San Jose State University

SJSU ScholarWorks

Master's Projects

Master's Theses and Graduate Research

Spring 5-21-2020

Pattern Analysis and Prediction of Mild Cognitive Impairment Using the Conn Toolbox

Meenakshi Anbukkarasu

San Jose State University

Pattern Analysis and Prediction of Mild Cognitive Impairment Using the Conn Toolbox

A Master's Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

Of the Requirements for the

Degree Master of Science

By

Meenakshi Anbukkarasu

May 2020

ACKNOWLEDGMENTS

I want to offer my sincerest thanks to Dr. Leonard Wesley for his appropriate guidance and encouragement throughout this project. I believe myself to be incredibly lucky to have had a chance to collaborate with somebody as experienced as him.

I am also thankful to my committee members Dr. Robert Chun and Mr. Saurabh Shirur for giving their significant guidance and advices.

At last, I thank my wonderful husband, parents, and friends for their endless support and encouragement throughout this journey.

ABSTRACT

Alzheimer's is an irreversible neurodegenerative disorder described by dynamic psychological and memory defalcation. It has been accounted for that the pervasiveness of Alzheimer's is to increase by 4 times in a few years, where one in every 75 people will have this disorder. Hence, there is a critical requirement for the analysis of Alzheimer's at its beginning stage to diminish the difficulty of the overall medical complications. The initial state of Alzheimer’s is called Mild cognitive impairment (MCI), and hence it is a decent target for premature diagnosis and treatment of Alzheimer's. This project focuses on coordinating numerous imaging modalities to identify people in danger for MCI. The current advancement of brain network connectivity analysis has led to the identification of neurological issues at an entire connectivity level, thereby providing a new road to the classification of brain-related diseases. Utilizing neuroimage pattern classification and various machine learning techniques, we endeavor to incorporate information from CONN toolbox and resting-state functional magnetic resonance imaging (rs-fMRI) for refining MCI prediction accuracy.

Index Terms - Mild Cognitive Impairment, Rest State Magnetic Resonance Imaging.

Table of Contents

INTRODUCTION

Different imaging modalities give fundamental integral information that can be utilized to improve our comprehension of brain diseases. Alzheimer's is a dynamic sickness that is described by the decrease in cognitive operations and memory. Because of the change in structural and functional networks of the brain in people with Alzheimer's, it has been labeled as a disconnection disease (Bokde et al., 2006).

MCI refers to a medial phase between cognitive descent and Alzheimer's (Petersen, 2000b, p. 93). The predominance of MCI is roughly 15% in people above 65 years and the greater part of patients with MCI advance to dementia in less than 6 years (Farlow, 2009, p. 362). Prior diagnosis of Alzheimer's is significant in people with MCI, who show a change of pace of about 15% every year. Diagnosing patients at a beginning period of Alzheimer's can alter its course by utilizing appropriate treatment and medicines.

This project focuses on incorporating various imaging modalities to identify people in danger of MCI. It is often a beginning period of Alzheimer's infection, is hard to analyze because of its insignificant or mild indications of cognitive impairment. The ongoing rise of brain connectivity graph examination has made the classification of the neurological issue at an entire brain connectivity level conceivable, in this manner giving new roads to mind sicknesses classification.

BACKGROUND

It has been demonstrated that Alzheimer's physiology can be distinguished by utilizing neuroimaging methods. An assortment of approaches has been proposed to determine MRI biomarkers for primitive characterization of Alzheimer's (Misra et al., 2009, p. 1415). In a recent report, Eskildsen et al. (2013) utilized patterns of cortical thickness and found cortical areas possibly discriminative for isolating MCI patients from Healthy Control (HC) patients. They showed promising outcomes for the identification of patients with prodromal Alzheimer's advancing to plausible Alzheimer's. Despite the utilization and effectiveness of MRI biomarkers in predicting MCI from HC have been exhibited in past researches, there is no investigation, as far as we could know, to research the utilization of rs-fMRI for characterizing the two classes.

The rs-fMRI has developed as an efficient and powerful instrument for surveying and mapping the functional design of the brain network. Until this point, only a few examinations have utilized rs-fMRI to identify Alzheimer's affected brain network (Khazaee et al., 2015, p. 2132). Identifying dependable biomarkers for separating MCI-C from HC patients is critical for primitive identification and treatment of Alzheimer's. The principle point of this project is to create and assess a method dependent on rs-fMRI and predict the presence of MCI.

When compared to normal cognition people, the patients with Alzheimer's have significantly less regional connectivity and exhibited disordered global functional organization (Supekar et al., 2008). Also, cognitive descent in patients with Alzheimer's is related to distorted functional connectivity in the whole brain. The graph-based approach has been demonstrated to be a powerful methodology in detecting modification of the brain network in mental and neurological disorders (Armstrong et al., 2016, p. 175). It has likewise been indicated that the blend of the graph theory with an AI approach based on rs-fMRI can precisely classify patients with mild cognitive impairment, and healthy patients (Khazaee et al., 2015, p. 2132).

This project creates and assesses a unique technique dependent on the graph theory, an Artificial Intelligence-based approach, and rs-fMRI data produced by utilizing the Conn Toolbox (www.nitrc.org/projects/conn) to precisely classify MCI people from HC patients. The most challenging part of this project is using the rest state fMRI image to predict an MCI patient. All the past investigations have used the basic MRI for classification of MCI patients from HC patients, but this is the first examination that will use rest state-fMRI for this classification. Additionally, the utilization and effectiveness of MRI biomarkers in identifying MCI from healthy patients have been resolved in past investigations. But there is no examination apparently to research the utilization of rs-fMRI for recognizing these two categories

METHOD AND APPROACH

DATA COLLECTION

The initial plan was to collect fMRI from real-time Dementia patients from Palo Alto Medical Center but due to legal issues, it took a long time to process our application. Hence, we did not get the desired data in time. The alternative plan is to use the data from Harvard. The fMRI resting-state data that is examined in this project is from the Resting-state fMRI in Dementia Patients dataset (Mascali et al., 2015, Harvard Data verse). The data were obtained from ten patients belonging to three different categories namely

Alzheimer's (AD) (Mckhann et al., 1984, p. 939), Mild cognitive impairment (Petersen et al., 2001, p. 1985) and 10 healthy control elders (HC).

The patients went through a resting state echo-planar imaging fMRI scanning process. The duration of the scan was 7 minutes and 20 seconds, producing 220 volumes. Patients were told to keep their eyes shut all through, withheld from envisioning anything specific and to abstain from sleeping. An anatomical scan was additionally procured for each subject. Every patient's information is labeled into Structural and Functional data to perform connectivity examinations on Region of Interest (ROI) utilizing the CONN Toolbox.

CONN TOOLBOX

CONN is a MATLAB-based software for the calculation, demonstration, and investigation of brain network connectivity using fMRI. Connectivity assessment technique includes seed-to-voxel network maps, ROI-to-ROI connectivity matrix, graph characteristics of network systems, brain interconnection, inherent connectivity, generalized psychophysiological interaction models and other voxel-to-voxel measures.

