Advertisements and Heart Problems

An objective look at classifier performance for predicting advertisements and heart arrhythmia

Jeff McGehee
Georgia Institute of Technology
CSC 7461: Machine Learning
Atlanta, GA
jlmcgehee21@gatech.edu

***Abstract*—This short paper seeks to present an analysis of the performance of various classification algorithms as tested on the "Internet Advertisements" and "Arrhythmia" UCI datasets. In order to understand the performance of each algorithm on the above problems, a statistical analysis of both sets will be presented. Following this analysis, an individual analysis on the performance of the following algorithms will be conducted: Decision Trees, Adaboost Boosted Decision Trees, K Nearest Neighbors, Support Vector Machines, and Artificial Neural Networks.**

Introduction

Classification is the typical machine learning problem [1]. Numerous algorithms have been developed to address this problem. This paper will evaluate an assortment of these algorithms as they perform on two classification problems. The problems were selected carefully in order to ensure that they are capable of being learned yet difficult enough to allow comparison of each algorithm.

All analyses performed in this summary were done using machine learning tools available for Python. The popular Python package SciKit Learn was used to implement all algorithms, with the exception of Neural Networks, which were implemented with PyBrain [2, 3]. All algorithm results presented were found using 10-fold cross validation.

Dataset Information

Preprocessing

Before testing any algorithms, it is often necessary or advantageous to perform some sort of preprocessing on the datasets. In the case of both of these sets, all samples with missing values were replaced by the average of their respective features using the SciKit Learn Imputer class, and all data was normalized.

Problem 1: UCI Internet Advertisements Dataset

Internet advertisements can be distracting, deceiving, and can greatly alter a user's perception of the internet. Because of this, many tools have been developed to detect the presence of internet advertisements in html pages and prevent them from being rendered by the user's browser. The "Internet Advertisements" dataset was used to develop such a tool [4]. The set contains information about images enclosed in html <A> tags. The original study using this dataset only presented the effectiveness of C4.5 decision trees. In this study, the authors were able to achieve an accuracy of approximately 97%. It will be desirable to compare the optimized CART algorithm implemented in SciKit Learn to the results of the original study, and to use this 97% accuracy as a benchmark for the other algorithms as well.

The dataset itself contains a simple binary set of classes: **ad**, and **non-ad**. After removing samples with missing data, the distribution of the set is 86% ads, 16% non-ad over a total of 3279 samples. This distribution qualifies this problem as "imbalanced". Imbalanced datasets are quite common and there are varying opinions about how they should be treated [5]. For this paper, it will simply be noted that the set is imbalanced, which may allow for better interpretation of the results.

The feature portion of the dataset consists of 1558 features, most of which are binary encoded text data. A more detailed description can be found at the UCI repository [6].

In order to explore the dataset further, a decision tree was used to determine the feature importances, then the distribution of the two most important features was plotted for each class. This data can be seen in Figure 1. It would be ideal to view a plot like this for all features in the dataset, but due to the large number of features, this was not possible.

