Analysis and Comparison of Three Classifier Algorithms on Three Datasets

Audrey Liang [1]

Abstract

This paper is a partial implementation of a 2006 paper by Caruana and Niculescu-Mizil (Caruana & Niculescu-Mizil, 2006). It is implemented as a part of the final project for the course COGS 118: Supervised Machine Learning Algorithms, taught by Professor Tu Zhuowen. The paper will look at three different classification algorithms, performed using three different splits, and on three different datasets taken from the UC Irvine Machine Learning Repository.

1. Introduction

The source code to this paper can be found at:

`https://github.com/oodball/ML-implementation.git`

Classification algorithms are extremely crucial in machine learning, where the intent is to classify and find patterns in places where we might not be able to discern using the human eye. Classification has historically been a crucial pillar of machine learning, since it is simply human nature to find patterns in everything.

This paper aims to implement three classification algorithms to evaluate their performance on binary classification tasks: Random Forest, Decision Tree, and Support Vector Machine (SVM). A full breakdown of each classification will be provided in **Section 2: Methodology**.

This was conducted as part of a final assignment for COGS 118A to gain experience with implementing classification algorithms on real-world datasets, taken from the UC Irvine Machine Learning Repository, which is widely used in machine learning research. The datasets were ensured to have no missing values, and they are adjusted for binary classification. In order to show representative performance, cross-validation was performed accross three different partitions.

To comply with course requirements, these implementations were taken by Caruana et.al's 2006 paper (Caruana & Niculescu-Mizil, 2006) on evaluating performances of different algorithms. Each algorithm was trained under three test partitions: 80/20, 70/30, and 50/50, using a averaged accuracy amongst the three splits to determine performance evaluation.

The primary objectives of this paper are:

- To implement and evaluate the performance of Random Forest Classification, Support Vector Machine Classification, and Decision Tree Classification Algorithms.
- To analyze the impact that a dataset has (Data size, instances, features, etc.) on model performance
- To compare said results from the other two objectives with the results found in the original paper (Caruana & Niculescu-Mizil, 2006)

Through the project's completion, I will build a greater understanding of the nuances between different algorithms and the effect of different datasets on model performance.

2. Problem Description:

The objective of this report is to evaluate and compare the performance of three machine learning classifiers on binary classification: Decision Tree, Random Forest, and Support Vector Machine. Each dataset, taken from the UC Irvine Machine Learning Repository, was selected based on their variety of features and instances to give a better understanding of each algorithm and their nuances. This report focuses on finding the best parameters and algorithms for each datasets.

The problem is to train classifiers to predict labels from previously unseen data to see how accurate they come out to be compared to prior work.

3. Methodology

All algorithms were implemented using the scikit-learn Python libraries. The algorithms below were implemented

[1]University of California San Diego, Department of Cognitive Science. Correspondence to: Audrey Liang <ayliang@ucsd.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

to the best of my ability to be as similar to the original parameters, although there will be some differences since I am using a different implementation method than the paper uses. I also kept the random state consistent between algorithms at 42, 43, and 44, since I didn't want it to be randomized if I was cross-comparing algorithms

3.1. Learning Algorithms:

SVM: This algorithm was selected since this was an algorithm we covered in class, so I was interested in implementing it on applicable datasets. I took the parameters from the papers, running each dataset on the 'linear', 'poly', and 'rbf' kernels with a degree of 2 and 3 for 'poly', widths of 0.001,0.005,0.01,0.05,0.1,0.5,1,2, Regularization widths from 10^{-7} to 10^{-3}

Random Forest: This algorithm was selected since it had a high performance evaluation in the original paper. I used the same metrics provided in the paper, with a max feature selection of [1, 2, 4, 6, 8, 12, 16, 20], and a tree size of 1024.

Decision Tree: This implementation used the basic scikit-learn library DecisionTreeClassifier.

3.2. Datasets Description:

Two of these datasets, the Diabetes and Parkinsons datasets, were intentionally selected since I plan to continue graduate studies in applying advanced technologies to the healthcare system. A comprehensive description will be given below.

Dataset 1: CDC Diabetes Health Indicators

This data set is a classification task dataset in the Health and Medicine area. It contains integer and binary data, so I did not do any preprocessing besides using StandardScaler from the sklearn.preproccessing library.
The CDC Diabetes Health Indicators as described by the UCI ML Repository contains lab test results, demographics, and survey questions, all to determine whether a person is diabetic, prediabetic, classified as 1, or healthy, classified as 0.