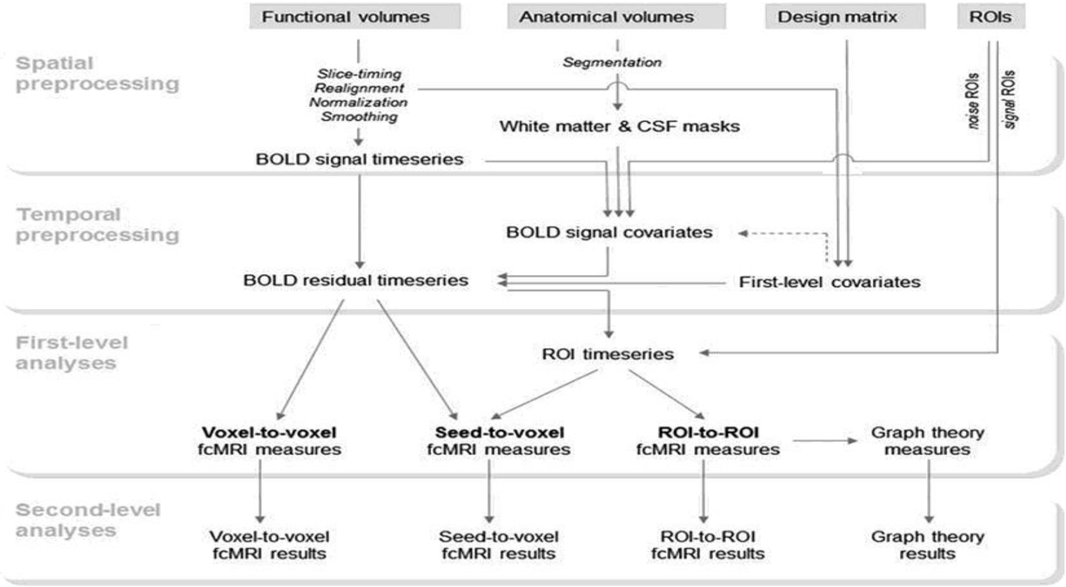
It is available for rs-fMRI and task-related plans. It covers the whole pipeline starting from basic fMRI information to proposition testing, including spatial co-registration, scrubbing, aCompCor technique for management of physiological and movement discomfit, first-level connectivity assessment, and second level arbitrary effect examinations. So, to do connectivity analyses utilizing CONN you will require the following **functional and structural data.** Functional data is either a resting state or a task plan that can be examined. Whereas, structural data is defined as at least one anatomical volume for each patient (this is utilized generally for plotting graphs in addition to determining the white/CSF/gray masks utilized in the aCompCor confound elimination technique)

DATA PROCESSING

The functional and structural data of both MCI and HC patients should be preprocessed and examined utilizing the CONN tool to produce a connectivity matrix that gives data about the association between ROIs present in a specific region of the brain. The preprocessing and examination of MCI and HC patients are done independently. The process for computing fMRI measures involves five steps namely setup, preprocessing, denoising, analysis, and results exploration. The steps associated with the calculation of fMRI measures are described in Fig.1.

Figure 1

Analysis steps involved in the calculation of fMRI measures

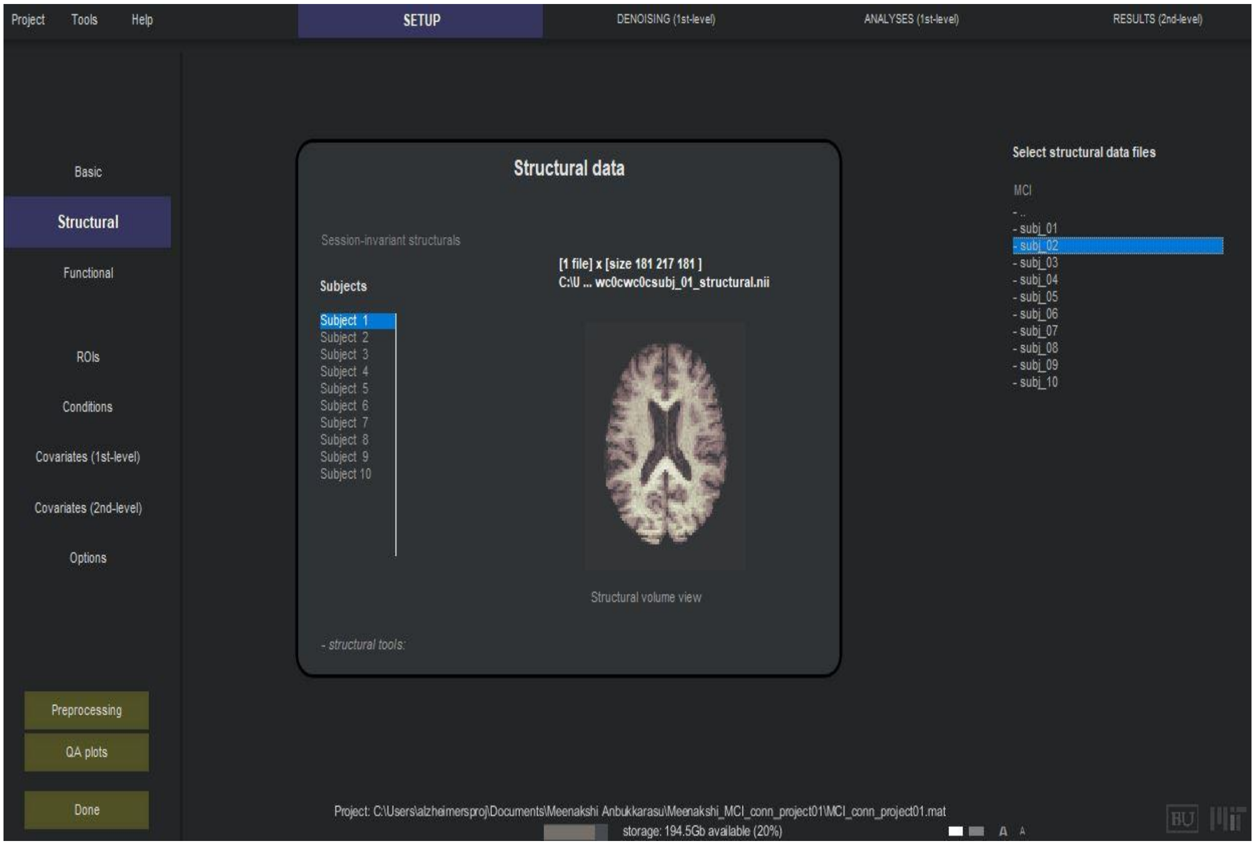


SETUP

Characterizes essential experiment data, ROIs(seeds), second-level models, and temporal covariates. Alternatively, performs anatomical and functional preprocessing steps if needed, including slice timing rectification, realignment, anomaly identification, co-registration, segmentation, and smoothing. This progression incorporates determining the quantity of subjects, designating the structural and functional fMRI data to the respective subjects. Fig.2. shows the CONN GUI for setup.

Figure 2

CONN GUI for Setup



[1 file] x [size 181 217 181]
C:\U ... wc0cwc0csubj_01_structural.nii

Project: C:\Users\alzheimersproj\Documents\Meenakshi Anbukkarasu\Meenakshi_MCI_conn_project01\MCI_conn_project01.mat
storage: 194.5Gb available (20%)

PRE-PROCESSING

The fMRI preprocessing pipeline portrays standard and progressed preprocessing steps in fMRI. These means are planned for adjusting or limiting the impact of notable components influencing the nature of functional and anatomical MRI information, including impacts emerging from patient movement inside the scanner, transient and spatial image contortions because of the consecutive nature of the scanning procurement protocol and anatomical contrasts among patients.

In this project, CONN's default preprocessing pipeline called "default preprocessing pipeline for volume-based examinations (direct standardization to MNI-space)" is utilized. It involves the following steps: unwrap and functional realignment; slice timing rectification; anomaly identification; and functional smoothing. Fig.3. shows a simplified illustration of the default preprocessing pipeline in the CONN toolbox and Fig.4. shows the CONN preprocessing GUI.