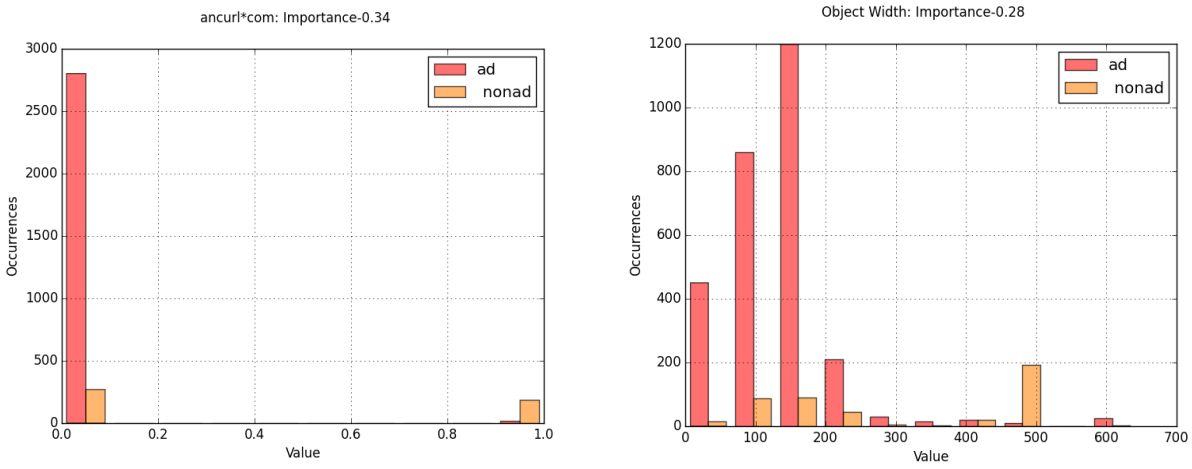

Figure 1: "Internet Advertisements" Feature Analysis

The most important feature listed is the presence of the string "com" at the end of the hyperlink url. It can be seen that nearly *all* occurrences of ad images end with "com" while it is much less biased for non-ad images.

Interestingly, the second most important feature is simply the image width. According to the dataset, if the image is greater than 300px wide, it is almost certainly not an ad.

This initial analysis shows that this set should be rather learnable as suggested in [4]. However, it also suggests that there are a large number of unimportant features in the dataset. In fact, the decision tree analysis of feature importance showed that less than 1% of the 1558 features were determined to have an importance greater than 1%. This means that a successful algorithm must be able to find the important features and ignore the large number of unimportant ones to be successful.

Problem 2: UCI Arrhythmia Dataset

The UCI "Arrhythmia" dataset was recorded in order to test classification of various levels and types of cardiac arrhythmia. Machine learning research is quite popular in the medical

field because there are often poor doctor/patient ratios and the amount of medical data recorded is so huge that managing this data has become its own industry [7]. Any tools that can be borne out of this data that will help doctors decipher patient information faster, could be very valuable.

The "Arrhythmia" dataset was originally used to test an experimental classification algorithm known as VFI5. According to the publication from this previous work, VF15 was able to achieve an accuracy of 68%. For reference, the authors also published the accuracy of the Naive Bayesian classifier (50%) and the K Nearest Neighbors classifier (53%) [8]. Similar to the "Internet Advertisements" dataset, these performance numbers will serve as benchmarks for the performance of the classifiers that will be presented in this paper.

The dataset contains 452 points which can fall under 16 possible classes for cardiac arrhythmia. However, three of these classes are not present in the dataset which leaves 13 possible classes as seen by the classification algorithms. A descriptive list of these classes can be seen in [6]. Similar to the "Internet Advertisement" dataset, the "Arrythmia" data is heavily imbalanced towards one single class (Figure 2). However, there are many more classes to choose from.

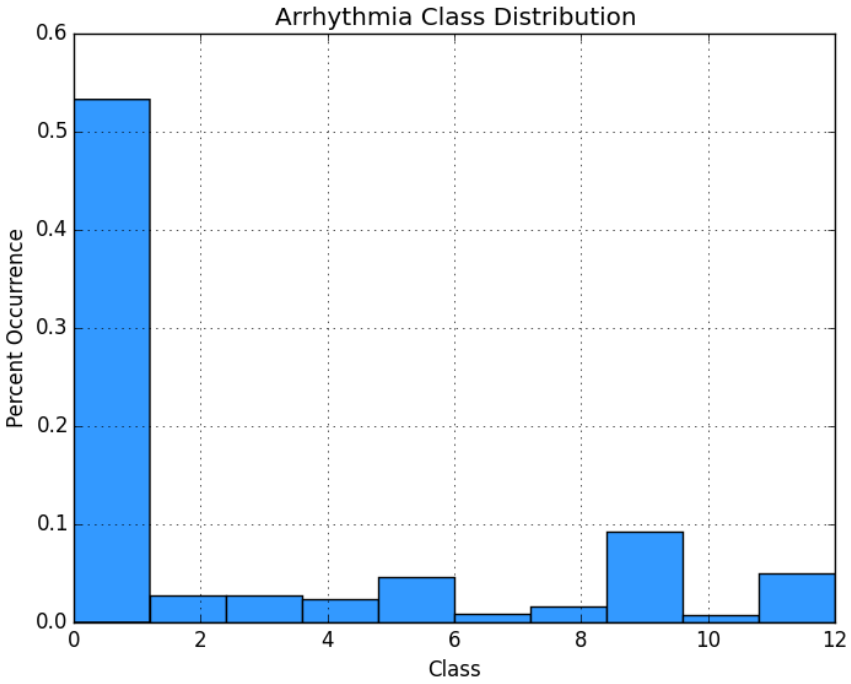

Figure 2: "Arrhythmia" Class Distribution

In order to be as concise as possible, the “Arrhythmia” dataset is being described as containing two main sets of features that are broken down to form 279 features in total. The first set is human readable data such as age, sex, height, weight, and heart rate. The other is ECG data from various channels and metrics that are derived from that data. If the reader wishes to examine the features in more detail, a detailed description is available in the UCI repository [6].

