Anand Mysorekar

Cogs 118a FA24 Final Project

Abstract

This project explores how three popular classification models—Random Forest, Support Vector Machine (SVM), and Logistic Regression—perform on binary classification tasks using datasets from the UCI Machine Learning Repository. Each dataset was set up for binary classification, and the models were tested under different train/test splits (20/80, 50/50, and 80/20) with hyperparameter tuning to optimize their performance. To evaluate the models, I compared their validation, and test accuracies across the splits. The results showed that Random Forest consistently came out on top, delivering the highest accuracy across all the datasets. Meanwhile, SVM and Logistic Regression showed their own strengths, depending on how complex the data was and how the features were distributed. Overall, this project highlights how choosing the right model and carefully tuning its hyperparameters can make a big difference, especially when dealing with different types of datasets and training/testing ratios.

Introduction

Classification is a core task in machine learning, where the goal is to sort inputs into specific categories. Binary classification, in particular, has tons of real-world applications, like spotting spam emails, diagnosing medical conditions, or predicting whether a customer will leave a service. Choosing the right classification model is key to achieving good accuracy and making sure it works well across different datasets.

How well a classifier performs often depends on the characteristics of the dataset—things like the number of features, how much training data is available, and how the data is distributed. Testing multiple classifiers on the same datasets can help uncover their strengths, weaknesses, and how well they generalize.

In this project, I'm comparing the performance of three popular classification models on three datasets. By testing them with different train/test splits and tweaking their hyperparameters, I aim to understand their behavior and identify trends and best practices for choosing the right model.

Methods

I am going to use the following models for each dataset: Support Vector Machine, Random Forest, and Logistic Regression. For each of the models I will:

- Preprocess the data

    - Scale the data using StandardScaler to standardize the input values for better model performance
    - Separate the data into features and target column
- Partition the data into training and testing sets
    - Use 20/80, 50/50, and 80/20 splits to evaluate model performance under different training/testing ratios
- Perform hyperparameter tuning using GridSearchCV
    - Tune the hyperparameters for each model to optimize performance
- Train the model
    - Fit the model on the training data for each partition using the best hyperparameters
- Evaluate the model
    - Evaluate the model using cross scores to compute the average validation accuracy
    - Evaluate the model on the test set to compute the test accuracy
    - Generate classification reports to analyze precision, recall, and F1 scores

Results

SVM Analysis

Dataset	Best Partition (test/train)	Validation Accuracy	Test Accuracy
Adult Dataset	20/80	0.84	0.85
Heart Disease Dataset	50/50	0.80	0.85
Red Wine Dataset	20/80	0.88	0.92
White Wine Dataset	20/80	0.84	0.85
Combined Wine Dataset	20/80	0.84	0.86

Random Forest Analysis

Dataset	Best Partition (test/train)	Validation Accuracy	Test Accuracy
Adult Dataset	20/80	0.86	0.86
Heart Disease Dataset	20/80	0.83	0.88
Red Wine Dataset	20/80	0.91	0.93
White Wine Dataset	20/80	0.88	0.89
Combined Wine Dataset	20/80	0.88	0.91

Logistic Regression Analysis

Dataset	Best Partition (test/train)	Validation Accuracy	Test Accuracy
Adult Dataset	50/50	0.83	0.84

Dataset	Best Partition (test/train)	Validation Accuracy	Test Accuracy
Heart Disease Dataset	20/80	0.84	0.87
Red Wine Dataset	20/80	0.88	0.90
White Wine Dataset	20/80	0.80	0.81
Combined Wine Dataset	20/80	0.82	0.83