Figure 3

Illustration of default minimal preprocessing pipeline in CONN

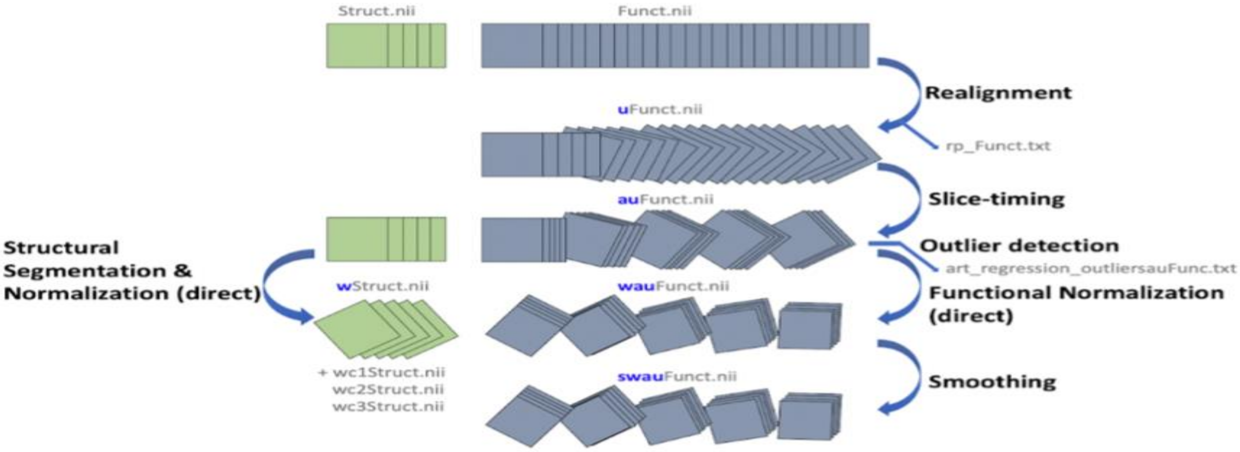
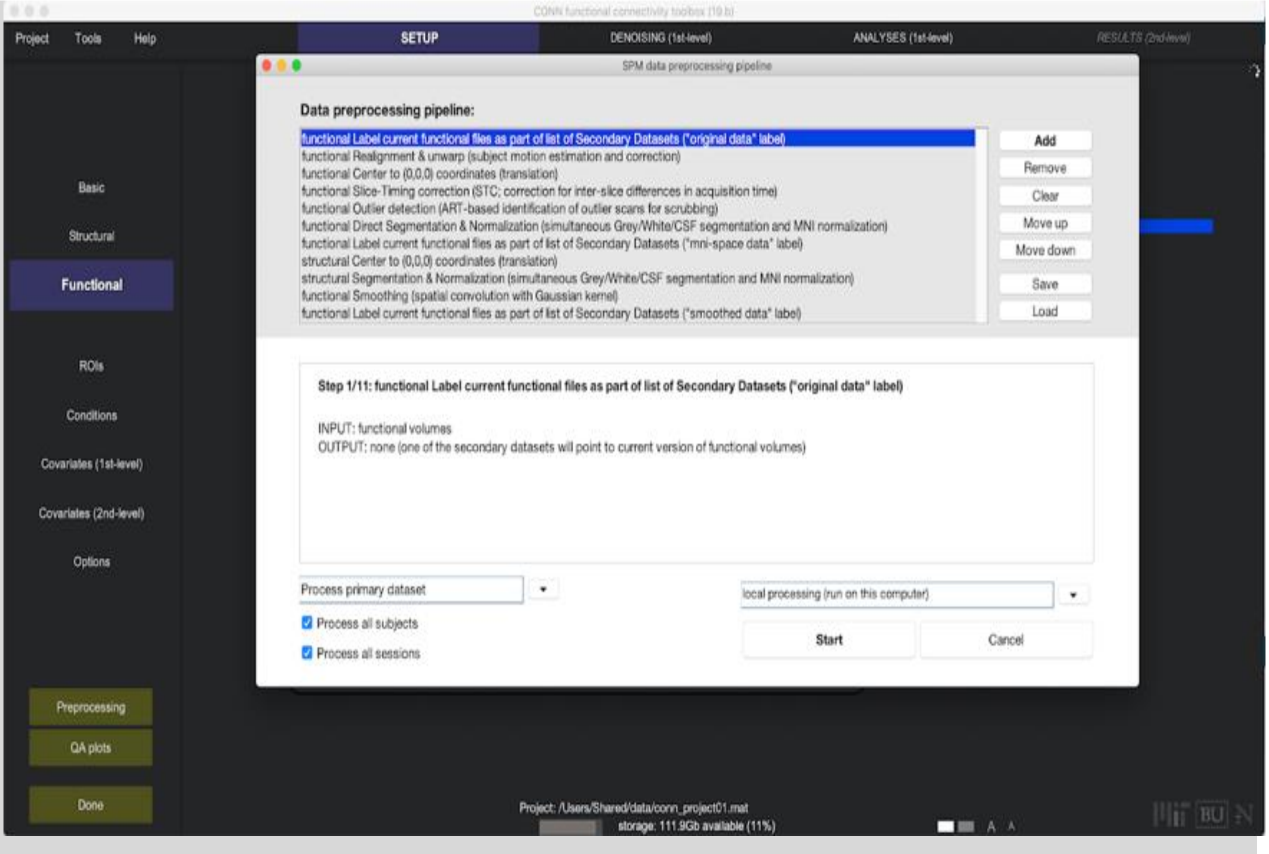


Figure 4

CONN preprocessing GUI



Data preprocessing pipeline:

functional Label current functional files as part of list of Secondary Datasets ("original data" label)
functional Realignment & unwarp (subject motion estimation and correction)
functional Center to (0,0,0) coordinates (translation)
functional Slice-Timing correction (STC; correction for inter-slice differences in acquisition time)
functional Outlier detection (ART-based identification of outlier scans for scrubbing)
functional Direct Segmentation & Normalization (simultaneous Grey/White/CSF segmentation and MNI normalization)
functional Label current functional files as part of list of Secondary Datasets ("mni-space data" label)
structural Center to (0,0,0) coordinates (translation)
structural Segmentation & Normalization (simultaneous Grey/White/CSF segmentation and MNI normalization)
functional Smoothing (spatial convolution with Gaussian kernel)
functional Label current functional files as part of list of Secondary Datasets ("smoothed data" label)

Step 1/11: functional Label current functional files as part of list of Secondary Datasets ("original data" label)

INPUT: functional volumes
OUTPUT: none (one of the secondary datasets will point to current version of functional volumes)

Process primary dataset ▾
local processing (run on this computer) ▾

☑ Process all subjects
☑ Process all sessions

Project: /Users/Shared/data/conn_project01.mat
storage: 111.9Gb available (11%)

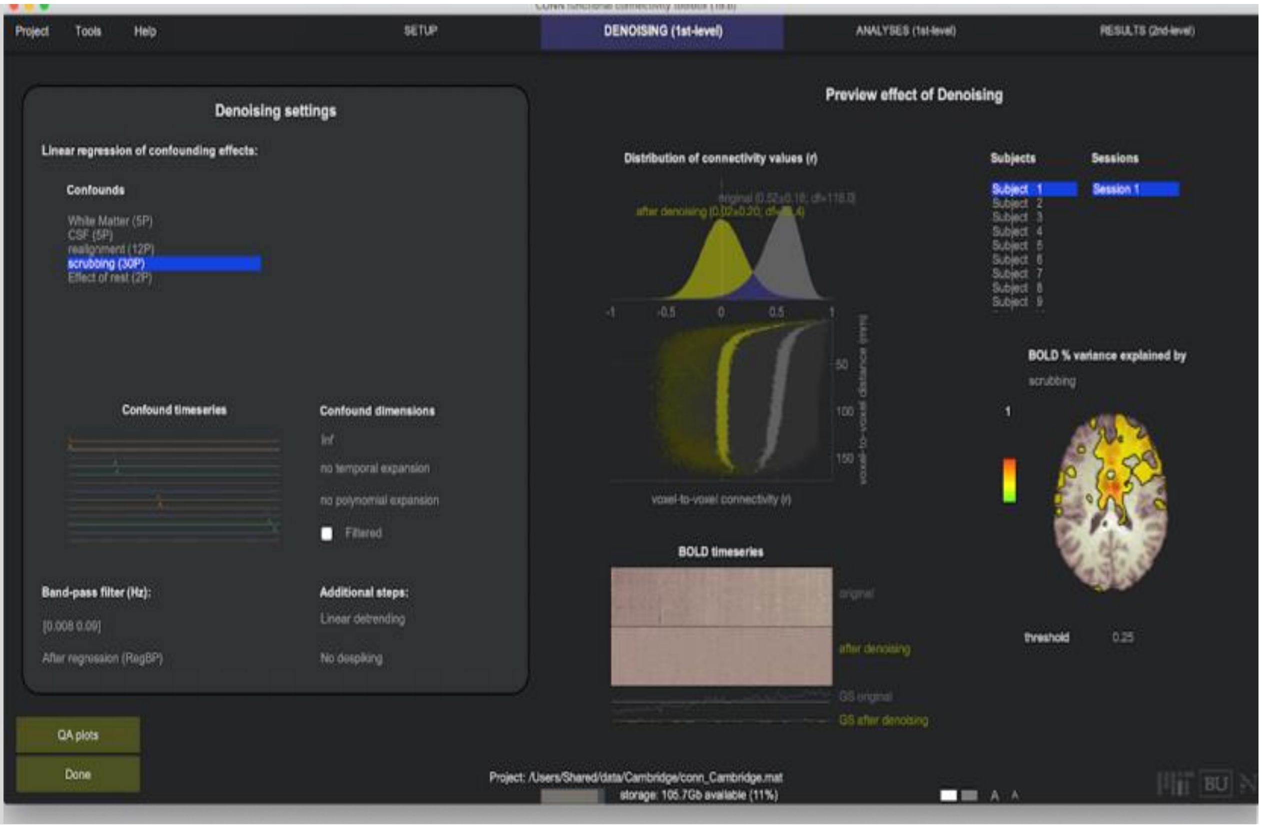
DENOISING:

Once the functional information has been preprocessed, the Blood Oxygen Level Dependent (BOLD) sign may still contain a lot of noise or non-neural inconstancy because of a blend of physiological, aberration, and residual patient-movement effects. These residual variables are especially problematic with regards to fcMRI investigations since they present extremely solid and perceptible inclinations in all functional connectivity measures. Due to this, customary preprocessing steps with regards to fcMRI examinations have supported significantly more conservative procedures than those regularly found in activation-based fMRI examination. These extra techniques are frequently confined under the general term of denoising.

