight the need for strategies like pruning or limiting the depth of the tree to mitigate this issue, making Decision Trees less effective for small training sets.

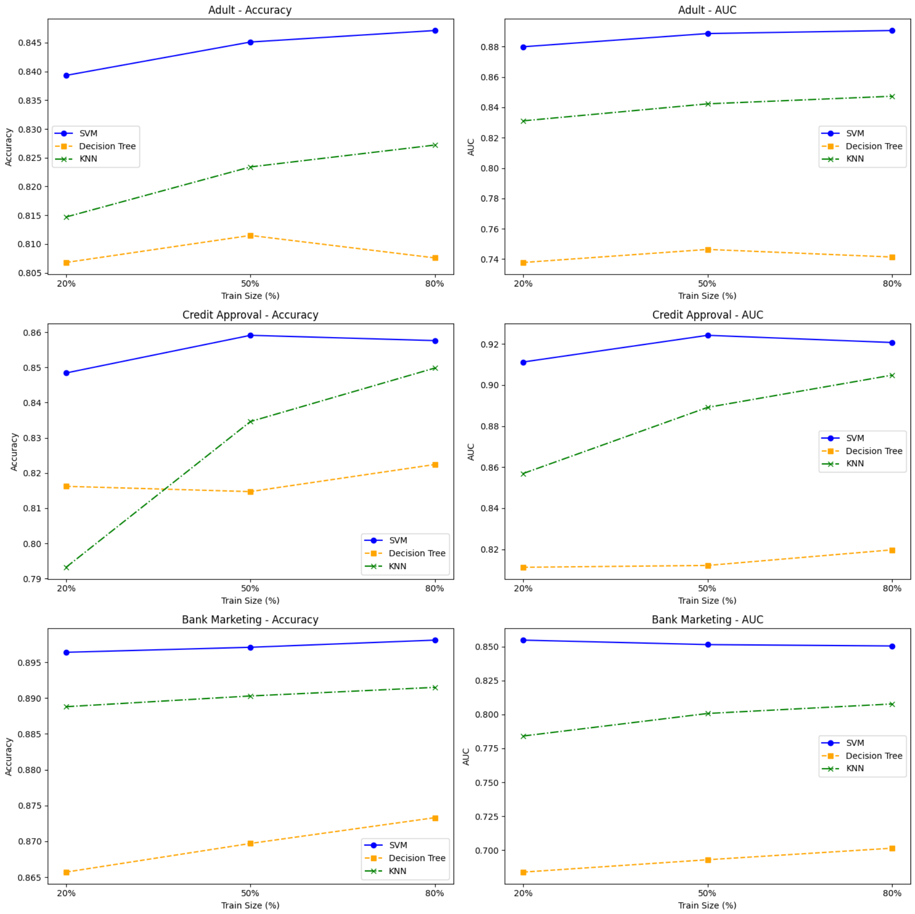

Figure 1: dataset results

KNN performed well with a specific choice of k but was sensitive to training set size and noise, particularly when data was limited. When the training set was small (20-80 split), KNN struggled to generalize well because it relies heavily on local density and distance measures to classify data points. The choice of k significantly impacts performance; if k is too small, the algorithm can be influenced by noise, while a large k may oversmooth the decision boundary. This sensitivity underscores the need for careful selection of k and the importance of having sufficient training data to ensure effective classification.

Overall, SVM was the preferred algorithm for handling complex, high-dimensional data with consistent performance across varying training sizes. Its ability to manage multiple dimensions and nonlinear relationships makes it particularly effective when there is ample data available. Decision Trees and KNN, however, require more meticulous parameter tuning and considerations, particularly with smaller datasets or when dealing with noisy data. These algorithms' limitations highlight the importance of understanding each method's strengths and weaknesses to select the most appropriate one for specific classification tasks.

4 Conclusion

Through this experiment, Importance of Selecting Appropriate Classification Algorithms: The success of classification tasks depends heavily on selecting the right algorithm. Each algorithm, such as Support Vector Machines (SVM), Decision Trees, and K-Nearest Neighbors (KNN), offers different strengths and limitations. Understanding these differences allows for better algorithm selection based on the specific characteristics and requirements of the data.

Utilizing Various Algorithms: It is crucial not to rely solely on one algorithm. Instead, using a combination of SVM, Decision Trees, and KNN can provide a more comprehensive approach to solving classification problems. By leveraging the strengths of each algorithm—SVM's power in handling high-dimensional and nonlinear data, Decision Trees' effectiveness in capturing complex patterns, and KNN's robustness in dealing with noise—one can achieve better results across diverse datasets.

The Importance of Continuous Learning: The machine learning field is constantly evolving, with new algorithms and techniques emerging regularly. To stay effective in solving real-world classification problems, it is essential to keep up with these advancements. Continuous learning and adaptation allow practitioners to apply the latest algorithms and methodologies to build more accurate and reliable models.

References

1. Becker, B., & Kohavi, R. (1996). Adult [Dataset]. UCI Machine Learning Repository. `https://doi.org/10.24432/C5XW20`.

2. Quinlan, J. (1987). Credit Approval [Dataset]. UCI Machine Learning Repository. `https://doi.org/10.24432/C5FS30`.

3. Moro, S., Rita, P., & Cortez, P. (2014). Bank Marketing [Dataset]. UCI Machine Learning Repository. `https://doi.org/10.24432/C5K306`.

4. IBM. (2024, December 11). *k-Nearest Neighbors (KNN)*. *IBM*. Retrieved from `https://www.ibm.com/topics/knn`

5. Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and Regression Trees.* CRC Press.