Classification Models Analysis

Meiyan Liu
UC San Diego
mel048@ucsd.edu

Abstract

This project explores the application and performance of three supervised learning algorithms, Support Vector Machines (SVM), Logistic Regression, and Decision Trees on the three different dataset. Data selection and preprocessing were conducted prior to modeling. Rigorous evaluation methods, including cross-validation and accuracy measurements, are used to ensure model comparison. The results provide insights into how different classifiers, hyperparameters, and data partitions influence classification performance.

Introduction

This project focuses on the empirical comparison of multiple classification algorithms applied to selected datasets. By carefully selecting a pair of class labels and ensuring a consistent two-class classification setting, the project aims to replicate and evaluate trends in classifier performance under varying conditions. Three classifiers—such as SVMs, Logistic Regression, and Decision Trees—are trained and tested on several datasets using systematic hyperparameter optimization and cross-validation, closely mirroring the experimental result described in the referenced paper. Through detailed analysis of validation and test accuracies, heatmap visualizations, and best-parameter selection. The project seeks to provide clear insights into the relative strengths, weaknesses, and general behaviors of these algorithms.

1 Method

1.1 Dataset selecting and Preprocessing

Each dataset was carefully examined to extract a binary classification task. Then preprocessing steps were applied across all selected datasets, converting any non-numerical (categorical) attributes into numerical 0 or 1.

1.2 Model and Algorithms Implementation

1.2.1 Support Vector Machines (SVMs)

The `SVM Classifier` is implemented using `sklearn`'s `SVC` class, with grid search and cross-validation for hyperparameter optimization.

- **Initialization**: Supports linear and RBF kernels, with parameters C (regularization strength) and γ (kernel coefficient).
- **Training**: Performs grid search with cross-validation over a hyperparameter space, identifying the best configuration based on training and validation accuracy.
- **Prediction**: Uses the trained SVM model to classify test samples.
- **Evaluation**: Provides metrics such as accuracy and confusion matrices. Additionally, heatmaps visualize accuracy across the hyperparameter grid.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

1.2.2 Logistic Regression Classifier

The `Logistic Regression Classifier` provides an efficient implementation of logistic regression with L2 regularization.

- **Initialization**: The model accepts parameters for learning rate (α), number of iterations, and regularization strength (C), initializing weights and bias to zero.
- **Training**: The `fit` method updates weights and bias iteratively using gradient descent. The sigmoid activation function transforms linear model outputs into probabilities, and weights are regularized to prevent overfitting.
- **Prediction**: The `predict` method calculates probabilities and classifies inputs based on a threshold of 0.5.
- **Evaluation**: The `evaluate` method computes the accuracy of predictions, providing an overview of model performance.

1.2.3 Decision Tree Classifier

The `Decision Tree Classifier` implements a custom decision tree algorithm, supporting features like depth control and minimum samples for splits.

- **Initialization**: The model allows customization of maximum tree depth and minimum samples required for splitting nodes.
- **Training**: The `fit` method builds the tree recursively, identifying the best splits using Gini impurity. The method handles stopping criteria like depth limits and class purity.
- **Prediction**: The `predict` method traverses the tree for each test sample, assigning class labels based on leaf nodes.
- **Evaluation**: The `evaluate` method calculates prediction accuracy and compares it to the ground truth.

2 Results and Analysis

2.1 Performance Across Training Splits

The models were evaluated on three datasets with varying training split ratios (80%, 50%, 20%). The following summarizes their performance:

2.1.1 Wine Quality Dataset

- **SVM**: Achieved a test accuracy of approximately **77.9%** with 20% training data after tuning C=1 and gamma=0.1 using the RBF kernel. While it demonstrated moderate generalization ability under these conditions, its accuracy varied obviously at lower training splits, underscoring the sensitivity of SVM performance to the amount of available data and the careful selection of hyperparameters.
- **Logistic Regression**: Reached a peak accuracy of **72.38%** when trained on 50% of the data, exhibiting strong and stable performance across various splits. Leveraging features such as fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol contributed to its consistent generalization capability.
- **Decision Tree**: Demonstrating greater sensitivity to overfitting, the decision tree achieved a high of **78.05%** accuracy when trained on 80% of the data, but this performance declined to **72.34%** as the training portion decreased to 20%, illustrating the reduced training data is overfitting tendencies and lowers overall accuracy.

2.1.2 Obesity Dataset

- **SVM**: These results indicate that while the SVM achieved the highest test accuracy (about **85.89%**) with a 50% training split and default parameters (C=1), increasing C to 10 at a 20%

training split still maintained strong test performance (about **84.72%**). Although validation accuracy is around 0.50 across splits, final test results suggest that well-tuned SVMs can remain relatively robust even with less training data.

- **Logistic Regression**: Across varying training splits (80%, 50%, and 20%), Logistic Regression demonstrated relatively stable performance. With 80% of the data, it achieved its highest test accuracy of approximately **83.54%** by a large regularization parameter (C=100). Reducing the training portion to 50% slightly lowered accuracy to around **81.53%** with a smaller C (0.1), while even at 20% training data, the model maintained a respectable accuracy of about **82.12%** with C=100.
  These results indicate that Logistic Regression provided robust and consistent results, showing only modest fluctuations in accuracy despite substantial variations in training set size and regularization strength.

