Predicting the Diagnosis of Alzheimer's Using Machine Learning

Daniel McCormick, Krysten Harvey, Chandler Barnes

Vanderbilt University, Nashville, TN

Abstract

For this project, we will be using the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset developed by the University of South Carolina located at http://adni.loni.usc.edu/ to build a neural network for classifying patients with Alzheimer's disease. The ADNI dataset contains 819 instances. These subjects have all been manually diagnosed as having Alzheimer's or not having Alzheimer's by professional doctors. In this paper, we will report summary statistics of the features in our dataset as well as the results of initial baseline and human experiments for classifying the instances.

Overview of ADNI Dataset[1]

Introduction

Alzheimer's disease is the sixth leading cause of death in the United States. Patients are usually diagnosed after the age of 65, but 5% of patients with Alzheimer's experience early onset of the disease between their 40s and 50s. Naturally, Alzheimer's symptoms usually go unnoticed for a longer period of time in younger patients, and these patients are often commonly misdiagnosed. Although Alzheimer's disease is incurable and irreversible, identifying symptoms and making a diagnosis in early stages of the disease is extremely beneficial to patients, as early treatment can potentially stop or slow its rate of progression.[1] We are interested in identifying key genetic and demographic risk factors as well as brain imaging trends that predispose patients to Alzheimer's disease in order to decrease misdiagnosis of patients with Alzheimer's and contribute to more personalized treatment plans that will lead to better outcomes for these patients. To do so, we plan on building a neural network that takes inputs including patient genotypes for the APOE gene, which is linked to Alzheimer's, demographic data (i.e. ethnicity, gender, age, marital status etc.), and brain volume data (i.e. Hippocampus volume, Whole Brain volume, caudate volume) and predicts the target concept of whether patients have Alzheimer's. Unfortunately, most of the available data is only for patients over the age of 65, but we believe that classifiers generated could still be useful in providing a diagnostic for younger individuals. Regardless, it will be a useful diagnostic tool for older individuals who may have Alzheimer's. This is a classification problem since the patients in our dataset either have or do not have Alzheimer's and a severity isn't listed.

For this project, we will be using the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset developed by the University of South Carolina located at http://adni.loni.usc.edu/. The ADNI dataset contains 819 instances, with each instance mapping to an elderly patient. These subjects have all been manually diagnosed as having Alzheimer's or not having Alzheimer's by professional doctors. All of these instances are currently available, and data collection has been completed over a six year span, using $67 million of public funding. Use of this data is protected by the ADNI data use agreement which among other restrictions requires that researchers not seek to obtain the identity of individuals from which the data was derived. Other than the ADNI data agreement, no other regulations apply.

Diagnosis	COUNT
AD	690
CN	1660
LMCI	2598
Grand Total	**4948**

Since the data features involving genotypes and demographic data are constant, easily measured through simple tests and surveys, and constant for each given

subject, it is unlikely that there will be significant noise in these features. The larger potential source of noise, however, is the brain imaging data. Since the ADNI dataset uses functional imaging which measures brain activity in certain areas, it can fluctuate a fair amount. This results in data that can vary a fair amount. Since the labeling of patients having or not having Alzheimer's is done by professional doctors, it is very likely that all of the instances are classified correctly and that no significant noise exists in the labeling. Detailed breakdowns of the data can be found in the appendix.

In a real world setting, if a supervised learning algorithm were able to learn a successful classification of similar features in a patient's record, healthcare providers could maintain on-the-fly metrics of the patient's expression of Alzheimer's disease. As well, doctors could see tangible results of treatments for Alzheimer's in their patients. Patients at risk of Alzheimer's would have more certainty in their levels of risk based on the classification of the algorithm. Thus, since patients classified as having Alzheimer's would be at a higher risk of actually having the disease, they should be submitted for further tests. While false positive reports of a patient being at risk for Alzheimer's based on their classification could lead to potential unneeded treatment, the false negatives of the algorithm are much more hazardous. Therefore, the algorithm should have a false negative rate of under 5% to be useful. Alzheimer treatment can be life-saving if begun early while rendered inert if the disease is given too much time to progress.[2] The false negative rate, a false bill of health for sick patients, should be minimized if all possible.

Train/Test Split

We are reserving 164 patients with 1,037 instances for our test set. Each instance represents a doctor's visit for some patient, so there are usually multiple instances per patient. We feel an 80/20 split to be appropriate for our dataset. To select this test set, we randomly selected 164 patients from the dataset and moved all of their corresponding instances to the test set. This ensures that the test set is independent from the training set by not allowing any patients to be in both sets, even if it is for different instances of them visiting the doctor. Even if there were some correlation between data points, a random sample should maintain an accurate model for the overall set. To ensure that no patient is overrepresented in the set, we randomly selected one data instance from each patient and removed the rest from the test set.