For this project, CONN's default denoising pipeline is utilized. It comprises two stages: temporal band-pass filtering and linear regression of potential perplexing impacts in the BOLD. Potential perplexing effects perform an anatomical component-based noise rectification strategy and incorporate evaluated subject-movement parameters (Friston et al., 1995), noise segments from cerebral white matter and cerebrospinal regions (Behzadi et al., 2007), and recognized anomaly outliers or scrubbing (Power et al. 2014).

Figure 5

CONN's denoising GUI



Denoising settings

Linear regression of confounding effects:

Confounds
White Matter (5P)
CSF (5P)
realignment (12P)
scrubbing (30P)
Effect of rest (2P)

Confound dimensions
Inf
no temporal expansion
no polynomial expansion
□ Filtered

Band-pass filter (Hz):
[0.008 0.09]
After regression (RegBP)

Additional steps:
Linear detrending
No despiking

Subjects: Subject 1
Sessions: Session 1

BOLD % variance explained by
scrubbing

threshold 0.25

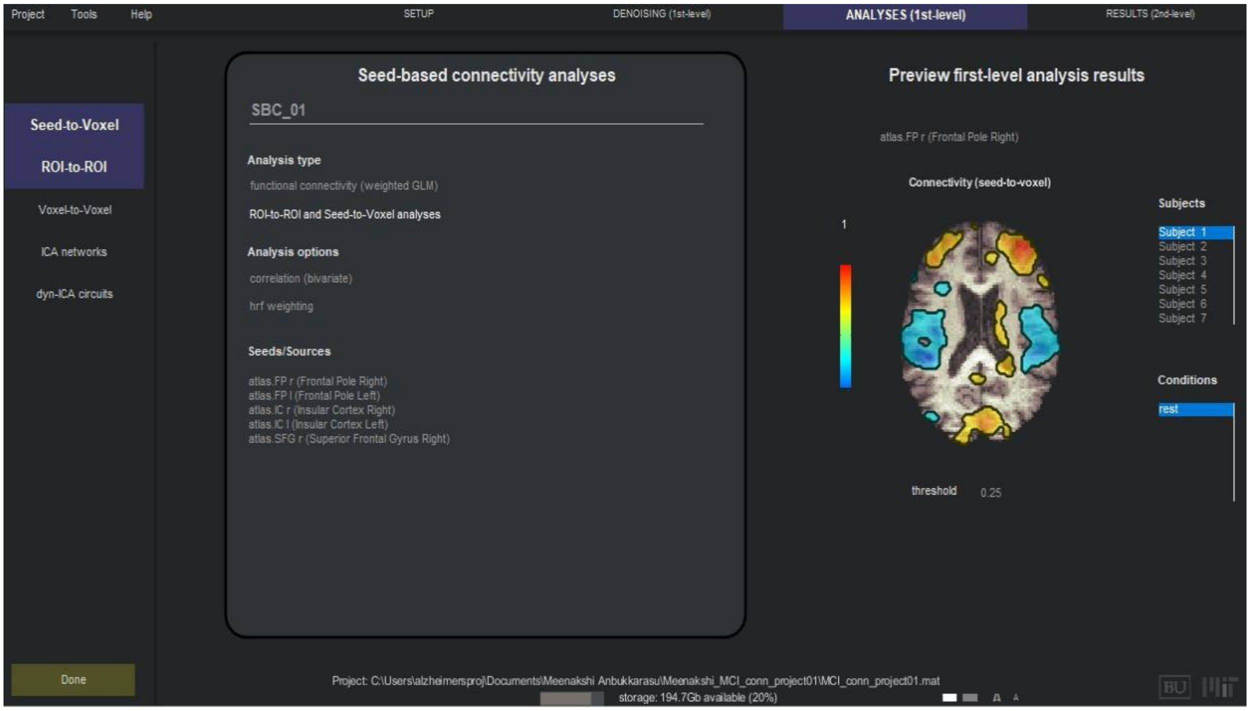
Project: /Users/Shared/data/Cambridge/conn_Cambridge.mat
storage: 105.7Gb available (11%)

ANALYSES (LEVEL 1):

The fMRI endeavors to evaluate the degree of functional integration across various brain regions by estimating the temporal interrelation among the BOLD sign changes in these regions. Despite the relative homogeneity of its definition, there are effectively many distinctive functional connectivity measurements and analytic methodologies. For our project, we utilized Seed-based availability measurements. It portrays the network connectivity with a pre-characterized seed or ROI. These measurements are generally utilized when scientists are keen on one, or a couple of individual regions and might want to break down in detail the network designs between these regions and the remainder of the brain. Fig.6. sows the CONN analyses GUI.

Figure 6

Conn analyses (level 1) GUI



Seed-based connectivity analyses

SBC_01

Analysis type

functional connectivity (weighted GLM)

ROI-to-ROI and Seed-to-Voxel analyses

Analysis options

correlation (bivariate)

hrf weighting

Seeds/Sources

atlas.FP r (Frontal Pole Right)
atlas.FP l (Frontal Pole Left)
atlas.IC r (Insular Cortex Right)
atlas.IC l (Insular Cortex Left)
atlas.SFG r (Superior Frontal Gyrus Right)

Preview first-level analysis results

atlas.FP r (Frontal Pole Right)

Connectivity (seed-to-voxel)