In a similar manner to the “Internet Advertisement” set, a feature importance evaluation was performed on the “Arrhythmia” dataset using a decision tree. The distribution of the two most important feature values for each class can be seen in Figure 3.

The two most important features of the “Arrhythmia” data are heartrate (12%) and T wave amplitude of channel v6 of the ECG—shown as “277” in Figure 3 (9%). It can be seen that for these features, the distribution does vary, meaning it should be possible to learn this dataset to some degree. However, the importances of the features are relatively low compared to what can be seen in highly learnable datasets like “Internet Advertisement”. This means that, at least for the decision tree used to generate the importances, it is difficult to find individual features that give a clear indication of a sample's proper class

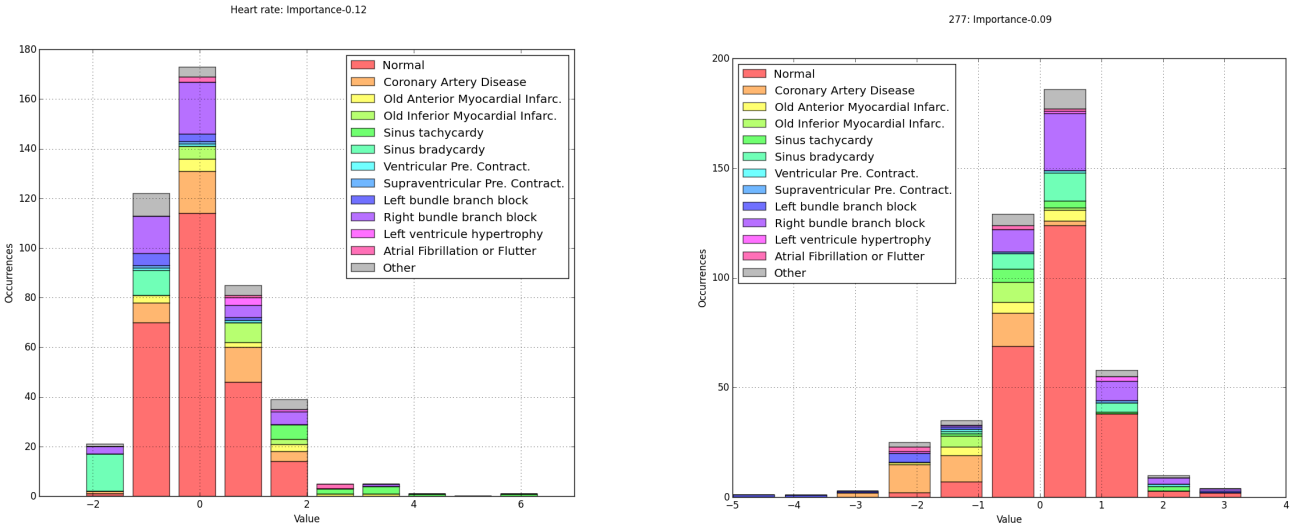

Figure 3: “Arrhythmia” Feature Analysis

Decision Trees

For both problems described, CART decision trees were created using SciKit Learn. The decision trees were trained and tested using 10 fold cross validation. Below is a plot of the cross validation learning curves for increasing numbers of training samples. The curves were generated for both "pruned" and unpruned versions of the decision trees. The training accuracy curves were left off of this graph because they were indistinguishably close to a perfect score for every sample.

"Pruned" is used in quotations above because SciKit Learn does not support traditional decision tree pruning. Instead, a limit was imposed on the minimum number of samples required to create a new branch of the tree. This was used with some degree of success, but it would be interesting to see the results of traditional pruning.

According to Figure 4, the top performing CART decision tree for the "Internet Advertisement" set achieved an accuracy of 97%. This is on par with the results of the original study performed with the C4.5 decision trees. The top scoring decision tree for the "Arrhythmia" set achieved an accuracy of 70%, which is better than the custom VFI5 created by the authors of the original study on the set.