Initial Experiments

Baseline

We used a random classifier as a baseline. The classifier labeled instances as having Alzheimer's 13.9% of the time and not having Alzheimer's the remaining 86.1% of the time since 13.9% of the instances came from Alzheimer's patients. This resulted in classifying 76.4% of all instances correctly, with 13.2% of Alzheimer's patients being classified correctly and 86.6% of non-Alzheimer's patients being classified correctly.

n=4948	Alzheimer's (n=690)	Non-Alzheimer's (n=4258)	Total
Correctly Labeled	91	3689	3,780
Incorrectly Labeled	599	569	1,168
Accuracy	13.2%	86.6%	76.4%

Figure 1: Baseline Random Classifier Performance

Team Member Experiments

Each team member took 50 instances of the data and set it apart as a validation set and attempted to classify them based on their observations from the rest of the training set. This led to the following observations:

Daniel observed that, for the most part, the individuals without cognitive impairment tended to be young and scored more highly on the Mini Mental State Examination (MMSE). In addition, the patients with one or two APOE4 genes were more likely to have Alzheimer's. Because of this, he manually classified patients with 2 of the following 3 characteristics as having Alzheimer's: if they are over 70, if they have one or two APOE4 genes, if they have a score of less than or equal to 20 on the MMSE. With this method, he correctly classified 52% of instances correctly, with 41.2% of Alzheimer's patients being correctly classified as having Alzheimer's and 57.6% of non-Alzheimer's patients being correctly classified as not having Alzheimer's.

n=50	Alzheimer's (n=17)	Non-Alzheimer's (n=33)	Total

Correctly Labeled	7	19	26
Incorrectly Labeled	10	14	24
Accuracy	41.2%	57.6%	52%

Figure 2: Daniel's Manual Classification

When observing the training data, Krysten noticed that most of the patients with an Alzheimer's diagnosis owned 1 or more deleterious APOE alleles and had a hippocampus volume of 7000 or less. She found that other features, like age, race, and gender, were most times inconclusive, and this was confirmed when consulting the data statistics displayed by the boxplots. She manually classified the validation set for Alzheimer's based on these two factors. This was a different approach from Daniel's, but since he observed that MMSE scores were significantly lower in Alzheimer's patients, it suggests that if these scores were considered in this classification algorithm, it might increase it accuracy.

n=50	Alzheimer's (n=17)	Non-Alzheimer's (n=33)	Total
Correctly Labeled	9	23	34
Incorrectly Labeled	8	10	18
Accuracy	52.9%	69.7%	68%

Figure 3: Krysten's Manual Classification

Chandler recognized a deteriorated hippocampus in patients scoring under 27 on the MMSE with more than 1 rampant APOE4 alleles correlated strongly with Alzheimer's Disease being present. In the absence of a hippocampal volume measurement, a check of either rampant APOE4 or low MMSE score was performed. The following approximation of a classifier was used:

(Hippocampus < 6,000) &
(#Defective APOE4 > 1 || MMSE < 27)

This resulted in an 88.2% correct identification of Alzheimer's in the validation set and 90.9% accuracy in identifying non-Alzheimer's.

n=50	Alzheimer's	Non-Alzheimer's	Total
	(n=17)	(n=33)	
Correctly Labeled	15	30	45
Incorrectly Labeled	2	3	5
Accuracy	88.2%	90.9%	90%

Figure 4: Chandler's Manual Classification

"Expert System" Approach

Our third approximation for a classifier outperformed the baseline significantly. The classifier function looks something like this:

(Hippocampus < 6,000) AND (#Defective APOE4 > 1 OR MMSE < 27)

With a default of true given to the first condition if no measurement exists. We chose this classifier because it was approximately the model Chandler came up with to classify the instances by hand and his was the most successful of the 3 human classifiers.

n=50	Alzheimer's (n=17)	Non-Alzheimer's (n=33)	Total
Correctly Labeled	15	30	45
Incorrectly Labeled	2	3	5
Accuracy	88.2%	90.9%	90%
Baseline Accuracy	13.2%	86.6%	76.4%

Figure 5: Expert System Classification

The handwritten classifier drastically improves on the identification of Alzheimer's over the random baseline.

Supervised Learning - Initial Approaches

K Nearest Neighbors (KNN)

The first supervised learning approach we tried was the K Nearest Neighbors algorithm. To calibrate this algorithm, there were three key design choices. The first was the choice of k, the second was the distance function used, and the third was the way to compensate for missing features in the dataset. For the distance function, we used age, APOE4 genotype, ventricles volume, hippocampus volume, whole

brain volume, and MMSE score as features. We tried 2 types of distance functions: Euclidean distance and Manhattan distance. For each of these types of distance function, we used non-weighted and weighted versions, where features are weighted by dividing their values by the range of values present. To compensate for missing features in the dataset, we tried 2 approaches: assigning each missing value to the value of 0 and assigning it the average value of the feature. For all of these approaches, we tried to classify whether the patient has Alzheimer's or does not have Alzheimer's.

From here, we evaluated each set of design choices by using values of k from 1-20. To test this, we used leave-1-out cross validation and measured the total accuracy of each approach.

This yielded the following results:

K	Non-weighted Distance				Weighted Distance			
	Manhattan Distance		Euclidean Distance		Manhattan Distance		Euclidean Distance	
	0 for missing vals	Avg for missing vals	0 for missing vals	Avg for missing vals	0 for missing vals	Avg for missing vals	0 for missing vals	Avg for missing vals
1	79%	79%	78%	79%	77%	77%	77%	76%
2	80%	79%	79%	78%	76%	77%	78%	78%
3	85%	84%	85%	85%	81%	80%	85%	84%
4	84%	84%	86%	84%	80%	80%	84%	83%
5	88%	88%	89%	88%	83%	82%	87%	88%
6	87%	86%	88%	87%	83%	82%	87%	87%
7	90%	90%	88%	89%	83%	83%	90%	91%
8	89%	90%	88%	88%	83%	82%	89%	90%
9	93%	92%	92%	92%	84%	83%	93%	93%
10	92%	92%	92%	91%	85%	84%	92%	93%
11	95%	95%	94%	94%	85%	84%	93%	95%
12	94%	95%	93%	93%	85%	85%	93%	95%
13	97%	94%	97%	97%	86%	85%	95%	96%
14	96%	96%	97%	96%	87%	84%	94%	96%
15	97%	98%	99%	99%	86%	85%	95%	97%
16	98%	98%	98%	99%	87%	84%	95%	96%
17	98%	99%	99%	100%	87%	85%	96%	97%
18	98%	99%	99%	99%	87%	86%	96%	97%
19	99%	100%	100%	100%	87%	86%	97%	98%
20	99%	100%	99%	99%	87%	86%	97%	98%