threshold 0.25

Subjects

Subject 1
Subject 2
Subject 3
Subject 4
Subject 5
Subject 6
Subject 7

Conditions

rest

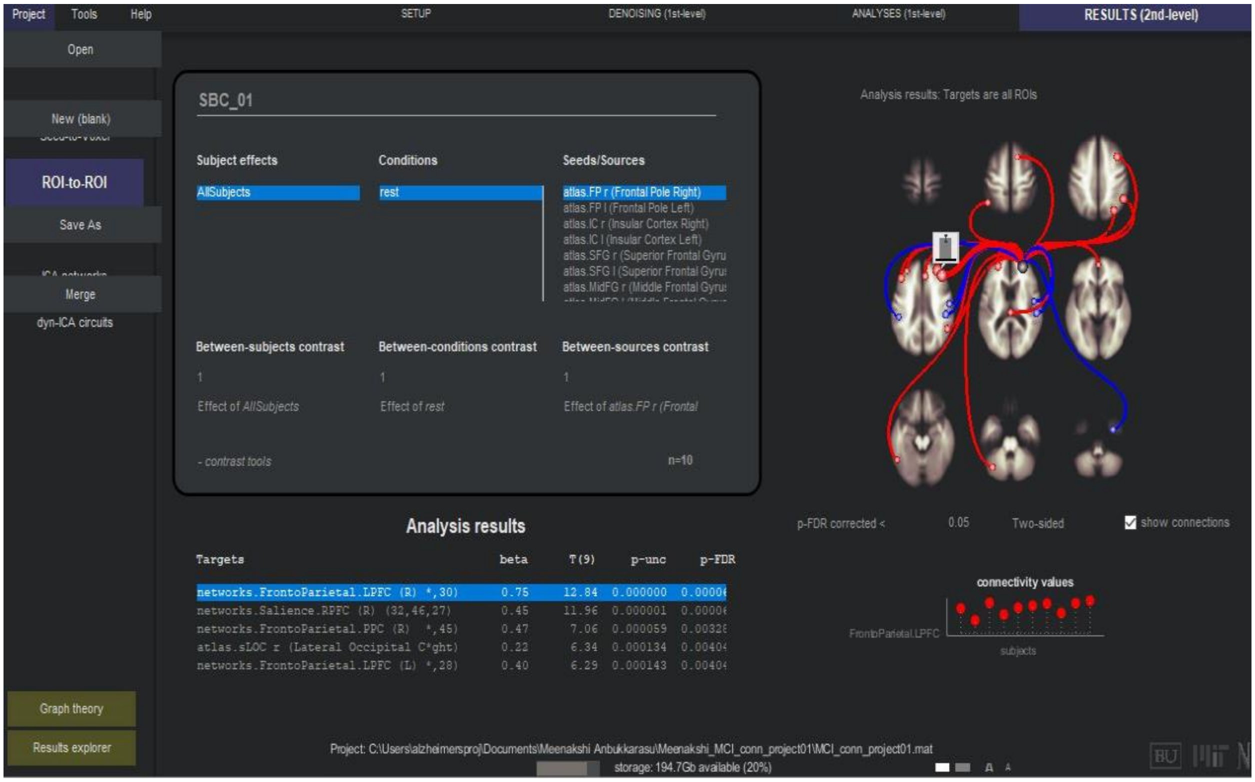
Project: C:\Users\alzheimersproj\Documents\Meenakshi Anbukkarasu\Meenakshi_MCI_conn_project01\MCI_conn_project01.mat
storage: 194.7Gb available (20%)

RESULTS (LEVEL 2)

The second level examination permit analysts to make assumptions about the properties of a population, by deriving from the investigation of just a subset of patients in an examination. The General Linear Model (GLM) is the methodology that is utilized in CONN for all second-level examination of functional connectivity measurements. This depiction incorporates GLM model definition, parameter estimation, and inference testing framework that planned to assist scientists with contemplating the properties of seed-based maps over various patients or ROI-to-ROI network matrices over numerous patients. Fig.7. shows the CONN results (level 2) GUI.

Figure 7

CONN results – level 2 GUI.



SBC_01

Subject effects: AllSubjects

Conditions: rest

Seeds/Sources:
atlas.FP r (Frontal Pole Right)
atlas.FP l (Frontal Pole Left)
atlas.IC r (Insular Cortex Right)
atlas.IC l (Insular Cortex Left)
atlas.SFG r (Superior Frontal Gyru
atlas.SFG l (Superior Frontal Gyrus
atlas.MidFG r (Middle Frontal Gyrus

Between-subjects contrast: 1 — Effect of *AllSubjects*

Between-conditions contrast: 1 — Effect of *rest*

Between-sources contrast: 1 — Effect of *atlas.FP r (Frontal*

- contrast tools n=10

p-FDR corrected < 0.05 Two-sided ☑ show connections

Analysis results

```
Targets                                        beta     T(9)    p-unc     p-FDR
networks.FrontoParietal.LPFC (R) *,30)         0.75    12.84  0.000000  0.0000
networks.Salience.RPFC (R) (32,46,27)          0.45    11.96  0.000001  0.0000
networks.FrontoParietal.PPC (R)  *,45)         0.47     7.06  0.000059  0.0032
atlas.sLOC r (Lateral Occipital C*ght)         0.22     6.34  0.000134  0.0040
networks.FrontoParietal.LPFC (L) *,28)         0.40     6.29  0.000143  0.0040
```

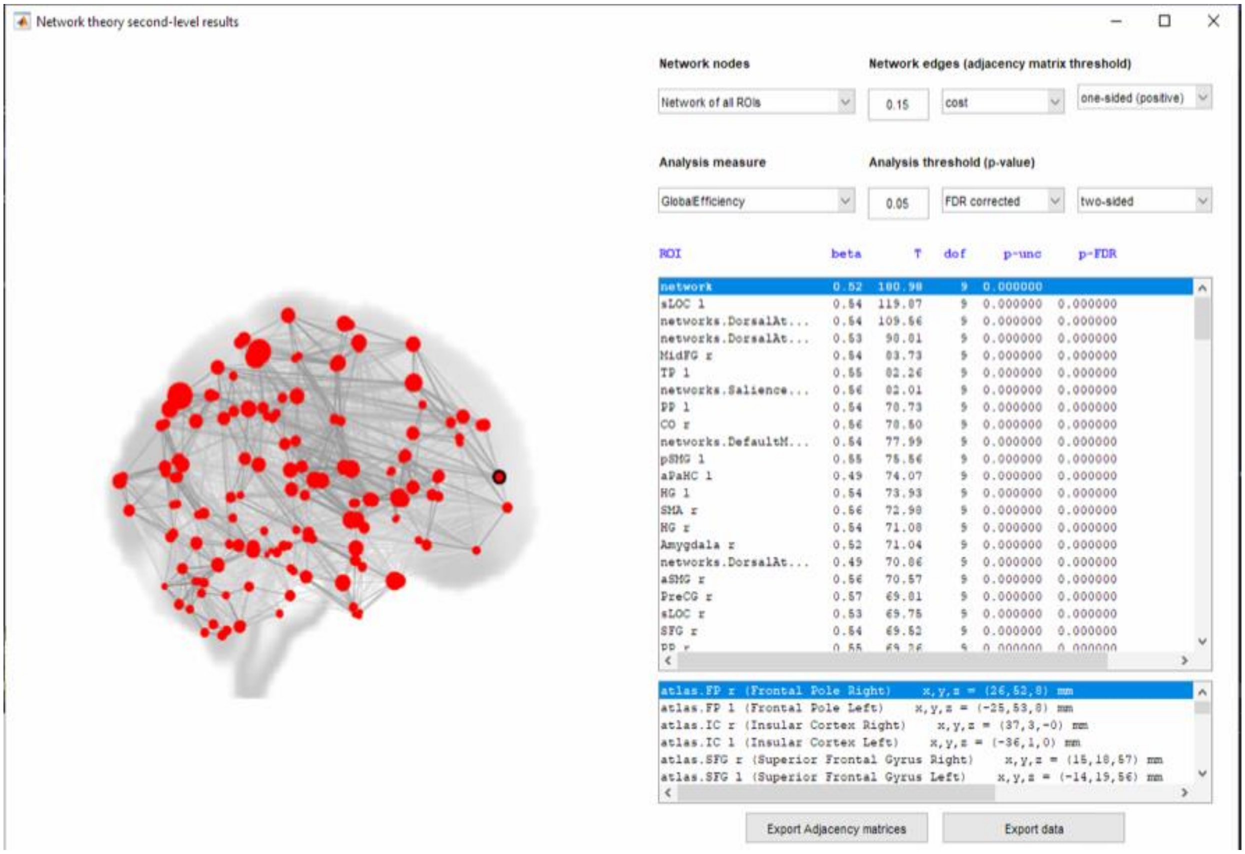
Project: C:\Users\alzheimersproj\Documents\Meenakshi Anbukkarasu\Meenakshi_MCI_conn_project01\MCI_conn_project01.mat

storage: 194.7Gb available (20%)

All ROI level graph measures illustrated below in Fig.8 depend on user characterized nondirectional graphs where nodes are the ROIs, and edges are the prior threshold associations. For each patient, an adjacency matrix is calculated by thresholding the related ROI-to-ROI Correlation matrix (RRC) by a relative or non-relative threshold. Then, from the emerging graphs, various measures can be calculated addressing to topological properties of every ROI in the graph and the whole network of ROIs. Cost at every node or ROI shows the amount of network centrality, describing the level of local connectedness of every ROI in a graph.

Figure 8

CONN GUI for graph theory that contains the RRC matrix information.



