Berry-Wilkins Technical Report

0. Abstract

In this report we analyze three different types of machine learning algorithms by using them on 4 varying datasets. Our goal was to gather some insight on how these algorithms function in different settings and how we should approach making predictions on new data that we gather in the real world. We find that the method we should use to make predictions on new data differs on the type of data that we gather. There is no ‘one size fits all’ solution to making predictions off of new data, and it very much matters on the kind of data we are working with.

1. Introduction

This project is motivated by the differences of machine learning models (Neural Networks, Decision Trees, and Naive Bayesian) and learning the advantages and disadvantages of each. To narrow the scope of the project, our datasets are all doing classification tasks of predicting labels with no regression tasks. The datasets we used for the experiments are:

* A data set describing two classes of robots using all nominal attributes and a binary label (monks1)
* A data set of optical character recognition of numeric digits from images (mnist\_100)
* A data set describing the voting histories of members of the U.S. Congress (votes)
* A data set describing patient health data that can be used to diagnose the health of a patient’s thyroid into four possible labels (hypothyroid)

The goal is to find which models make the most accurate predictions for each dataset and why. Each dataset has differences in the attributes that lead to the prediction that gets made and some, like the hypothyroid dataset, have multiple labels that need to be predicted. Some datasets have nominal attributes (where data for each category is a word or descriptor), numeric attributes (data for each category is a numeric measurement), or a mix of both. Each model is better at handling different types of attributes and representing the classification task at hand, so in reaching our goal we will have to discuss what models are best at analyzing the data of each dataset.

We are going to accomplish the goal by fitting each model with each dataset and measuring the confidence and accuracy of the predictions that each model produces. Each dataset was split into training and testing sets with 75% of the rows of data in the dataset going to training/fitting the model and 25% going towards the testing set. The training/fitting process teaches the model how the attributes of the dataset contribute to each label and the testing process tests the models predictions to see if they were accurate. We find the accuracy of the predictions that each model generates by seeing the label that the model predicted for each row of data in the testing set, comparing it to the actual label of that row of data, and finding the percentage of times that the model made a correct prediction. We find the confidence of the predictions by finding the margin of error between the predicted labels and the actual labels. Once we have the accuracy and confidence of the predictions that each model generates for each dataset, we will have a good idea of which models are more useful for the tasks represented by each dataset.

We found that, generally speaking, Decision Trees are best suited for generating predictions for the hypothyroid dataset, Neural Networks are best suited for generating predictions for the mnist\_100 and monks1 datasets, and each model is well suited for generating predictions for the votes dataset, but the Neural Network and Decision Tree perform the best.

1. Problem

Our 4 datasets have varying properties, so it is important for us to understand how these datasets different from each other in order to interpret our findings. Our datasets are as listed above: monks1, mnist\_1000, vote, and hypothyroid. Below is a table containing summary information of these four datasets as well as a description of what we are trying to predict.

|  | monks1 | mnist\_1000 | votes | hypothyroid |
| --- | --- | --- | --- | --- |
| # of attributes | 7 nominal | 28x28 continuous | 15 nominal but some missing values | 6 continuous  21 nominal  27 total |
| # of possible labels | 2 | 10 | 2 | 4 |
| Proportion of each label | 58/108 - 1  50/105 - 0 | 1/10 for all labels | 45/109 - rep.  64/109 - dem. | 3481/3772 - negative  194/3772 - compensated  95/3772 - primary  2/3772 - secondary |

* monks1 is a data set describing two classes of robots using all nominal attributes and a binary label yes/no. The dataset has a simple rule designed to train machine learning algorithms. If body\_shape=head\_shape or if jacket\_color=red, then yes, else no.
* Mnist\_100 is a data set containing instances of a 28x28 display that contains handwritten numbers. The object is to label each instance with the number that was hand-written on it.
* Votes is a data set that contains the voting histories of members of the U.S. Congress. We are aiming to predict how a U.S. congress member will vote given 15 different nominal attributes.
* Hypothyroid is a dataset containing health information of 3772 patients for which given their health information, we aim to predict cases of hypothyroid.

1. Solution

To make predictions on new data, we used three different methods: Neural Networks, Decision Trees, and Naive Bayes.

1. Neural Network

Neural Networks make a prediction by taking an input, and running each attribute value through multiple layers of nodes. Each node takes in weights from all the nodes from the previous layer and then outputs a weight to each node in the next layer. To determine the output weight of a node, we use logistic regression with all the given weights as the output. The layer structure is categorized into three regions: The input Layer, the hidden layers, and the output layer. The input layer has as many nodes as there are attribute values for each instance in the data. The hidden layers come after the input layer. There can be as many hidden layers as we choose, with an amount of nodes that we choose as well. The output layer is typically one node signifying the prediction.

1. Decision Tree

Decision Trees function by trying to determine which attribute is the most important and then determining what the next most important attribute is given the value for that attribute. This continues to branch in a tree-like fashion until we generate leaves, which are our predictions. We determine which attribute is most important by calculating the gain of each attribute and selecting the attribute with the highest gain value.

1. Naive Bayes

Naive Bayes functions by asking “what is the most probable to be the label?”. Instead of constructing a complex model to transform an input to a predicted output, Naive Bayes calculates the most probable label given the input by using probability theory and the assumption that all of the attributes are independent from one another. Every attribute being independent means we can represent the probability of a label given the attributes as a probability equation rather than a model. This makes the process of making a prediction much simpler and easier to understand.