Figure 6: Percent of Instances Correctly Classified using KNN with Leave-1-Out Validation (Rounded to nearest percent)

Using these results, we were able to see that in general, using a non-weighted distance function produced better results than using a weighted distance function, though these results were somewhat mitigated by using Euclidean distance instead of Manhattan distance. This indicates that the values that naturally have a larger range of values may be more predictive of whether a patient has Alzheimer's. This seems to make sense, since the values with the largest range are ventricle weight, hippocampus weight, and total brain weight, the raw signals received from an MRI. To test this hypothesis, we tried eliminating the other features - age, APOE4 genes, and MMSE score - from the distance calculation. This greatly improved the results using the weighted distance, resulting in the following accuracy measures:

	Manhattan Distance		Euclidean Distance	
K	0 for missing vals	Avg for missing vals	0 for missing vals	Avg for missing vals
1	78%	77%	77%	76
2	77%	75%	78%	78%
3	83%	83%	85%	84%
4	82%	80%	84%	83%
5	85%	86%	87%	88%
6	85%	85%	87%	87%
7	88%	89%	90%	91%
8	87%	88%	89%	90%
9	90%	90%	93%	93%
10	89%	90%	92%	93%
11	93%	90%	93%	95%
12	92%	91%	93%	95%
13	93%	92%	95%	96%
14	92%	92%	94%	96%
15	93%	93%	95%	97%
16	93%	93%	95%	96%
17	93%	94%	96%	97%

18	93%	94%	96%	97%
19	94%	94%	97%	98%
20	93%	94%	97%	98%

Figure 7: Percent of Instances Correctly Classified using KNN with Leave-1-Out Validation, ignoring age, APOE4, and MSSE (Rounded to nearest percent)

As anticipated, we saw an increase in accuracy of weighted distance when ignoring these features, particularly when using Manhattan distance. Despite this, the approach still does not seem to be as accurate as using the unweighted distance function, indicating that these features may have some predictive value, and at the very least don't seem to be having a strong negative effect when they are not weighted highly.

To verify this, we ran KNN using a non-weighted Manhattan distance function while replacing missing values with the average for that feature. We chose this set of parameters because it produced the highest accuracies initially. We ran it both with age, APOE4, and MMSE included and not included. In addition, we tried k values of up to 25 to determine the best possible k value. This produced the following results:

K	Including Age, APOE4, and MMSE	Not Including Age, APOE4, and MMSE
1	79.42%	79.88%
2	79.27%	79.57%
3	84.15%	84.45%
4	84.30%	84.76%
5	87.50%	87.50%
6	86.13%	86.43%
7	90.24%	90.24%
8	89.63%	89.79%
9	92.38%	92.38%
10	92.23%	92.38%
11	94.82%	94.82%
12	93.60%	93.75%
13	96.34%	96.34%
14	95.88%	96.04%
15	98.17%	98.32%
16	97.87%	97.87%
17	99.09%	99.09%
18	98.93%	98.93%
19	99.70%	99.70%
20	99.70%	99.70%
21	99.85%	99.85%
22	99.54%	99.54%
23	99.70%	99.70%
24	99.54%	99.39%
25	99.54%	99.54%

Figure 8: Percent of Instances Correctly Classified using KNN with Leave-1-Out Validation, an unweighted Manhattan distance function, and replacing missing values with the average for that feature

In general, the version that didn't include age, APOE4, and MMSE information was slightly more accurate. Because of this, for our final KNN algorithm, we will use the unweighted Manhattan Distance function while replacing missing values with the average for that feature and ignoring age, APOE4, and MMSE. To select the ideal value of K, we will examine performance of the algorithm using various values of K.

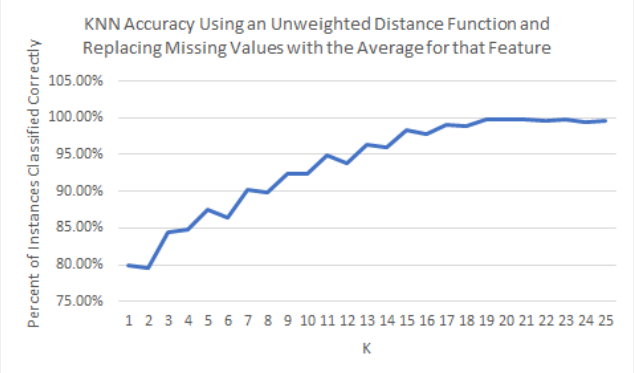


Figure 9: Percent of Instances Correctly Classified using KNN with Leave-1-Out Validation, an unweighted Manhattan distance function, replacing missing values with the average for that feature, and ignoring age, APOE4, and MMSE score