Network theory second-level results

Network nodes: Network of all ROIs

Network edges (adjacency matrix threshold): 0.15 cost one-sided (positive)

Analysis measure: GlobalEfficiency

Analysis threshold (p-value): 0.05 FDR corrected two-sided

```
ROI                      beta      T   dof     p-unc      p-FDR
network                  0.52  100.98    9  0.000000
sLOC l                   0.54  119.07    9  0.000000  0.000000
networks.DorsalAt...     0.54  109.56    9  0.000000  0.000000
networks.DorsalAt...     0.53   90.01    9  0.000000  0.000000
MidFG r                  0.54   83.73    9  0.000000  0.000000
TP l                     0.55   82.26    9  0.000000  0.000000
networks.Salience...     0.56   82.01    9  0.000000  0.000000
PP l                     0.54   78.73    9  0.000000  0.000000
CO r                     0.56   78.50    9  0.000000  0.000000
networks.DefaultM...     0.54   77.99    9  0.000000  0.000000
pSMG l                   0.55   75.56    9  0.000000  0.000000
aPaHC l                  0.49   74.07    9  0.000000  0.000000
HG l                     0.54   73.93    9  0.000000  0.000000
SMA r                    0.56   72.98    9  0.000000  0.000000
HG r                     0.54   71.08    9  0.000000  0.000000
Amygdala r               0.52   71.04    9  0.000000  0.000000
networks.DorsalAt...     0.49   70.86    9  0.000000  0.000000
aSMG r                   0.56   70.57    9  0.000000  0.000000
PreCG r                  0.57   69.81    9  0.000000  0.000000
sLOC r                   0.53   69.75    9  0.000000  0.000000
SFG r                    0.54   69.52    9  0.000000  0.000000
PP r                     0.55   69.26    9  0.000000  0.000000
```

```
atlas.FP r (Frontal Pole Right)    x,y,z = (26,52,8) mm
atlas.FP l (Frontal Pole Left)    x,y,z = (-25,53,8) mm
atlas.IC r (Insular Cortex Right)    x,y,z = (37,3,-0) mm
atlas.IC l (Insular Cortex Left)    x,y,z = (-36,1,0) mm
atlas.SFG r (Superior Frontal Gyrus Right)    x,y,z = (15,18,57) mm
atlas.SFG l (Superior Frontal Gyrus Left)    x,y,z = (-14,19,56) mm
```

ADJACENCY MATRIX

The adjacency or RRC matrix describes the connectivity between all pairs of ROIs among a pre-characterized set of brain regions. The meaning of these measurements follows the very same association and properties as that of the seed-based connectivity measures in the earlier sections. These measurements are frequently utilized when scientists are keen on the simultaneous investigation of complete networks of connections. A pair of ROIs in the matrix has an entry as 1 if there exists a connection between them else has a zero. For each category of patients, we export the RRC matrix for 11 different cost functions [0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6] and process it to generate the dataset for classification. The following section explains the dataset generation in detail.

EXPERIMENTAL METHODS

ENVIRONMENT

The entire project is developed by utilizing the Python programming language. The explanation is that Python is an adaptable language with a lot of machine learning libraries streamlining the efforts of creating AI applications. There is additionally a lot of online help accessible to build the machine learning models utilizing Python. Python additionally gives a lot of libraries to code the machine learning algorithms like Scikit-learn and Keras. We have utilized scikit learn to implement classifiers like decision tree, Random Forest, Adaboost, Logistic Regression, and SVM. Whereas Keras which utilizes TensorFlow as the backend is used for deep learning classification. The Python 3.6.8

version was utilized in this project. The editor utilized was Jupyter Notebook which has installed Python kernel and the other kernels requiring no additional set up. The code is executed on a machine with a higher limit of Graphics Processing Unit (GPU) and Tensor Processing Unit (TPU) offering access to additional processing power as machine learning model training includes plenty of calculations particularly for a large dataset.

FRAMEWORKS

PANDAS

Pandas is a Python library that offers a framework to build information into a table format and allows us to manipulate the rows, columns, and cells. It is a valuable tool to store, inspect, and extract data for observation. It can also arrange the data into a form that can be used by machine learning methods.

NUMPY

NumPy is a Python library that is utilized alongside the pandas library to deal with multidimensional information and perform complex logical and scientific procedures on the information.

SCIKIT_LEARN

Scikit – learn is another Python library that gives the capacities to effectively construct different classification, clustering, and regression algorithms. It permits to build a pipeline and verify the output with an assortment of evaluation measurements. It also permits to change the algorithm by tuning the attributes of the machine learning models. It contains methods that are utilized to preprocess data, select and lessen the dimension of the feature set, etc.

KERAS

Keras is a neural network library in Python which gives a layer of reflection to the complications of making a neural network and which increases the process of rapid testing.

DATASET GENERATION

The RRC matrix obtained from CONN for a single cost function is in the form of .mat file and it contains the RRC for all the subjects together as a single list. So, the content of the mat file needs to be processed to retrieve the RRC connectivity of each patient. In our case, we have 10 RRC matrices each corresponding to our 10 patients. Note that the number of ROIs in our case is 164 and hence the connectivity matrices are of dimension 164 X 164. Since brain connectivity graphs are undirected the RRC matrices are always square matrix. A probability matrix of dimension 164 X 164 is calculated using the 10 RRC square matrices where each cell corresponds to the ratio of numbers of subjects having connectivity between a pair of ROIs to the total number of subjects. Each cell of the computed probability matric is then transferred to a CSV file along with its cost value and pair of ROI names. These correspond to the feature set of our dataset. A target column with labels zero for HC patients and 1 for MCI is also added to the .csv file. The process of data generation is repeated for different cost functions for both MCI and HC patients. The final .csv file has 297660 rows with four features namely (Cost, ROI_A, ROI_B, and

Probability) and one target column. This file is used as the input for all the machine learning classifiers.

DATA SET LABELING

POSITIVE DATASET

Each of the pre-processed data related to Healthy Controls is labeled with “0". Each data corresponds to a row in the data set.

NEGATIVE DATASET

Each of the pre-processed data related to MCI patients is labeled with “1". Each data corresponds to a row in the data set

DATA VISUALIZATION

Data visualization is the portrayal of information or data in a figure, graph, or other visual arrangements. It conveys connections of the data with visual techniques. This is significant as it permits patterns and trends to be all the more effectively observed. With the ascent of enormous information upon us, we should have the option to decipher progressively bigger clusters of data. Machine learning makes it simpler to conduct investigations, which would then be able to fill in as supportive visualization to present. In this project, we utilized the histogram plot to understand the distribution of data. The x-axis corresponds to discrete intervals for the examination and the y-axis corresponds to the recurrence or tally of the quantity of examinations in the dataset that have a place with each interval.

SCALING

Machine learning classifiers perform better or converges quicker when features are on a generally comparable scale as well as near to normally distributed. We have experimented with the two most well-known scaling techniques to be specific Normalization and Standardization. Standardization, for the most part, implies standardizing the features by eliminating the mean and scaling to unit fluctuations. Whereas, normalization implies changing features by scaling every feature to a given interval.