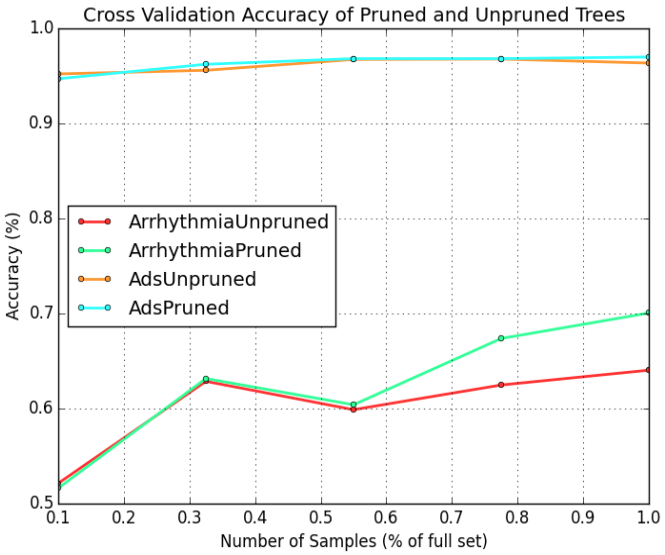

Figure 4: Pruned vs. Unpruned Decision Trees

Decision Trees Boosted with Adaboost

The Adaboost algorithm was applied to the CART decision trees using both 25 and 50 estimators. For both datasets, a slight improvement was reached over the results for the non-boosted decision trees (Figure 5). Similar to the non-boosted decision trees, the classifiers are extremely effective at learning the training set, but are not necessarily capable of creating a generalized model that explains the complexity of the data overall. This is especially true for the "Arrhythmia" data. The Adaboost algorithm simulation took a significant amount of time to run (approximately 15 minutes). This is because the simulation was required to train 25, then 50 times more decision trees than the unboosted decision tree model.

As mentioned, the results of the Adaboost-ed learners show a slight improvement over the non-boosted decision trees. The reason this is not greater is because Adaboost is made to improve *weak* learners. The CART decision trees already proved to be *strong* learners on these two problems, performing nearly perfectly on their training data. The accuracy issues seen on the "Arrythmia" testing are not because the trees are weak learners, but rather because the dataset is too complex for the CART algorithm to build trees capable of generalizing beyond the training examples. This cannot be helped by the Adaboost algorithm.