As can be clearly seen, the performance of the algorithm steadily rises as k rises until it flattens around once k reaches 20. Because of this, we will use a value of 20 for k, which produces the following results:

n=656	Alzheimer's (n=153)	Non-Alzheimer's (n=503)	Total
Correctly Labeled	152	502	654
Incorrectly Labeled	1	1	2
Accuracy	99.35%	99.80%	99.70%

Figure 10: KNN Classification

This algorithm performs significantly better than the baseline, predicting instances with Alzheimer's 86.15% more accurately and predicting instances without Alzheimer's 13.2% more accurately. In addition, it performs significantly better than human classifiers, classifying instances with Alzheimer's 11.15% more accurately and predicting instances without Alzheimer's 8.9% more accurately than the best human classifier, Chandler. In addition, using leave-1-out validation ensures that there is not significant variability using different validation sets because it allows the algorithm to train on essentially the entire dataset.

Basic, Fully-Connected Neural Network

Next, we built a neural network to classify the instances in our dataset as either Control, LMCI (cognitive impairment), or Alzheimer's. The neural network was built in R using the "neuralnet" package. After delineating our dataset of instances with missing values for hippocampus volume, we trained our network on a training set of 475 instances, and tested the network on two cross-validation sets of 53 instances. We chose to remove instances with missing values for simplicity rather than trying to substitute in values which could have a large impact on the weights. After viewing the summary statistics of our dataset, we decided to use hippocampus volume, race, gender, MMSE score, APOE genotype, and marital status as our features for classifying a patient's diagnosis. If features were continuous, they were recoded as categorical, and this categorization was guided by the summary statistics.

In the three experiments ran, we modified the following parameters: threshold, number of hidden layers, and the learning rate, while leaving the stepmax (1e+06) and number of epochs (30) constant across all experiments. The threshold parameter is used as a threshold for the partial derivative of the error function before it converges, and the stepmax is the maximum number of iterations the algorithm will complete before it converges. The final classifications were determined by the network's predicted value for the diagnosis at baseline (DX.bl). If the diagnosis was 0-0.5, the result was labeled as Control. If it was 0.5-1.5, the result was labeled as LMCI, and finally any value over 1.5 was labeled as Alzheimer's. The accuracy of our experiments is described below.

Experiment 1: Change in Learning Rate, 30 epochs, 3 hidden layers, 10 neurons in each layer,1e+06 stepmax, 0.01 threshold

Accuracy	rate= 10	rate = 30	rate= 50
Validation Set #1	~91%	~91%	~91%
Validation Set #2	~93%	~94%	~94%

Experiment 2: Change in #of Hidden Layers, 30 epochs, rate= 30, 10 neurons in each layer, 1e+06 stepmax, 0.01 threshold

Accuracy	1 layer	5 layers	10 layers
Validation Set #1	~94%	~95%	~95%
Validation Set #2	~91%	~91%	~91%

Experiment 3: Change in Threshold Value, 30 epochs, rate= 30, 2 hidden layers, 10 neurons in each layer, 1e+06 stepmax

Accuracy	0.001	0.01	0.5
Validation Set #1	~91%	~91%	~79%
Validation Set #2	~94%	~94%	~74%

The results of these experiment were quite surprising. First, we expected that varying the learning rate would have a much more significant effect on results, especially since the number of epochs were held constant

in that experiment. We expected to see one of two results: for the accuracy at a learning rate of 50 to be highest or for it to be the lowest. If it proved to be highest at 50, then that would mean that 30 epochs was an appropriate number of epochs needed to train the network. Conversely, if the accuracy decreased with a learning rate of 50, then that would mean that the network trained too quickly and errors resulted from overfitting. However, we observed neither trend, and we are quite skeptical of these results. Additionally, the number of hidden layers used in the network also did not have an effect on the accuracy in either of the validation sets. Since there were a small number of features used in this network and instances were categorical, we knew that increasing the layers would not produce substantially greater results. Still, we expected some improvement when using five or ten layers, since that would mean the network would have greater coverage for the features and also outputs would be more precise for the final output layer. However, one layer produced a high accuracy, and the results remained constant. Lastly, we found that increasing the threshold value had a great impact on the network's accuracy. This is what was expected, since a higher threshold would mean stricter constraints in the error function for the classification Alzheimer's.

Our network had great accuracy in all validation sets tested. The highest average accuracy across the validation sets was achieved in when both 5 and 10 layers were used in the network (Experiment 2). We decided to use 5 layers since this produced the best results against the validation set and it is less likely to fall into overfitting since there are fewer layers. In addition, since it was tested against a validation set it was blind to, clearly overfitting isn't affecting it too much. A matrix describing the results of the second trial of this experiment (using 5 hidden layers) are below.

n=106	AD (n=23)	LMCI (n=37)	Control (n=46)
Correctly Labeled	21	35	42
Incorrectly Labeled	2	2	4
Accuracy	91.3%	94.6%	91.3%

Validation set #2's accuracy was only slightly higher than #1's in this experiment. Although there was a fairly equal distribution of characteristics across the initial training set, the instances in validation set #2 may have had features that produced output that more closely fit our categorization cutoffs for classifying patients. In comparison to our baseline algorithm for classifying patients, this network approach proved to be much more sophisticated. Its accuracy for predicting Alzheimer's dramatically improved from 13% to 91%, and it also classified patients in categories as LMCI or Control with accuracies > 91% for both, which was not done in the baseline characterization. In addition, this experiment uses more features and proved to have higher accuracy for predicting both Alzheimer's and Non-Alzheimer's than Chandler's manual experiments in Update 1, which used just MMSE score, APOE genotype, and Hippocampus volume.