Number of Features: 21
Number of Instances: 253,680

Dataset 2: Rice (Cammeo and Osmancik)
This dataset is also a purely classification task in the Biology area. It contains information such as feature inferences on two different rice types to determine if a grain of rice is of the Cammeo variety or Osmancik variety.

Number of Features: 7
Number of Instances: 3810

Dataset 3: Parkinson's

This dataset is also in the Health and Medicine realm, with data taken from 31 people, 23 confirmed to have Parkinson's disease. This dataset is the smallest out of the three, but it has the most features in the dataset, which is why I chose it, since I wanted to see the difference between this and the Diabetes dataset, since they have similar feature numbers. The dataset is determined by whether a patient has Parkinson's Disease, classified by 1, or is healthy, classified by 0.

Number of Features: 22
Number of Instances: 197

3.3. Experiment Design

The experiment was conducted in Visual Studio Code (VSCode) in a Jupyter Notebook environment. In the source code, each classifier is run on its own notebook, with each of three datasets run inside each notebook.

The format structure of each notebook is described below:

- **Installations**
  A code block installing all the necessary libraries and fetching all datasets and a global variable for the training partitions. For the SVM notebook, extra global variables, such as the radial widths, kernels, regularization values, poly degrees, and n estimators was added into this section. For the Random Forest notebook, an additional max features variable was put in for future reference.
- **K-Fold Cross Validation Function**
  An extra cross validation step that finds the best hyper=parameters for the model and returns a cross-validation score, as well as the best parameters to use.
- **Model Evaluation Function**
  A function to train and test the data after cross-validation. An accuracy score is returned
- **Accuracy Curve Plot Function**
  A function to plot the train and test accuracy of each of the nine trials. A plot is returned.
- **Dataset 1 Results and Visualizations**
  The results of running the functions on the Diabetes dataset.
- **Dataset 2 Results and Visualizations**
  The results of running the functions on the Parkinson's dataset.
- **Dataset 3 Results and Visualizations**
  The results of running the functions on the Rice dataset

Each dataset was run 3 times for a total of 9 trials per dataset. With the exception of the SVM Classifier on the Diabetes dataset, every dataset was put through the same K-Fold

Cross validation. Even with a GPU, the model had taken over 4 hours to get through one trial out of nine, so the cross-validation was taken out for this reason.

Additionally, similar problems were seen with the Random Forest/diabetes dataset. It ran for nine hours, so there may be errors in the code as revisions were made afterward, although they were superficial ones such as the naming of certain plots. The data in the plot itself still very much reflects the data from the function run. In the Random Forest Classifier, errors were shown, and I was not able to have the time to run the dataset again, so the dataset was unfortunately further cut to 70,000 instances for the sake of time. While I should have chosen a different dataset, the server was down, so I decided to cut down on the already downloaded and processed dataset instead of cleaning a new dataset.

4. Results and Experiment

From the results run from the experiment, we see that Random Forest and SVM generally performed very similar to each other. With this, we can dive into an analysis of each classifier, as well as some nuances between each one that were noticed while implementing and developing the code.

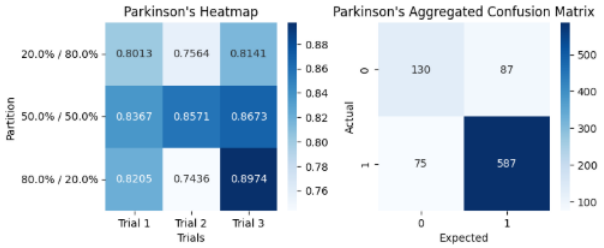

Figure 1. Heatmap of Accuracies for Parkinson's Dataset using Decision Tree, as well as an aggregated confusion matrix

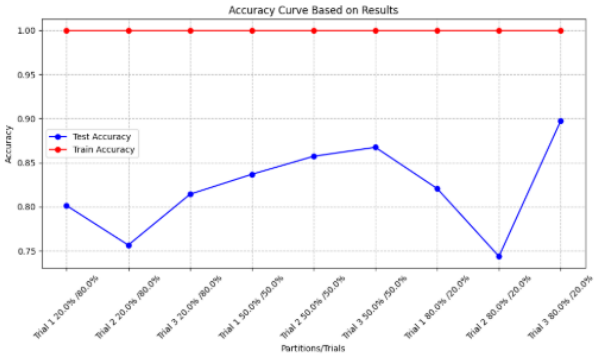

Figure 2. Accuracy Curve for Decision Tree classifier on the Parkinson's Dataset

Data set	Decision Tree	Random Forest	SVM
Diabetes	79.50	86.28	86.05
Parkinson's	82.16	87.08	89.30
Rice	89.27	92.95	93.24
Average	83.64	xx	89.53

Table 1. Classification accuracies for Decision Tree, Random Forest, and Support Vector Machine on various datasets.

