Final Project Report

Harry Wang A17326775

December 14, 2024

Abstract

Several supervised learning classification algorithms have been in use for the last few decades. However, for binary classification tasks, determining an ideal model remains a challenge. We present an empirical comparison between 3 prevalent candidate classification methods: boosted trees, random forests, and support vector machines, and evaluate the models' accuracy on 3 distinct datasets adapted for binary classification. Additionally, we discuss the potential trade-offs associated with each method and leverage these insights to propose and evaluate a stacked ensemble model that combines the individual models.

1 Introduction

Classification tasks form a cornerstone of machine learning, offering solutions to applications from spam detection to medical diagnosis. Among these, binary classification stands out for its applicability and straightforward evaluation process. Despite extensive research and applications, selecting the optimal classification algorithm for binary tasks remains a challenge influenced by dataset characteristics and performance demands.

In a Cornell research, algorithms like boosted trees, random forests(Ho, 1995), and support vector machines (Hearst et al., 1998) have demonstrated robust benchmarks across performance metrics (Caruana & Niculescu-Mizil, 2006). For general applications, popular sentiment appears to favor SVM and Random Forest models, while boosted trees (Chen & Guestrin, 2016) are also widely used for their ease-of-use and speed(Gozhulovskyi, 2022). A direction of research with fewer formal results is ensemble deep learning: by combining several different models, it may be possible to leverage the strengths and vulnerabilities in a stacked ensemble. While these models are computationally expensive, theoretical studies have shown that they generally yield higher accuracies than the individual models they are composed of (Mohammed & Kora, 2023).

Our study aims to empirically evaluate and compare the performance of these three popular algorithms on three distinct datasets tailored for binary classification. By doing so, we seek to identify specific conditions under which one algorithm may outperform others and explore the potential of a stacked ensemble approach to incorporate an additional layer of emergence (Mohammed & Kora, 2023). While often infeasible for a simple binary classification task, our results demonstrate an effective proof-of-concept for stacked ensemble classifiers.

2 Methodology

To evaluate our models, we tested them on 3 different datasets sourced from the UC Irvine Machine Learning repository (Markelle Kelly, 2017). The data are from different domains to demonstrate the transferability of the classifier architectures themselves.

The first dataset was the Wine Quality Dataset (Cortez & Reis, 2009). This dataset contains 5000 instances of red and white wine data, with each instance containing quantitative information concerning chemical features such as acidity, sulfur dioxide, alcohol content. Each wine is also scored 1-10 on the integer scale as the target variable. For the binary classification task, exploratory data analysis indicated that the median score in the dataset was between 5 and 6. As a result, we modified the target data to give a score of 1 if the quality was above a 6, and a score of 0 otherwise; essentially splitting the instances into good or bad quality wine.

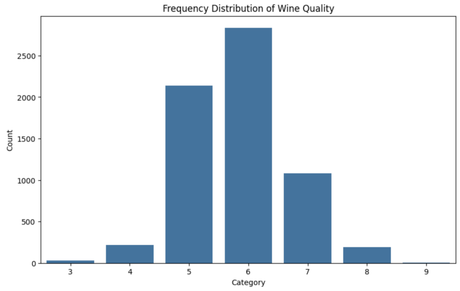

Figure 1: Distribution of wine quality scores

The second dataset was Bank Marketing (Moro & Cortez, 2014). This data concerned direct marketing campaigns of a Portuguese banking institution. The classification variable is whether the client subscribed a term deposit. This dataset required significant amounts of processing, as much of the features were recorded as strings. Since the classical SVM structure is solely for numerical data, categorical features were one-hot encoded.

The final dataset was Default of Credit Card Clients (Yeh, 2009). This data concerned customers' default payments in Taiwan. Similar to the Bank Marketing dataset, instances in the Credit Default data contained demographic information concerning education, marital status, and employment status which were one-hot encoded.

After the data was initially cleaned and processed, we reserved 10% of each dataset to evaluate the stacked ensemble model after all training/testing of its composite classifiers.