Model Comparison

Dataset	Model	Best Partition (test/train)	Validation Accuracy	Test Accuracy
Adult Dataset	**Random Forest**	20/80	0.86	0.86
	SVM	20/80	0.84	0.85
	Logistic Regression	50/50	0.83	0.84
Heart Disease Dataset	**Random Forest**	20/80	0.83	0.88
	Logistic Regression	20/80	0.84	0.87
	SVM	50/50	0.80	0.85
Red Wine Dataset	**Random Forest**	20/80	0.91	0.93
	SVM	20/80	0.88	0.92
	Logistic Regression	20/80	0.88	0.90
White Wine Dataset	**Random Forest**	20/80	0.88	0.89
	SVM	20/80	0.84	0.85
	Logistic Regression	20/80	0.80	0.81
Combined Wine Dataset	**Random Forest**	20/80	0.88	0.91
	SVM	20/80	0.84	0.86
	Logistic Regression	20/80	0.82	0.83

Discussion

1. **Random Forest Performance**
   Random Forest stood out as the best performer across all datasets for binary classification, and there are a few reasons why:
   - **Capturing complex patterns**: Random Forest works by combining many decision trees, which makes it great at finding non-linear relationships in data. This flexibility gave it an edge in datasets like **Red Wine** and **Combined Wine**, where

the relationships between features and the target variable are more intricate.

- **Built-in robustness**: By using ensemble learning techniques like bootstrap sampling and random feature selection, Random Forest is naturally resistant to overfitting. This stability translated into consistently strong performance on the test sets.

- **Adaptability to feature types**: Random Forest handles both categorical and numerical data well, which was particularly useful for datasets like **Adult** and **Heart Disease**, where feature types and scales vary a lot.

2. **Best Partition: Why 20/80 Test-Train Split Often Performs Best**
   - **Training data matters**: With 80% of the data used for training, models get more opportunities to learn patterns effectively. Smaller training sets, like in a 50/50 split, can leave models like SVM and Logistic Regression struggling to capture the full complexity of some datasets.

   - **Stable evaluations**: Using 20% of the data for testing offers a good balance—large enough to give reliable test accuracy but not so large that the training set becomes too small.

   - **Random Forest's gain from 20/80**: Random Forest thrives on having more data to train its ensemble of trees, which is why it performed especially well with this split in datasets like **Red Wine** and **Combined Wine**.

3. **Exceptions: 50/50 Partition for Best Performance**
   Interestingly, the **Adult Dataset** for Logistic Regression and **Heart Disease Dataset** for SVM showed better results with a 50/50 split:
   - **Simpler patterns**: Logistic Regression, being linear, didn't need a massive training set to learn the relatively straightforward relationships in the **Adult Dataset**. The 50/50 split likely helped strike a good balance between learning and evaluation.

   - **Noise control**: For SVM, the **Heart Disease Dataset** might contain some noise or redundant features. A smaller training set in the 50/50 split could have helped prevent overfitting, especially when tuning the regularization hyperparameter `C`.

4. **SVM Performance**
   SVM generally outperformed Logistic Regression but didn't surpass Random Forest. Here's why:
   - **Kernel magic**: SVM's RBF kernel allows it to handle non-linear data, which gave it an advantage in datasets like **Red Wine** and **Combined Wine**.

   - **Strong regularization**: By tuning its `C` parameter, SVM can find a good trade-off between keeping the margin wide and minimizing classification errors, which helps it generalize well.

   - **Challenges compared to Random Forest**: SVM is more sensitive to hyperparameter tuning and can struggle with high-dimensional or noisy data.

Random Forest, with its built-in handling of such challenges, was better suited to those scenarios.

5. **Logistic Regression Performance**
   Logistic Regression performed well in specific cases, especially where relationships between features and the target were linear:
   - **Quick and effective**: Logistic Regression did well on the **Heart Disease Dataset**, where its simplicity and ability to converge quickly on scaled features made it a strong contender.
   - **Struggles with complexity**: On datasets like **Red Wine** and **Combined Wine**, where relationships between features are more intricate, Logistic Regression's assumption of linearity limited its performance.
6. **Why SVM Outperformed Logistic Regression in Most Cases**
   - **Handling complexity**: SVM's RBF kernel enabled it to handle datasets with more complex relationships, which gave it an advantage over Logistic Regression in most cases.
   - **Flexibility in tuning**: SVM's hyperparameters (`C` and `gamma`) allowed it to adapt better to the structure of individual datasets, giving it an edge.