Although we observed promising results for this neural network, we are still skeptical about the results, as they did not reflect that increasing the complexity network had an effect on its accuracy (i.e. changing training rate from 10 to 30). In the future, we plan to rebuild this network to include continuous variable and to rethink the categorization of our features.

Decision Tree

The next supervised learning approach we tried was the Decision Tree algorithm. We used an ID3 implementation. To calibrate this algorithm, there were three key design choices. The first was the choice of how to discretize the data, the second was the maximum depth of the tree, and the third was whether or not to prune the tree generated during the training step. To discretize the data, we decided to split each feature into bins of equal size. We tried several different numbers of bins, ranging from 2 bins per feature to 5 bins per feature (we stopped here since it seemed to be getting worse as we added bins). To choose the maximum depth, we simply tried a number of different depths. Since we chose to only use 8 features (age, gender, race, apoe4, ventricles, hippocampus, whole brain, mmse, and diagnosis) to construct the decision tree, we looked at depths from 3 to 8. Finally, we tried all of these options both with pruning and without pruning. We chose to use k-fold validation with a value of 5 for k. For all of these approaches, we tried to classify whether the patient has Alzheimer's or does not have Alzheimer's.
This yielded the following results without pruning:

K	2 bins per feature	3 bins per feature	4 bins per feature	5 bins per feature
3	75.30%	75.76%	74.24%	74.70%
4	75.15%	73.93%	72.10%	73.32%
5	75.91%	71.19%	69.67%	68.90%
6	73.48%	71.65%	67.23%	64.48%
7	72.87%	69.21%	64.33%	59.15%
8	78.81%	69.82%	61.74%	56.86%

Figure 11: Percent of Instances Correctly Classified using Decision Trees without Pruning

It also yielded the following results with pruning:

K	2 bins per feature	3 bins per feature	4 bins per feature	5 bins per feature
3	76.68%	76.52%	76.52%	74.70%
4	75.15%	74.24%	73.63%	72.71%
5	74.85%	72.41%	71.49%	71.19%
6	75.00%	71.12%	67.07%	66.62%
7	72.26%	70.43%	66.92%	62.80%
8	72.26%	67.99%	62.65%	58.69%

Figure 12: Percent of Instances Correctly Classified using Decision Trees with Pruning

Collectively, the data generally got worse as we added more depth and more bins per feature, and seemed to do better with pruning. This trend generally held true, with a depth of 2 and 2 bins per feature with pruning producing the best results, with one notable exception: creating a tree of depth 8 with 2 bins per feature and no pruning actually produced the best results. To determine the best possible decision tree, therefore, we narrowed our focus to these 2 approaches. To do so, we used a different train-validation split to see if both approaches still performed well. Upon further inspection, each produced the following results across a 10-fold validation:

Validation sets	No pruning, depth 8, 2 bins per feature	Pruning, depth 3, 2 bins per feature
Set 1	77.82%	74.43%
Set 2	74.26%	70.59%
Set 3	77.11%	73.63%
Set 4	69.01%	64.79%
Set 5	79.76%	77.34%
Set 6	79.30%	77.78%
Set 7	79.61%	75.70%
Set 8	80.42%	77.38%
Set 9	80.37%	76.99%
Set 10	80.34%	76.68%
Total	77.80%	74.53%

Because the version without pruning still performed better, we decided to go with this version. Interestingly, there was relatively little variance between the performance on the various validation sets, indicating that the algorithm doesn't seem to be overfitting too much and is able to learn all of the data fairly well. Furthermore, on further inspection it became clear that the pruned tree was actually just classifying all patients as not having Alzheimer's, and was not providing useful information. Because of these factors, we decided to use the unpruned tree of depth 8 with 2 bins per feature. Finally, we examined the breakdown of the data that this produced:

n=656	Alzheimer's (n=153)	Non-Alzheimer's (n=503)	Total
Correctly Labeled	80	447	527
Incorrectly Labeled	73	56	129
Accuracy	52.29%	88.87%	80.34%

Figure 10: Decision Tree Classification

This was significantly outperformed by the expert system classifier, which performed 2% better on non-alzheimer's patients and 30% better on Alzheimer's patients. It does

worse than the human classifier of Chandler by the same margin, but it does at least do better than the baseline classifier by 2% on non-alzheimer's patients and 39% on Alzheimer's patients

Random Forest

The final supervised learning approach we tried was a forest of randomized trees. To calibrate this algorithm, there were three key design choices: the maximum number of features to be considered when looking at each split, the maximum depth of the trees, and the number of trees in the forest. Since we are using 8 features, we initially tried values of 2, 4, and 6 for the maximum number of features to be considered at each split. Likewise, we considered depths of 2, 4, and 6 initially. Lastly, we initially tried values of 5, 10, and 50 for the number of trees in the forest. We chose to use k-fold validation with a value of 5 for k. For all of these approaches, we tried to classify whether the patient has Alzheimer's or does not have Alzheimer's.