To train and evaluate the performance of each classifier model, two scripts were run. The first script concerned the optimal hyperparameters of each model: it utilized the GridSearchCV method from scikit-learn to conduct a systematic search through a predefined range of hyperparameter values. For each classifier—Random Forest, SVM, and XGBoost—a grid search was performed on the training data split from the dataset to identify the combination of parameters that yields the best model performance based on cross-validation results.

Each grid search was configured to explore a range of hyperparameters specific to each model type. For the Random Forest, parameters such as the number of trees and maximum depth were varied; for the SVM, parameters included the penalty parameter C and the choice of kernel; and for the XGBoost, parameters like learning rate, max depth, and number of estimators were considered. Optimal parameteres per model per dataset are discussed in the Experiment (Result) section.

Once the hyperparameters were determined. we sought to replicate the benchmarks from the Caruana paper (Caruana & Niculescu-Mizil, 2006). We conducted an evaluation of the three models across three different training and test set ratios (0.2, 0.5, and 0.8). The evaluation process involved running each model three times for each partition to ensure reliability. The datasets were split accordingly, and each model was trained and tested with its respective best parameters. After training, the performance of each model was assessed by computing the average accuracy over the three runs. We discuss the performance and rank the models further in the Experiment section.

Once the benchmarks of the individual models were established, we proceeded to train a stacked ensemble model on the results, and evaluate it on the 10% of the datasets set aside previously. The stacked ensemble was configured by first retraining the base classifiers. These models were then used to generate a new set of features by predicting probabilities on the test portion of the data. These probability predictions served as input features for the meta-learner, a logistic regression model, which was trained to combine the predictions of the base models. Following the training of the meta-learner, we tested its performance on a separate ensemble dataset, specifically reserved for final validation.

3 Experiment

3.1 Hyperparameters

We first present the optimal hyperparameters used for each model on each dataset. Note that these exact hyperparameters were used to evaluate the models for the benchmark scores presented later.

Table 1: Optimal parameters

Model	Parameters
Wine Dataset	
Random Forest	Max Depth: None, Min Samples Split: 2, N Estimators: 200
SVM	C: 1, Kernel: Linear
XGBoost	Learning Rate: 0.1, Max Depth: 7, N Estimators: 200
Bank Dataset	
Random Forest	Max Depth: None, Min Samples Split: 10, N Estimators: 100
SVM	C: 0.1, Kernel: Linear
XGBoost	Learning Rate: 0.1, Max Depth: 5, N Estimators: 200
Credit Dataset	
Random Forest	Max Depth: 10, Min Samples Split: 10, N Estimators: 200
SVM	C: 0.1, Kernel: RBF
XGBoost	Learning Rate: 0.1, Max Depth: 3, N Estimators: 50

3.2 Classifier Model Benchmarks and Comparison

Now, we present the individual model benchmark accuracy scores after being trained on 0.2, 0.5, and 0.8 of the dataset.

Table 2: Average model accuracy for datasets and splits

Split	Random Forest	SVM	XGBoost
	Wine Dataset		
0.8/0.2	0.8316	0.7547	0.8171
0.5/0.5	0.7988	0.7404	0.7948
0.2/0.8	0.7763	0.7443	0.7621
	Bank Dataset		
0.8/0.2	0.8861	0.8684	0.8893
0.5/0.5	0.8815	0.8665	0.8842
0.2/0.8	0.8777	0.8679	0.8804
	Credit Dataset		
0.8/0.2	0.8242	0.7831	0.8244
0.5/0.5	0.8228	0.7832	0.8248
0.2/0.8	0.8182	0.7798	0.8184

As we can see, the SVM underperformed on the wine and credit dataset, which is expected as these datasets contained fewer categorical values and more real-numbered features. Note that the overall best performance of each model

Table 3: Model rankings for different datasets (80/20 train/test split)

Dataset	Random Forest	SVM	XGBoost
Wine	1	3	2
Bank	2	3	1
Credit	2	3	1

was 0.8861 for RF, 0.8684 for SVM, and 0.8893 for XGBoost. This is in line with the empirical paper benchmarks, which presented 0.872 for the best RF performance, 0.843 for boosting trees, and 0.824 for SVM (Caruana & Niculescu-Mizil, 2006). Our model rankings also generally reflect this, with the XGBoost outperforming RF on the bank and credit data. This is to be expected as XGBoost itself is a newer boosted tree architecture that was not evaluated in the Caruana paper. Overall, the SVM architecture performed the worst, which is also consistent with results from the Caruana paper.