4.1. Decision Tree

This classifier ranked third overall, as it was consistently the lowest ranking classifier. While it is the lowest, the accuracy scores were not actually that low, with the lowest being in the high 70s. It is important to note that this classifier ran the quickest out of all other classifiers, and had a solid performance, although it did not outrank the others. It is important to note that it did peak around the mid-size dataset, and did worst on the largest dataset.

After further digging online, I found out that the reason for this was that it was not optimized, and running Decision Trees on larger datasets is also more computationally complex. In future elaboration, I will try to optimize this classifier to fit this type of data.

4.2. Random Forest

Data set	Optimal Features	Optimal Split
Diabetes	50/50	2, 4
Parkinson's	50/50	1,4
Rice	80/20	1,2

Table 2. Optimal Statistics Per Datset

This classifier was ranked second overall, as it ranked highest in one of the three datasets. As the most computationally expensive classifier, it is important to note that the datasets took exponentially longer to run, even on a GPU.

However, the results were on par with those of the SVM classifier. Each of the accuracy scores did not score much lower than the SVM ones.

In further searching, I found out that Random Forest Classifiers do better on larger datasets, so having a dataset of 250,000 instances and 21 features, which is also the one that the classifier did the best on, makes sense.

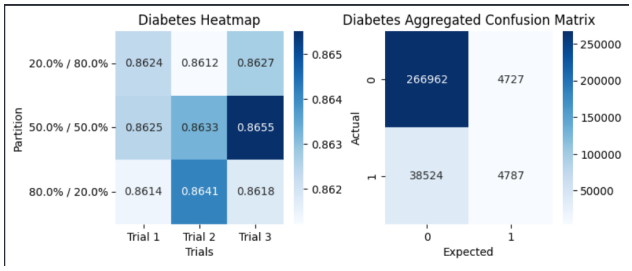

Figure 3. Heatmap of Accuracies for Diabetes Dataset using Random Forest, as well as an aggregated confusion matrix

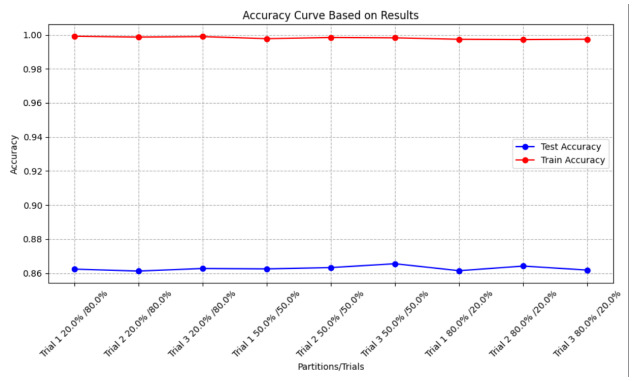

Figure 4. Accuracy Curve for Random Forest classifier on the Diabetes Dataset

4.3. SVM

DATA SET	SPLIT	KERNEL
DIABETES	80/20	N/A
PARKINSON'S	80/20	RBF
RICE	50/50, 80/20	LINEAR, RBF

Table 3. Optimal Statistics Per Dataset

Out of the classifiers, SVM ranked highest in two out of three of the datasets. Some things to note were that although the Parkinson's and Rice Datasets ran fast on this classifier even with the K-Fold Cross Validation, which was very similar to the Decision Tree Classifier. Although K-Fold tested every single hyper-parameter, the only dataset it seemed to have trouble with was the Diabetes dataset. This was the only model to have run on a different evaluation function, as running the classifier was too computationally expensive, similar to the Random Forest Classifier on the same model.

Deeper searching online showed that SVM works better on smaller datasets with more features, rather than larger datasets.

However, it is also important to note that the Aggregated Confusion Matrix showed that although there is a high accuracy score, this is because the prediction was just that every patient was predicted a 0 on the Diabetes dataset. In a future extension of this work, I would like to study the reasons why that is, and whether a poly or rbf kernel would be better suited, since no optimizing was done for the linear kernel on diabetes.