This yielded the following results when run against the validation sets (results per validation set are averaged):

Max Depth	# Trees	Max number of features/split		
		2	4	6
2	5	85.21%	87.80%	87.65%
	10	85.98%	87.96%	88.41%
	50	86.59%	88.11%	88.41%
4	5	87.35%	86.59%	87.20%
	10	87.50%	89.33%	86.89%
	50	88.57%	88.27%	88.41%
6	5	86.28%	85.06%	86.13%
	10	86.89%	85.98%	86.89%
	50	88.11%	87.96%	86.59%

Figure 11: Classification using Random Forest

Unfortunately, there was not a clear correlation between the data and the hyperparameters. Because there was no clear correlation here, we decided to use the set of hyperparameters that produced the best results: a maximum of 4 features per split, with a maximum depth of 4 and 10 trees. This ultimately resulted in a classifier that when run again using leave-1-out validation resulted in 87.96% accuracy with the following confusion matrix.

n=656	Alzheimer's (n=153)	Non-Alzheimer's (n=503)	Total
Correctly Labeled	88	489	577
Incorrectly Labeled	65	14	79
Accuracy	57.52%	97.22%	87.96%

Figure 12: Classification using Random Forest with a maximum depth of 4, 10 trees, and a maximum of 4 features per split

This algorithm performs significantly better than the baseline, predicting instances with Alzheimer's 44.32% more accurately and predicting instances without Alzheimer's 10.62% more accurately, for a total performance that is 11.56% more accurate. In addition, it performs similarly to human classifiers, classifying instances with Alzheimer's 31.00% less accurately, but predicting instances without Alzheimer's 6.32% more accurately than the best human classifier, Chandler, for a total that is 2.02% worse than the human. It is also worse than our expert machine classifier by the same margin.

Results

Supervised Learning on Test Set

Finally, we ran our classifiers against the test set that we had initially set aside and had not used during training. Run against the test set, the K-Nearest-Neighbors classifier produced an accuracy of 99.39% while producing the following confusion matrix:

n=164	Alzheimer's (n=39)	Non-Alzheimer's (n=125)	Total
Correctly Labeled	39	124	163
Incorrectly Labeled	0	1	1

Accuracy	100%	99.20%	99.39%

Figure 13: KNN Classification of Test Set

This produced approximately the same accuracy as running the KNN using leave-one-out validation. This indicates that the algorithm wasn't overfitting and was exploiting actual patterns in the data. The success of KNN most likely points to the brain weight features being highly predictive of Alzheimer's given that they tended to control the distances since an unweighted distance was used and they had the largest variability.

Next, we tested the Neural Network against the test set using the parameters that produced the best results in our validation sets: 5 hidden layers, 30 epochs, rate= 30, 10 neurons in each layer, 1e+06 stepmax, 0.01 threshold. Additionally, we used the same features that the neural network was originally trained on: race, gender, hippocampus volume, age, MMSE, and marital status to predict a Non-Alzheimer's or Alzheimer's diagnosis. If the features were continuous, they were re-coded as categorical, and this process was guided by the summary statistics. The diagnosis for each patient was also relabeled. Control patients and LMCI patients were labeled as 0 and 1, respectively, and AD patients were labeled as 2. If the network produced an output for 0-1.5 for the predictive variable (diagnosis), this output was classified as Non-Alzheimer's, and any output over 1.5 was classified as Alzheimer's.

After removing missing values in the dataset, there were 132 instances. When run against the test set, the Neural Network classifier produced the accuracies as described in the following confusion matrix:

n=132	Alzheimer's (n=30)	Non-Alzheimer's (n=102)	Total
Correctly Labeled	25	96	121
Incorrectly Labeled	5	6	11
Accuracy	83.33%	94.12%	91.67%

Figure 14: Neural Network Classification of Test Set

In comparison to the validation set, when run on the test set, the neural network's accuracy decreased from 91% to 83% for predicting Alzheimer's, and it only rose from 93% to 94% for predicting Non-Alzheimer's patients. When the test set was run again using other parameters (learning rate = 50, 3 hidden layers), its accuracy was 75%, which is much lower than the accuracy observed using the validation set under these parameters. Thus, this leads us to believe that the accuracy may have decreased because neural network may have overfitted in favor of the training set. Perhaps, training the network on an increased number of instances in order to diversify the dataset, using more features, and re-evaluating the cutoffs for AD and non-AD classification would lead to a higher accuracy.

Third, when run against the test set, the Decision Tree classifier produced an accuracy of 69.51% while producing the following confusion matrix:

n=164	Alzheimer's (n=39)	Non-Alzheimer's (n=125)	Total
Correctly Labeled	0	114	114
Incorrectly Labeled	39	11	50
Accuracy	0.00%	91.20%	69.51%

Figure 15: Decision Tree Classification of Test Set

This produced significantly worse results than running the classifier against validation sets. This is almost certainly due to overfitting, with the 0% accuracy in classifying Alzheimer's patients demonstrating that the classifier wasn't actually learning much beyond the fact that most of the data corresponds to non-Alzheimer's patients. Overfitting was clearly a concern, with a tree depth of 8, but was initially ignored since the alternative best approach was to have a tree of depth 2 that classified all instances as not having Alzheimer's.