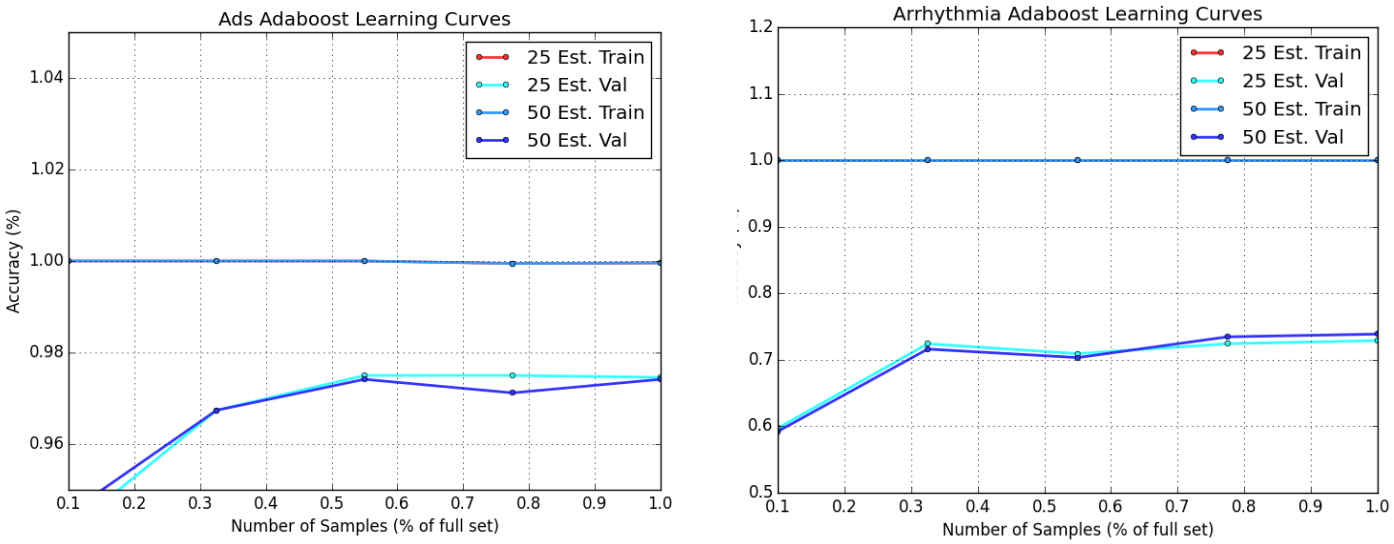

Figure 5: Adaboost Performance

K Nearest Neighbors

K Nearest Neighbors, or kNN, is a relatively simple algorithm that takes advantage of the fact that a given sample will likely fall into the same class as samples that are similar. This can make it difficult for kNN to understand a very complex learning space without sufficient samples. An analysis of kNN was performed using various numbers of linearly weighted (by Euclidian distance) neighbors. The results can be seen below.

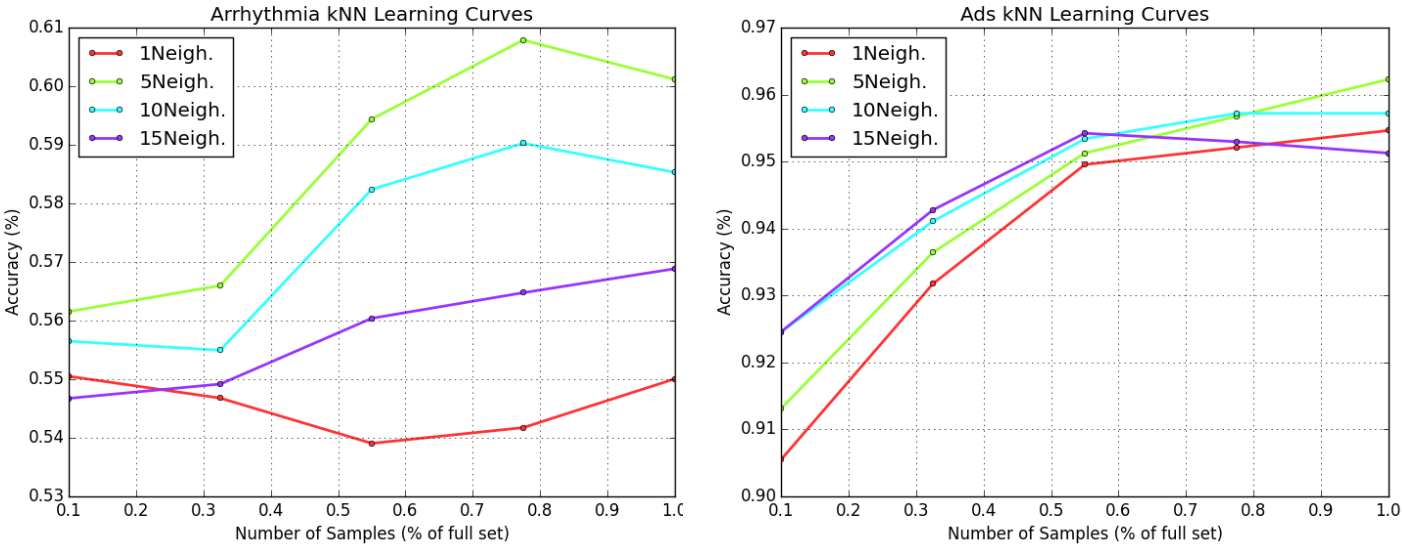

Figure 6: kNN Performance

Again in these figures, the training data is not shown because kNN will always perform perfectly on it's training samples. However, the kNN algorithm displays the worst performance on test data of any method explored thus far. For both problems, kNN's performance was below that of the reference learners originally studied with each dataset. It can also be seen that the value chosen for k has a relatively large impact on the performance of the algorithm. Of the k values shown here, both cases achieved optimal performance with 5 neighbors. In a more thorough study it might be advantageous to investigate k values between 5 and 10 to see if any greater performance gains can be had.

Support Vector Machines

Support Vector Machines, or SVMs, are classifiers that represent each data sample as a point in space, and through the "kernel trick", can map a complex n-dimensional dataset to a simpler space where the data may be more easily separated into its respective classes.

SVMs are binary classifiers by nature, but can be adapted to multiclass classification problems. This is done by combining multiple binary classifiers into one predictor. SciKit Learn uses the "one-against-one" method which is described in [2]. The results for SVM classifiers using a few popular kernel types can be seen in Figure 7.