Figure 9

Histogram of normalized data

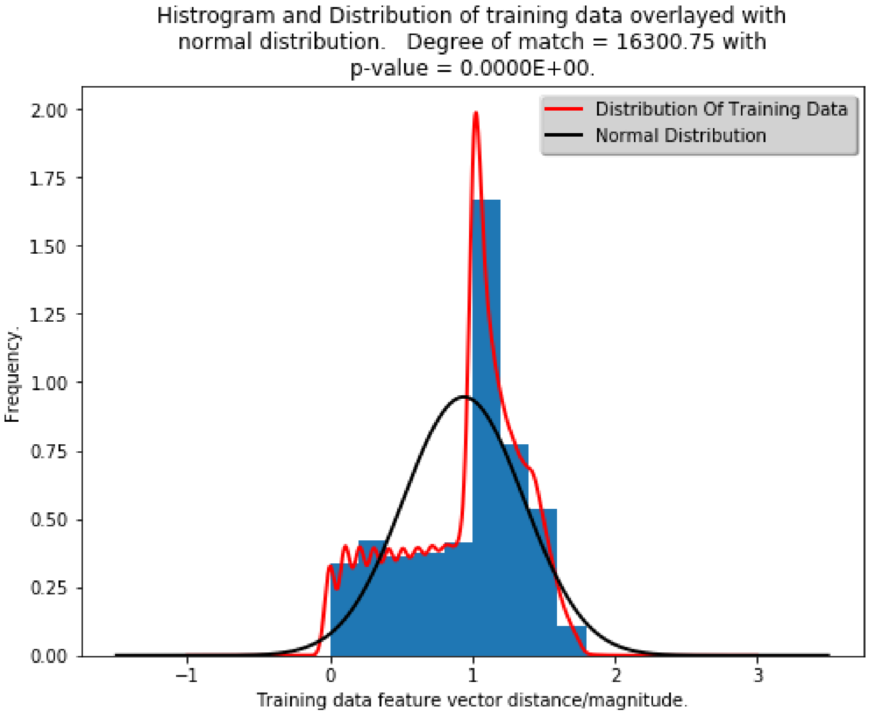
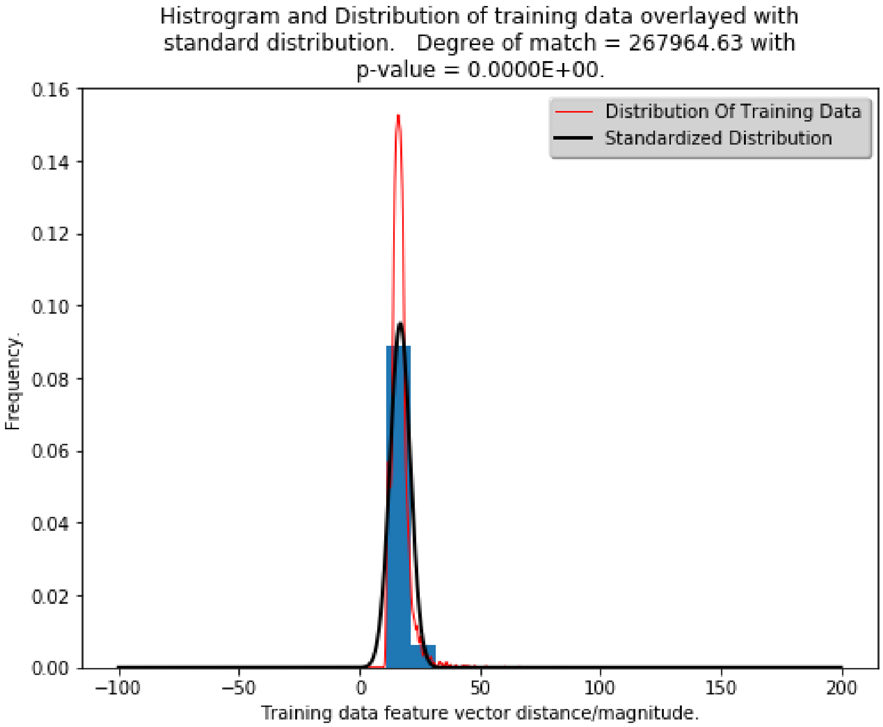


Figure 10

Histogram of Standardized data.



From Fig. 9 and Fig. 10 we can see that standardization of data has a better degree of match when compared to normalization. So, the standardized data was used as input to all the machine learning models.

DATASET SPLIT

The training set contains a known result and the model studies this data so as to classify new data in the future. We have the test dataset to assess the classifier's prediction on this batch of data. Utilizing the Scikit-learn library and especially the train_test_split method we split the entire data thereby allocating 80 % of the data for training and 20 % for testing.

CLASSIFIERS AND THEIR PARAMETER SELECTION

DECISION TREE

The decision tree is a popular machine learning algorithm that is used for both classification and regression. It is a tree where each node signifies a feature, each branch signifies a decision and each leaf signifies a target class. It works well for binary classification for non-linear sets. The decision tree is a greedy algorithm that performs a split on each of the features of the data at a specific threshold. It tends to maximize the difference between the parent node loss and the sum of losses of all the child nodes to select the split. Here we use the Gini loss function.

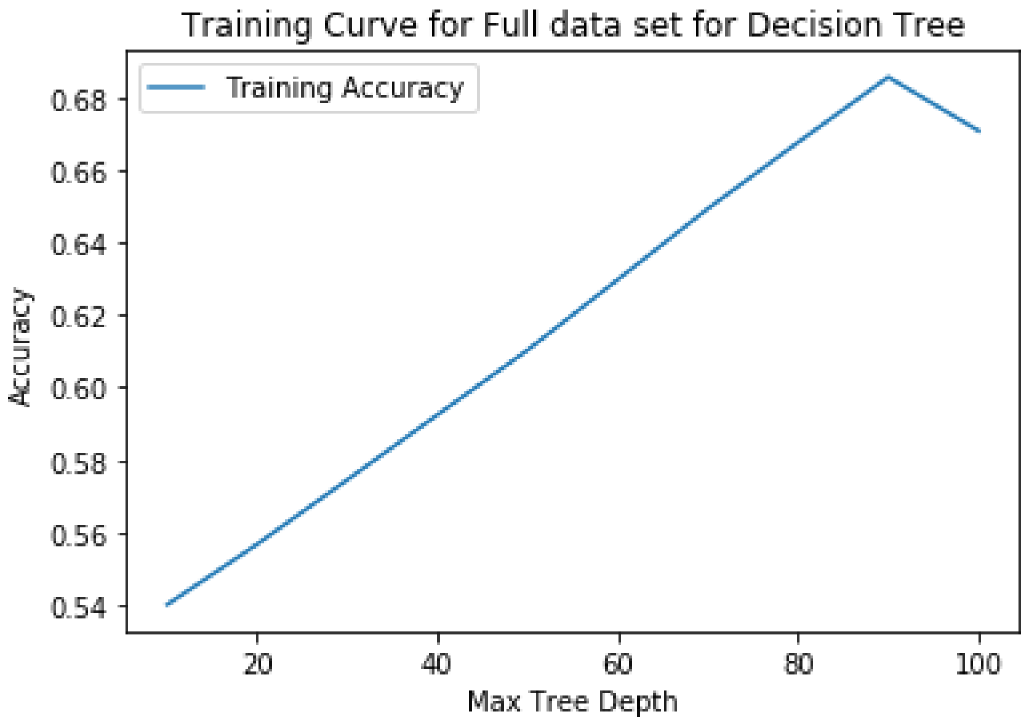
Because of the greedy approach, the decision tree always tends to overfit the test data. There are several methods to prevent overfitting like decreasing the number of leaf nodes or fixing a maximum depth. For this project, we decided to regularize using maximum depth. All of these decision tree classifiers were implemented using the sci-kit learn library.

The decision tree classifier creates a tree with the help of the training data then uses the resulting tree to test the test data. While this usually performs well, these individual decision trees tend to have high variance which leads to low accuracy. In this project, we have used the sklearn library function to implement the classic decision tree algorithm. As mentioned earlier the overfitting is prevented by setting a maximum tree depth.

In order to tune the maximum depth, sci-kit learn's validation curve function has been used to perform 5-fold cross-validation on the dataset. The training accuracy curve is shown in Fig.11 below as a function of the maximum tree depth. We used the famous "Grid Search" and the validation curve method to configure optimal parameters for a given model. Using this method, the results indicate that a maximum depth of 90 can be used in the training phase.

Figure 11

Training Curves for using Ordinary Decision Trees.