Finally, run against the test set, the Random Forest classifier produced an accuracy of 89.63% while producing the following confusion matrix:

n=164	Alzheimer's (n=39)	Non-Alzheimer's (n=125)	Total
Correctly Labeled	25	122	147
Incorrectly Labeled	14	3	17
Accuracy	64.10%	97.60%	89.63%

Figure 16: Random Forest Classification of Test Set

This produced very similar, and actually slightly better, results than running the classifier against the

training set using leave-1-out validation. Unlike the simple decision tree, overfitting clearly wasn't an issue here, which makes sense as that is one of the purposes of randomness in the random forest design. It is still concerning, however, that it was only able to classify 64.10% of patients with Alzheimer's correctly

Analysis

Overall, the classifiers had mixed success classifying the data, producing the following aggregated results when run against the test set:

Classifier	Percent Alzheimer's Classified Correctly	Percent non-Alzheimer's Classified Correctly	Percent Total Classified Correctly
KNN	100%	99.20%	99.39%
Neural Net	83.33%	94.12%	91.67%
Decision Tree	0%	91.20%	69.51%
Random Forest	64.10%	97.60%	89.63%

Figure 17: Classifier Accuracies on Test Set

Unsurprisingly,the more sophisticated algorithms generally performed better than the decision tree algorithm. This is most likely due to a combination of the aforementioned overfitting, which is always a danger with decision trees of large depth, as well as simply having a better suited search space for this data. In particular, it was challenging to discretize the data in a meaningful way that separated the data into the correct categories but didn't chunk it up so much to fall into overfitting.

Also of interest is the fact that KNN clustering did so well, performing significantly better than all the other algorithms that were tried. This indicates that while it may be challenging to find clear functions that divide the data, the data tended to be clustered together by label. In addition, because we used an unweighted distance function for this, it weighted the brain weights significantly more than other features, indicating that these features may be more predictive.

Because KNN performed so well, there is not a ton of room for improvement in our results (it incorrectly classified only 1 test instance out of 164 and only 2 training instances out of 656). Clearly the model is exploiting most patterns present in the data. In order to move closer to 100% accuracy, the algorithm would probably simply need more data - this would help the algorithm recognize potential outliers or patterns not present in the data. It may be, however, that even though the model incorrectly classified a few points, that was actually a smart thing to do in the general case when similar points appear, so it is unclear if adding data would actually improve accuracy.

For the other algorithms, more data would certainly help as well, especially with the decision tree. Provided enough data, the decision tree's overfitting problem would most likely go away because the trends that it is currently incorrectly finding in the data would be balanced out by data that didn't follow those trends. It may be, however, that no matter how much data is present a decision tree approach would fail to succeed without a better way of discretizing the data.

Improving the random forest model would also most likely be easiest done by adding more data. Right now, the random forest is doing significantly better than a single decision tree, but each tree is probably facing some of the same challenges: an inability to discern between actual trends in the data and those that seem to exist due to a lack of data. Adding data would solve those issues.

Improving the Neural Net would most likely take a combination of more data and more computing power. This would allow the Neural Net to learn more about the dataset, but would also allow it to run more epochs which generally correlated to better results, though those benefits may dwindle as the algorithm reaches a local minimum.

Authors Contributions

All 3 authors collaborated on the introduction. Daniel created the baseline classifier. All 3 authors performed human experiments and Chandler created the expert system from that data.

Daniel wrote the KNN, decision tree, and random forest classifiers and sections. Krysten wrote the neural net classifier and section. Daniel wrote the analysis section, though Krysten provided final data from the neural net classifier.

Reference(s)

ADNI, University of Southern California, adni.loni.usc.edu/

NeuralNet, Stefan Fritsch [aut], Frauke Guenther [aut, cre], Marc Suling [ctb], Sebastian M. Mueller [ctb], https://cran.r-project.org/web/packages/neuralnet/neuralnet.pdf

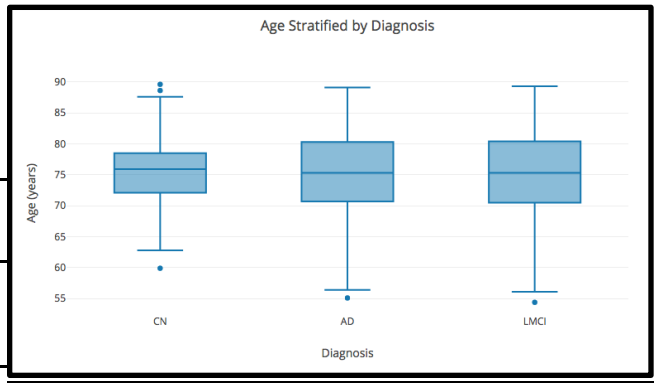
Patel, Savan. "Chapter 2 : SVM (Support Vector Machine) - Theory – Machine Learning 101 – Medium." *Medium*, Machine Learning 101, 3 May 2017, medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72.

