Project Proposal Title: Alzhiemer's disease prediction using ADNI neuroimaging data

Group members: Charlee Cobb

Topic and Research Question: While single classifiers have been the preferred method of machine learning for neuroimaging machine learning tests, they have fallen out of favor to adapt to the complexity and nuances of diagnostic neuroimaging (Dimitriadis, 2018). During the kaggle competition “A Machine learning neuroimaging challenge for automated diagnosis of Mild Cognitive Impairment”, the winning team discovered that ensemble algorithms were more effective at using neuroimaging data to diagnose and to differentiate Alzhiemer’s disease from Mild Cognitive Impairment. The goal of this project is to use Random Forest modeling on the neuroimaging data and compare the results to SVM modeling.

Hypothesis: Random Forest model performs better than SVM for classifying Mild Cognitive Impairment and Alzheimer’s disease in a patient.

Data sources: https://www.kaggle.com/competitions/mci-prediction/data

Methods: Use the given test and train data sets on Random Forest models and non linear kernel SVM models. Provided by ScKitlearn in python. For the Random Forest model, I will limport the classifier from scikit-learn, and set the criterion parameters to ‘gini’. On one classifier object, I will set the n_estimators to 50 and random_state to 1. On another classifier object, I will increase the n_estimators to 100, max_depth to 10, and random_state to 1. For the SVM, I will set the SVC object parameters as follows: kernel = ‘rbf’, gamma=0.10, C=10.0, random_state = 1.

Expected Results: I expect the Random Forest models to perform better at diagnosing Mild Cognitive Impairment due to the fact that it’s an ensemble algorithm. I expect that using a low number of trees will be efficient for prediction.

Potential Problems and Solutions:

The provided dataset is high-dimensional, which is a strength for Random Forest algorithms. However, this is a relatively large dataset, so it may take too long to make accurate predictions after training. To accommodate for this, I plan on adjusting the *k* number of trees. As mentioned in the ‘Python Machine Learning’, the larger the *k* value, the better the performance of the prediction will be at the cost of time. I plan on using two different k values to test whether a lower *k* still outperforms the SVM model.

Random Forest also tends to overfit the training data, especially in comparison to a linear model. While Random Forest has a limited number of parameters to adjust in comparison to the SVM model, I can reduce tree depth to help limit the amount of overfitting. However, the size of the dataset may be large enough to limit overfitting while training.