1. Experimental Setup

For this experiment, we ran each of our three machine learning algorithms on all four of our datasets and recorded the accuracies and recalls. We are measuring the accuracy of a machine learning algorithm on a dataset by calculating the proportions of correct predictions over the total number of predictions. We are measuring the recall of each label in a dataset for each machine learning algorithm by calculating the proportion of times we correctly guessed that label over the total number of times we guessed that label.

For the ease of testing, we used scikit-learn’s implementations of a Neural Network (MLPClassifier), Decision Tree (DecisionTreeClassifier), and Naive Bayes (GaussianNB). For all of the algorithms we are using 75% of the datasets for the training set, and 25% percent for the test set. For our random seeds we used the numbers 0-49. For the neural network, we use scikit-learn’s default number of neurons, number of hidden layers, and learning rate of 100, 15, and adaptive respectively. An adaptive learning rate here means the algorithm starts with a learning rate of .001 but then will change this value if loss is not being minimized.

1. Results

| Figure 1 | Figure 2 |
| --- | --- |
|  |  |
| Figure 3 | Figure 4 |
|  |  |

There are several observations of note in our findings, but the most important of them is that the neural network outperformed the other machine learning models on half the datasets while Naive Bayes consistently underperformed.

For the Hypothyroid dataset, the Decision Tree generated the most accurate predictions, the most consistently followed closely by the Neural Network and far surpassing Naive Bayes. This dataset has a mix of continuous and categorical attributes, meaning that some of the columns of data were descriptive and others were numeric. Scikit-Learn documentation maintains that the optimized CART (Classification and Regression Trees) algorithm that we used for the Decision Tree model does not support categorical attributes, though it seems to perform very well with a mix of continuous and categorical attributes. High accuracy with decision trees usually means that the algorithm was able to learn the important factors that go into predicting the labels, but it can also be a sign of overfitting (training the model on data that is too similar so it does not learn much after initial training) which means it would not be likely to accurately predict the label for rows of data that are unlike what it has encountered before. The Neural Network was consistently a little bit worse, likely because it had a harder time handling the mix of attribute types than the Decision Tree. Naive Bayes performed terribly, this is likely because it could not handle the mix of attribute types as well but also this kind of model is not modeled well for this kind of classification task, where the distribution of the labels is not necessarily gaussian.

For the mnist\_1000 dataset, the Neural Network generated the most accurate predictions, the most consistently followed distantly by the decision tree and with the Naive Bayes model even further behind. Conceptually, this is the kind of classification problem a Neural Network would be the most suited for. The individual weights of the Neural Network’s layers seem to have an easier time learning the relevance of each data point than the long and branching paths that the Decision Tree has to build. The Naive Bayes model performed better than the previous dataset, this is likely due to the probability of the occurrence of each number (1 through 8) having a more gaussian distribution than the possibility of one of four hypothyroid states.

For the monks dataset, the Neural Network generated the most accurate predictions with staggering consistency of 100% correct predictions. The results for this dataset are interesting because we are starting to see high variance in accuracy between random seeds, with the Decision Tree’s predictions being as accurate as 100% and as inaccurate 75%. It could be that the Neural Network had an easier time learning with fewer iterations while the Decision tree requires more data to fully map out the rules. This is interesting because, conceptually, a Decision Tree should be very accurate when working with datasets with simple rules ( if head\_shape = body\_shape OR jacket\_color = red, then yes, else no) such as monks. The Naive Bayes model had about the same average accuracy as the worst run of the Decision Tree, this makes sense because a probabilistic approach would have a hard time learning the rules, but it could handle the categorical attributes of the dataset.

The votes dataset is the only dataset with no clear best model for generating predictions. The average accuracies for the Neural Network, Decision Tree, and Naive Bayes models respectively are 95.1%, 94.3%, and 94.6% for all 25 seeds rounded to the nearest tenth. With the high variance in all of the models’ accuracies on this dataset, testing the accuracy of the models on more seeds would likely lead to different models pulling slightly ahead of the others just to fall back behind. All of the models had high average accuracies and each approach conceptually makes sense for this problem. The Decision Tree had the worst accuracy by a small amount, but it would be the easiest for the humans using its findings to understand than the other models, which are less transparent.

1. Conclusion and Future Work

To conclude, we set out to analyze the results of three different machine learning algorithms, on four different datasets, so we can understand how these algorithms function in the real world. We found that the best algorithm to use varies depending on the case. While neural networks typically get a higher accuracy, in some cases we want to focus on the recall in which case either a decision tree or naive bayes may be preferred. That said, what was most clear was when determining which possible solution is best, we have to analyze multiple algorithms and see which model best fits the data we are trying to work with.

As for future work, we believe that it would be helpful to analyze more algorithms on more datasets. This was a fairly small experiment designed to think about machine learning in a different way, but it is limited by its size. By expanding the amount of datasets, we could possibly draw more definitive answers for when to use a specific algorithm. At the end of the day, we want to know how machine learning can help people, and looking at more algorithms and more data will absolutely help in making progress towards answering that question.

We have adhered to the honor code on this assignment. -AW, MB