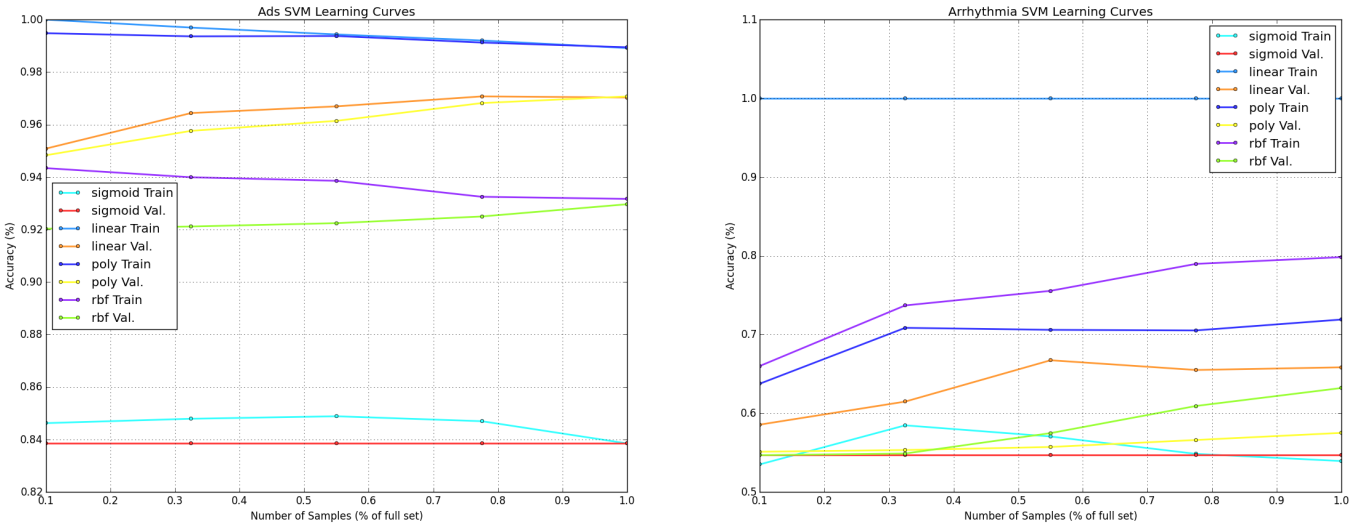

Figure 7: SVM Performance

For the "Internet Advertisement" problem, the SVM classifier had a best of 97% accuracy, which was achieved with the linear kernel. This is very similar to the performance of the decision tree algorithms. For the "Arrhythmia" dataset, the SVM with the radial basis function (RBF) kernel was able to achieve 80% accuracy, which will prove to be the best of any classifier tested for this study. Because of the nature of the "one-against-one" method, the SVM training took the longest of any algorithm. The simulation for the SVM took around 25 minutes in real time. However, it did yield the most accurate results of any classifier being tested on the "Arrhythmia" dataset.

Artificial Neural Network

For this test, a set of Neural Networks were generated using various numbers of hidden nodes. Each network has only one hidden layer and uses a Sigmoid activation function on that layer. Various numbers of hidden layers and activation functions were tested, but the authors found this configuration most successful. All networks were trained for 50 epochs using the RProp training algorithm. The results can be seen in the figures below.