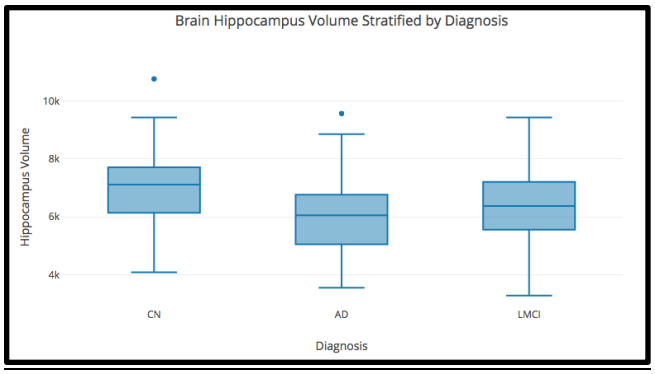
Appendix A

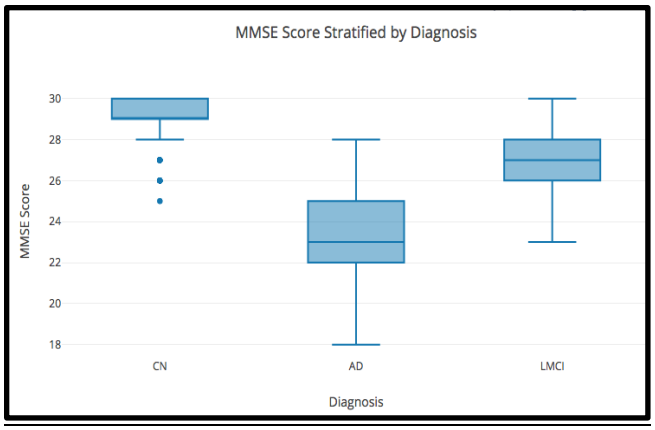
ADNI Dataset Summary Statistics:

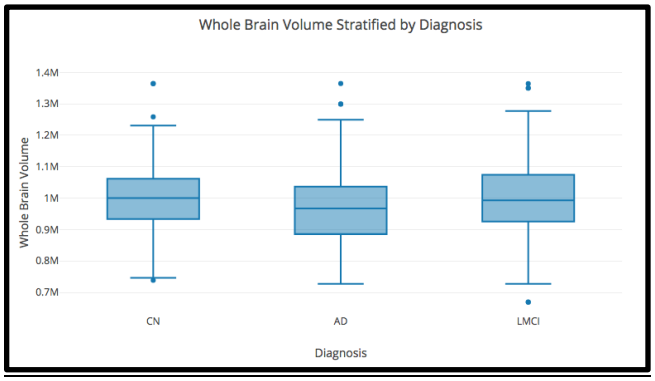
	Alzheimer's (AD)
Total=819	*N=192*
Gender **Male** = 478 (~58%) **Female**= 341 (~42%)	*Male=101* *Female=91*
Race **American Indian/Alaskan**=1 (0.12%) **Asian**=14 (1.7%) **Black**=39 (~4.8) **More than one** = 3 (0.4%) **White** = 763 (~93.2%)	*American Indian/Alaskan=0* *Asian=2* *Black=8* *More than one = 2* *White = 180*

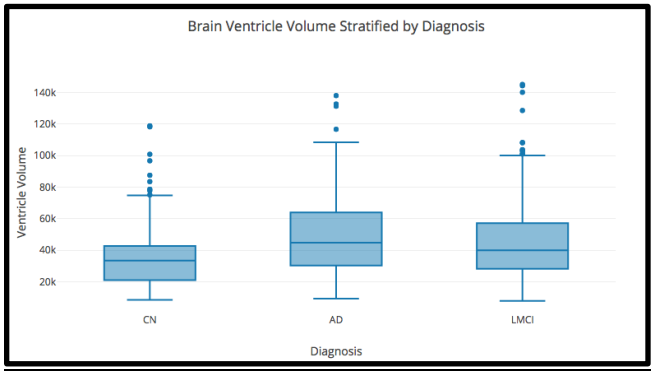
Cognitive Impairment (LMCI)	**Control (CN)**
N=398	*N=230*
Male=257 *Female=141*	*Male=120* *Female=109*
American Indian/Alaskan=1 *Asian=9* *Black=15* *More than one = 1* *White = 372*	*American Indian/Alaskan=0* *Asian=3* *Black=16* *More than one = 0* *White = 211*

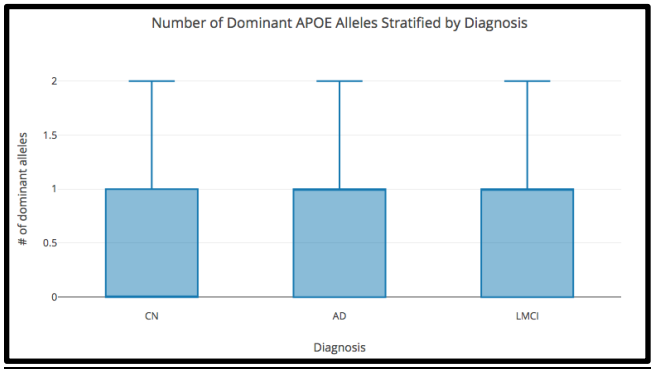



Brain Hippocampus Volume Stratified by Diagnosis
Hippocampus Volume
10k
8k
6k
4k
CN
AD
LMCI
Diagnosis


MMSE Score Stratified by Diagnosis
MMSE Score
30
28
26
24
22
20
18
CN
AD
LMCI
Diagnosis


Whole Brain Volume Stratified by Diagnosis
Whole Brain Volume
1.4M
1.3M
1.2M
1.1M
1M
0.9M
0.8M
0.7M
CN
AD
LMCI
Diagnosis


Brain Ventricle Volume Stratified by Diagnosis
Ventricle Volume
140k
120k
100k
80k
60k
40k
20k
CN
AD
LMCI
Diagnosis


Number of Dominant APOE Alleles Stratified by Diagnosis
# of dominant alleles
2
1.5
1
0.5
0
CN
AD
LMCI
Diagnosis