Summary of Key Trends

- **Random Forest's Strengths**: Random Forest consistently delivered the highest test accuracy, thanks to its ability to handle non-linear relationships, noisy features, and complex data variability.
- **Partition Impact**: The 20/80 split provided reliable results by giving models enough training data while still leaving a sufficiently large test set. Exceptions, like the 50/50 splits, often came down to dataset simplicity or noise.
- **SVM's Flexibility**: SVM performed strongly on datasets with non-linear relationships, showing competitive results with Random Forest on structured datasets like **Red Wine** and **Combined Wine**.

Data Availability

The datasets selected for this project were sourced from the UCI Machine Learning Repository. While the initial goal was to use larger datasets for more robust analysis, two of the datasets considered had relatively few instances, limiting their suitability for more complex analyses. Additionally, due to constraints in computational resources, smaller datasets were prioritized to ensure that the experiments could be conducted within a reasonable time frame. This choice allowed for efficient hyperparameter tuning and cross-validation while still maintaining the rigor of the experimental design.

Limitations

There were a few limitations in this study that could have impacted the results. One of the main challenges was working with smaller datasets. Small datasets can make it harder for models, especially complex ones like SVM and Random Forest, to generalize well. They might not capture all the important patterns in the data and can be more prone to overfitting. Additionally, smaller datasets can lead to less reliable test accuracy results since random splits might introduce more variability in performance.

Another limitation was the simplicity of the grid search used for hyperparameter tuning. Due to limited computational resources, it wasn't feasible to perform a more comprehensive search or use advanced methods like randomized search or Bayesian optimization. This meant that some models, particularly SVM, might not have been tuned to their full potential, as their performance can be very sensitive to parameters like C and gamma.

Finally, the limited compute power also restricted the scope of the experiments. I had to work with smaller datasets and couldn't run as many trials as I would have liked. With access to more powerful resources, future work could expand on this by using larger datasets, more sophisticated tuning strategies, and testing a wider range of models to potentially achieve better results.

Conclusion

In this project, I explored how three different models—Support Vector Machine (SVM), Random Forest, and Logistic Regression—perform on binary classification tasks using three datasets. The datasets included the Adult Dataset (predicting whether someone earns over $50K), the Heart Disease Dataset (predicting the presence of heart disease), and the Wine Quality Dataset (predicting whether a wine is of good quality). Each dataset was tested with three train/test splits (20/80, 50/50, and 80/20), and I ran three trials for each split to average the results and reduce the effect of random variations.

To get the data ready for modeling, I preprocessed it and set up the architectures for each of the three models. Then, I ran experiments, fine-tuned hyperparameters using grid search, and analyzed how each model performed on the different datasets and splits.

The results showed that Random Forest consistently outperformed the other models, demonstrating its strength in handling diverse datasets and capturing complex relationships. SVM generally came in second, thanks to its flexibility in using kernels to model more complicated patterns. Logistic Regression struggled with datasets requiring more complex decision boundaries, but held its own on datasets with more straightforward, linear relationships.

Overall, the project confirmed the strengths of Random Forest as a top-performing model for binary classification tasks. It also highlighted how understanding the strengths and limitations of each model is crucial for making the right choice based on the characteristics of the data and the task at hand.

References

1. Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery, 2*(2), 121–167. DOI: 10.1023/A:1009715923555

2. Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5–32. DOI: 10.1023/A:1010933404324

3. Lecture Notes: Shalizi, C. R. (2012). Logistic Regression. Carnegie Mellon University. Retrieved from http://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf

4. Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, 161–168. Association for Computing Machinery, New York, NY, USA. DOI: 10.1145/1143844.1143865