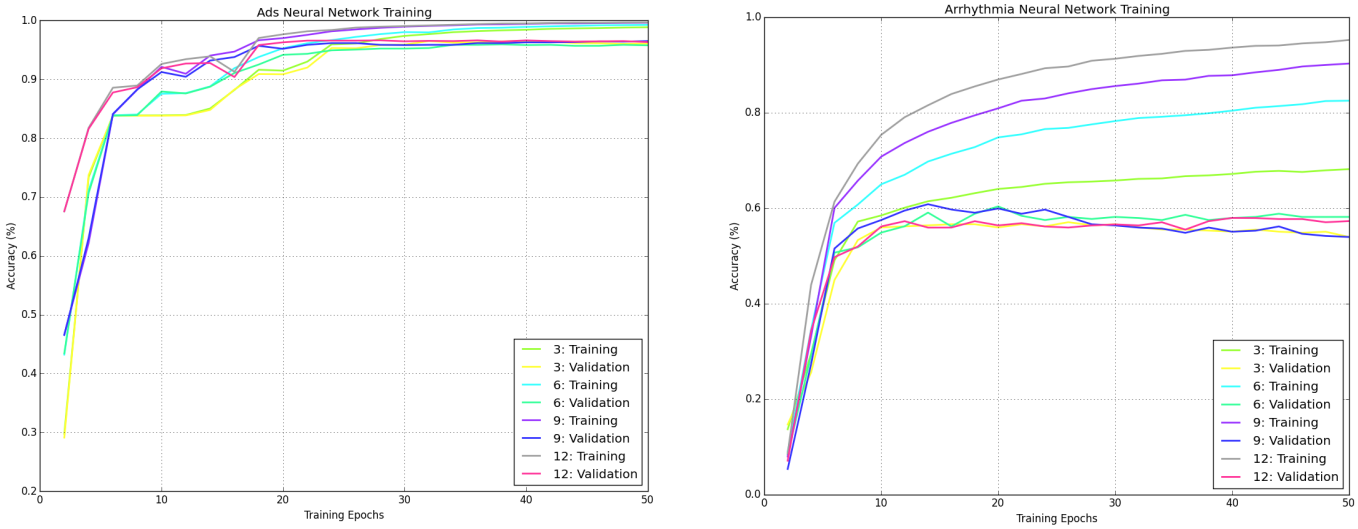

Figure 8: ANN Performance

On the "Internet Advertisement" set, the ANNs performed at the same level as all of the other classifiers presented so far, with a test accuracy of 97%. On the "Arrhythmia" data, the ANNs were the worst performing of all the classification algorithms with a test accuracy of 59%. A second test was run on the "Arrhythmia" set with 100, 150, and 200 hidden nodes. This did yield slightly better performance, but not enough to warrant the extra training time.

Conclusions and Final Analysis

In this study, five algorithms were tested on two separate classification problems: the UCI "Internet Advertisements" dataset and the UCI "Arrhythmia" dataset. Each dataset was originally the focus of a machine learning publication from which a baseline performance expectation could be determined. In the chart below, a summary of each algorithms performance is presented, along with their relative performance compared to the problem's respective reference.

Table 1: Performance Summary

Classifier	***Advertisements***	***vs Reference***	***Arrhythmia***	***vs Reference***
Decision Tree	97%	+0%	70%	+2%
Adaboost	98%	+1%	74%	+6%
kNN	96%	-1%	60%	-8%
SVM	97%	+0%	80%	+12%
ANN	97%	+0%	59%	-9%

All algorithms performed well on the "Internet Advertisement" set. However, this was not the case for the "Arrhythmia" set. The author suspects there are several reasons for this. The main one being that the dataset has over half as many features as it does samples, and over 50% of these samples are a single class. This makes it very difficult for an algorithm to determine which features implicate a given class.

Though it is beyond the scope of this report, in order to achieve very high performance on the "Arrhythmia" dataset, a more complex learning strategy should be applied. It may be possible to perform feature selection to remove confusing features from the set. It could be useful to test more ensemble methods aside from the Adaboosted CART. In every case, it was shown that the algorithms were capable of learning the training data, but struggled with testing data. This indicates that there are simply not enough samples in the set to allow for a predictor to accurately predict unknown samples.

Bibliography

[1] L. Valiant, "A Theory of the Learnable," *Communications of the ACM,* vol. 27, no. 11, pp. 1134-1142, 1984.

[2] D. Cournapeau, J. Millman, et al. *SciKit Learn*. Computer software.*SciKit Learn*. Vers. 0.15. N.p., n.d. Web.

[3] J. Bayer, M. Felder, et al. *PyBrain*. Computer software. *PyBrain*. Vers. 0.3. N.p., n.d. Web.

[4] N. Kushmerick, "Learning to Remove Internet Advertisements", *University College Dublin*, Dublin, Ireland.

[5] J. Van Hulse, T. Khoshgoftaar, A. Napolitano . "Experimental Perspectives on Learning from Imbalanced Data" *International Conference on Machine Learning,* 2007.

[6] "UCI Machine Learning Repository" *UCI Machine Learning Repository*. University of California Irvine, n.d. Web. 31 Jan. 2015.

[7] G. Magoulas, A. Prentza. "Machine Learning in Medical Applications", *University of Athens,* Athens, Greece.

[8] A. Guvenir, B. Akar, et al. "A Supervised Machine Learning Algorithm for Arrhythmia Analysis ", *Bilkent University*, Ankara, Turkey.