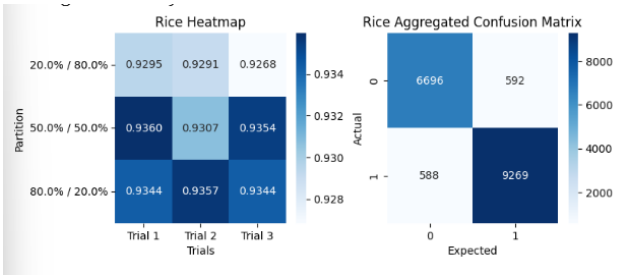

Figure 5. Accuracy Curve for SVM classifier on the Rice Dataset

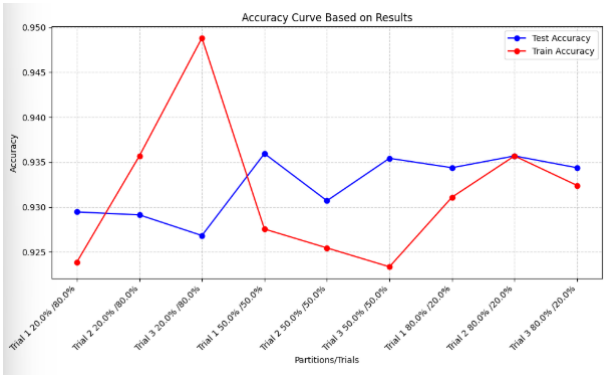

Figure 6. Heatmap of Accuracies for Rice Dataset using SVM, as well as an aggregated confusion matrix

Thus, with the results provided, here are the rankings of each classifier:

1. Support Vector Machine
2. Random Forest Classifier
3. Decision Tree Classifier

5. Conclusion and Discussion

This paper studies the performances of three different classifiers: SVM, Decision Trees, and Random Forest Classifiers.Through this paper and its results, we conclude that out

of each of three classifiers, SVM ranks the highest, followed by Random Forest, then Decision Trees.

Future work on this paper could extend to implementing a more optimized version of the Decision Tree Classifier, as this study does not optimize it, which is likely to cause a small shift in the accuracy scores. Additionally, the accuracy scores can be cross referenced with other scores such as the F1 Score. As seen in the SVM Diabetes Dataset, accuracy score is not necessarily the only score that should be used, as without a confusion matrix, I would not have otherwise known that the high accuracy score was due to the model predicting only 0.

Furthermore, this paper would have ideally been able to fully apply the cross validation to datasets of higher instances. In future work, I hope to be able to run datasets on a machine with higher capabilities. It is a shame that I was not able to run the full Diabetes Dataset on the Random Forest Classifier, as well as the K-Fold Cross Validation on the SVM dataset.

Also, future work could show more analysis on the classifiers, whether to look at the difference between the same classifier across different implementations, or to do a deeper analysis to see if there are any problems with the classifiers (ex. over-fitting, noise, etc.)

While it is unfortunate that my work does not reflect the results shown in the original paper, I believe that this is a good foundation to iterate on to see if future work could work towards those results seen before.

6. Bonus Points:

I believe that I deserve some bonus points due to my implementation of the K-Fold Cross Validation function, rather than importing a Grid Search CV library. I wanted to fully understand the reasons for tuning the hyper-parameters, so I wrote a customized function for the Random Forest Classifier and SVM which tests each hyper-parameter given in the paper for those two classifiers.

Additionally, visualizations are added to see the data through a confusion matrix on top of the heatmaps and accuracy curves. In doing so, I believe my data visualizations have been heightened to where the data are not just shown by accuracy scores, but also with a better understanding of the predictive analytics. Without these matrices, I would not have been able to catch the SVM error in the Diabetes Dataset.

Thank you for this quarter!

References

Caruana, R. and Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, 2006. URL https://www.cs.cornell.edu/~caruana/ctp/ct.papers/caruana.icml06.pdf.

Contributors, E. *Data Science Random Forest*, 2024a. URL https://emeritus.org/in/learn/data-science-random-forest/.

Contributors, S. Is svm a good choice for large datasets?, 2024b. URL https://datascience.stackexchange.com/questions/123647/is-svm-is-a-good-choice-for-large-dataset.

Cınar, A. and Köklü, R. Classification of rice varieties using artificial neural networks. *Semantic Scholar*, 2024. URL https://www.semanticscholar.org/paper/Classification-of-Rice-Varieties-Using-Artificial-C4%B1nar-Koklu/4e508bb906c8fdc04ead6f20bd8918fcb3

Scikit-learn. *Random Forest Classifier*, 2024. URL https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

(Cınar & Köklü, 2024; Caruana & Niculescu-Mizil, 2006; Scikit-learn, 2024; Contributors, 2024b;a)