- **Decision Tree**: With an 80% training split, the decision tree achieved its highest accuracy **92.07%** using an unrestricted depth, indicating the model could fully adapt to the available data. As the training portion decreased to 50%, performance dropped to **84.56%**, and optimal parameters required limiting the tree depth to 10. With just 20% training data, the accuracy decreased further to **81. 94%**, now best achieved with a shallower tree (with max_depth=3).
  These results illustrate shrinking the training set reduces accuracy.

2.1.3 Maternal Health Dataset

- **SVM**: Achieved strong test accuracies ranging from approximately 85% to nearly 88%. With a larger training portion (80%), a linear kernel combined with a low regularization parameter (C=0.1) delivered the highest test accuracy **87.68%**. Reducing the training data to 50% still produced robust results **85.99%** but favored an RBF kernel with C=1 and gamma=1, suggesting that the model needed a more flexible decision boundary. At only 20% training data, low C leads to a solid test accuracy **85.47%**.
  Overall, these outcomes demonstrate that SVM can adapt well to different training data sizes by selecting appropriate kernel and regularization settings, maintaining relatively stable performance across substantially varied training portions.

- **Logistic Regression**: Exhibited stable performance across all training. With 80% training data, it attained an accuracy of about **85.47%** using a high regularization value (C=100) and L2 penalty. Even when reducing the training split to 50%, accuracy slightly increased to around **86.19%** with the same parameter, suggesting that less data did not substantially degrade performance. At only 20% training data, the model still maintained about **85.47%** accuracy.

- **Decision Tree**: The decision tree maintained relatively consistent accuracy levels across different training splits, ranging from about **84.98%** to **85.21%**. With 80% and 50% training data, a maximum depth of 5 and a minimum split size of 2 produced the best results. Even when the training data was reduced to 20%, the model still performed well (about **84.98%**), though a shallower tree (max_depth=3) was preferred.
  These findings suggest the decision tree can adapt effectively to changes in training set size, requiring only modest adjustments to its depth to maintain stable performance.

3 Discussion

3.1 Model Strengths and Weaknesses

3.1.1 SVM

- **Strengths**: Excels in capturing both linear and non-linear patterns, with robust performance through hyperparameter optimization. The SVM demonstrated strong generalization capability, maintaining relatively high test accuracies even as the amount of training data was reduced. With careful hyperparameter selection—particularly choosing appropriate kernel types, C values, and gamma—it consistently achieved robust performance. This indicates that SVMs can adapt well to different conditions and data sizes, making them a reliable option in scenarios where data availability may be limited or diverse.

- **Weaknesses**: A key challenge with SVMs lies in their sensitivity to hyperparameter choices, which requiring careful tuning of C and γ for large datasets. Additionally without proper parameter tuning, SVMs may show suboptimal validation accuracy, and their decision boundaries may be less interpretable compared to simpler models like logistic regression or decision trees.

3.1.2 Logistic Regression

- **Strengths**: Logistic Regression showed stable and consistent performance across different training split sizes, often maintaining comparable accuracy even with substantially less training data. It also demonstrated relatively straightforward hyperparameter tuning, with a single regularization parameter (C) and L2 penalty to achieve strong results. This stability and simplicity make it a reliable baseline method, and its linear nature ensures more interpretable decision boundaries compared to more complex models. Also Linear regression is robust to high-dimensional data and effective in datasets with linear separability.
- **Weaknesses**: Although Logistic Regression performed consistently, it did not always achieve the highest possible accuracy. Its linear decision boundary may limit its ability to capture more complex patterns in data that non-linear models can exploit. Additionally, while it is less sensitive to hyperparameters than models like SVM, the fixed form of its decision boundary may prevent it from excelling when the underlying data distribution. Therefore, Logistic Regression struggles with non-linear decision boundaries and datasets with complex feature interactions.

3.1.3 Decision Tree

- **Strengths**: Decision trees excel at uncovering complex, non-linear relationships between features, and they are relatively straightforward to interpret. Each split within a tree offers an intuitive explanation for the prediction. They are also flexible, adapting to various data sizes and structures.
- **Weaknesses**: Decision trees can be highly susceptible to overfitting if allowed to grow without constraints, capturing specific patterns that do not generalize well. They are also sensitive to small changes in the training data, which can lead to unstable results and drastically different tree structures. As a result, decision trees may not always achieve the highest accuracy compared to models with stronger regularization.

4 Conclusion

The study highlights the strengths and limitations of Support Vector Machines, Logistic Regression and Decision Tree models across different datasets and scenarios. SVMs performed robustly when properly tuned, showcasing adaptability across conditions, but required careful parameter selection to reach their potential. Decision Trees excel in capturing non-linear relationships, while Logistic Regression proves robust for linearly separable datasets. Ultimately, no single algorithm outperformed the others in all scenarios. These findings emphasize the importance of model selection based on dataset characteristics.

References

1. Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408, 189–215. `https://doi.org/10.1016/j.neucom.2019.10.118`
2. Alzen, J. L., Langdon, L. S., Otero, V. K. (2018). A logistic regression investigation of the relationship between the Learning Assistant model and failure rates in introductory STEM courses.*International Journal of STEM Education, 5,* , Article 56. `https://doi.org/10.1186/s40594-018-0152-1`.
3. Quinlan, J. R. (1987). Simplifying decision trees. *International Journal of Man-Machine Studies*, 27(3), 221–234. `https://doi.org/10.1016/S0020-7373(87)80053-6`.

4. Salazar, D. A., Vélez, J. I., Salazar, J. C. (2012). Comparison between SVM and logistic regression: Which one is better to discriminate? *Revista Colombiana de Estadística*, 35(2), 223–237.