3.3 Stacked Ensemble Model

Finally, we present the accuracy of the stacked ensemble in comparison with the best-performing model from the dataset.

Table 4: Comparison of best individual model and stacked model accuracies

Dataset	Best Individual Model	Accuracy	Stacked Model Accuracy
Wine	Random Forest	0.8316	0.8231
Bank	XGBoost	0.8893	0.8907
Credit	XGBoost	0.8244	0.8223

As we observe, the stacked ensemble model does not perform much worse than the best individual model, and in the case of the most complete dataset actually performs better than any individual model. It is worth noting final binary classifier model outperforms all accuracy benchmarks from the original paper (Caruana & Niculescu-Mizil, 2006).

4 Conclusion

4.1 Summary

This study presented a comprehensive comparison of three prominent machine learning models—boosted trees, random forests, and support vector machines across three distinct datasets, each adapted for binary classification tasks. The study revealed that while Random Forest and XGBoost consistently showed robust performance across the datasets, the effectiveness of SVM varied depending on specific dataset characteristics. Notably, the SVM tended to underperform on datasets with fewer categorical features and a predominance of continuous variables.

Our exploration of a stacked ensemble model demonstrated its potential to incorporate the strengths of each base model, leading to enhanced predictive accuracy in some cases. Our findings point towards the utility of combining algorithms to improve prediction accuracy, especially when resource usage is not a concern.

4.2 Future work

Future work should focus on expanding the diversity of datasets and exploring more complex ensemble strategies. In addition, we only test on 3 well-proven classifiers. It may be possible that incorporating weaker models into a stacked ensemble may inadvertently improve its accuracy score as it is able to take weaker predictions into account.

5 Bonus Points

5.1 Hyperparameter tuning

The project undertook a rigorous hyperparameter tuning process involving 3-fold cross-validation testing. As noted in the attached Jupyter notebook, optimizing the parameters for each model on each dataset took a total of 13 hours running on a newer Apple laptop dedicated to machine learning applications.

5.2 Stacked Ensemble Model

To enhance model performance, a stacked ensemble model was developed to integrate the strengths of the individually tuned models. This approach combined the predictions of the base learners to produce a final output that is more accurate than any single model's prediction and exceeded the highest benchmark presented in the reference paper Caruana & Niculescu-Mizil (2006).

References

Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 161–168, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143865. URL `https://doi.org/10.1145/1143844.1143865`.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794. ACM, August 2016. doi: 10.1145/2939672.2939785. URL `http://dx.doi.org/10.1145/2939672.2939785`.

Cerdeira A. Almeida F. Matos T. Cortez, Paulo and J. Reis. Wine Quality. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C56S3T.

Andrii Gozhulovskyi. Choosing a model for binary classification problem, Sep 2022. URL `https://medium.com/@andrii.gozhulovskyi/choosing-a-model-for-binary-classification-problem-f211f7a4e263`.

M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998. doi: 10.1109/5254.708428.

Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pp. 278–282. IEEE, 1995.

Kolby Nottingham Markelle Kelly, Rachel Longjohn. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Ammar Mohammed and Rania Kora. A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University - Computer and Information Sciences*, 35, 02 2023. doi: 10.1016/j.jksuci.2023.01.014.

Rita P. Moro, S. and P. Cortez. Bank Marketing. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5K306.

I-Cheng Yeh. Default of Credit Card Clients. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C55S3H.