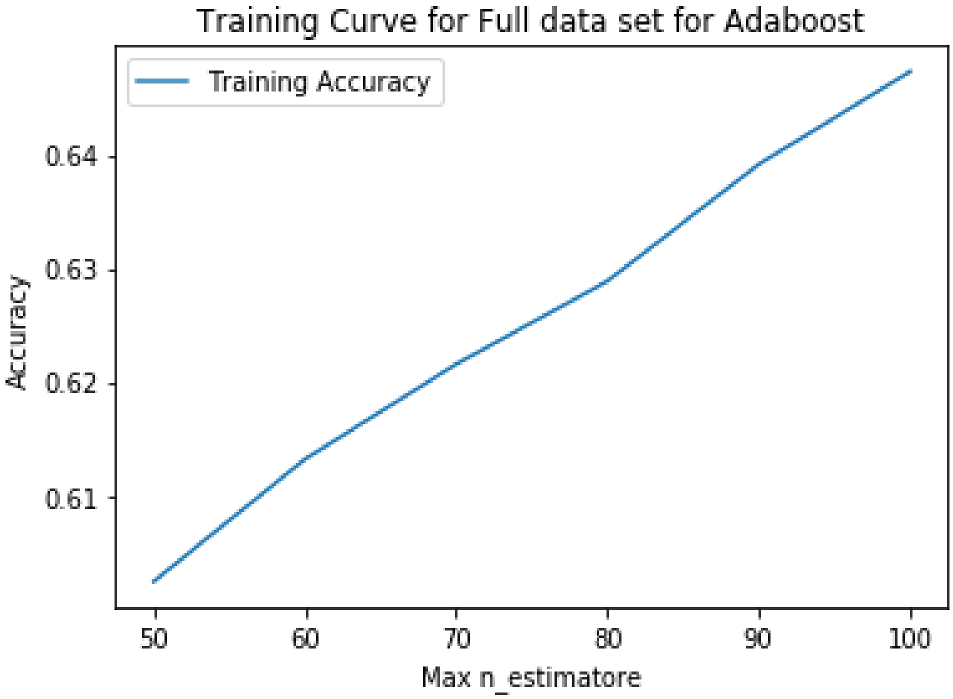


ADA BOOST

Boosted Decision trees are formed by combining multiple simple decision trees (weak learners). They are used to decrease the bias in the model with an increase in variance. Though each tree is weak as an individual, when put together they form a strong learner. Boosted are trees are ideally used to improve the accuracy, though it may lead to overfitting. In this project, we used a specific boosting algorithm called AdaBoost. For boosted decision trees we manually tuned the maximum depth of base decision tree and also the number of weak learners. We computed the sci-kit learn validation curve for the different number of weak learners (50, 60, 70, 80, 90, 100). Fig.12. shows the training accuracy curve for the full data set.

Figure 12

Validation Curves for Full dataset using AdaBoost.



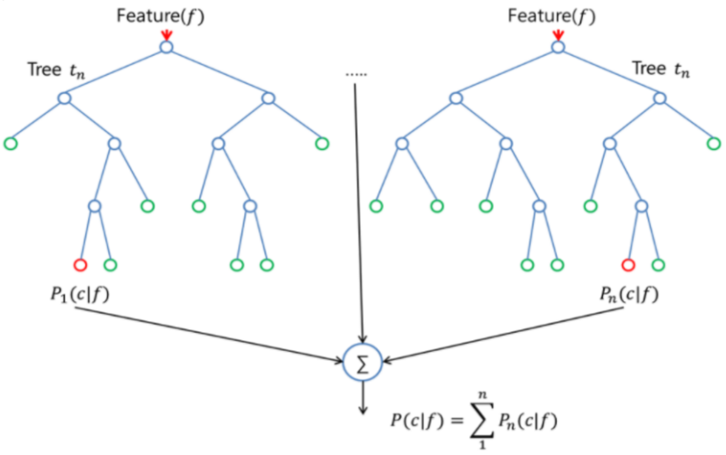
From the results, it was found that 100 weak learners performed better than others. So, we decided to use 100 weaker learners with each having a maximum depth of 10.

RANDOM FOREST

The random forest classifier is another type of ensemble technique for decision trees that are used for improving accuracy. It uses a bagging technique that is if there are n samples in the dataset D, the model performs sampling with random replacement from D. This reduces the variance (at the expenses of increase in bias) and helps in avoiding overfitting. The higher bias can be reduced by considering only a random subset of features while splitting a node in the decision tree. In this project, we have taken only a square of the total number of features at each split.

Figure 13

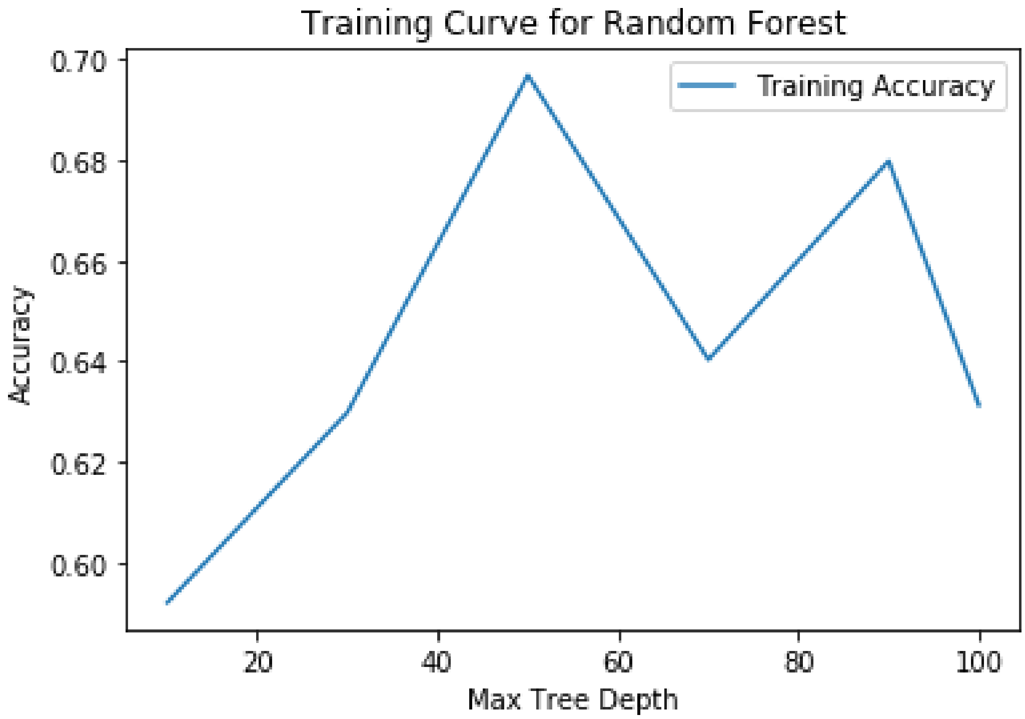
Validation curve for maximum depth using Random Forest.



For our purpose, we decided to include 100 decision trees in our model because as seen before 100 has a better trade-off between time and accuracy. Having more trees in the random forest only decreases the variance but does not increase the likelihood of overfitting. Scikit learn's validation curve function with 5-fold cross-validation for tuning the maximum depth of the decision tree is shown below in Fig.14. as a maximum depth function.

Figure 14

Validation curve for maximum depth using Random Forest.



The training accuracy curve the results suggest that a maximum depth of 50 works well for the dataset. Once a decision has been made, the datasets are trained with the number of trees as 100 and max depth as 50.

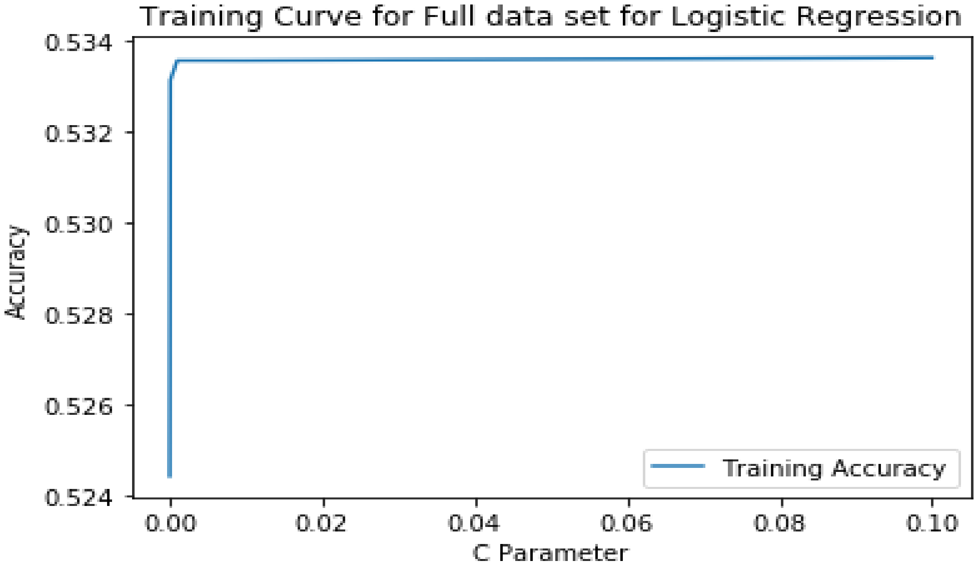
LOGISTIC REGRESSION

The implementation is carried out with a standard logistic regression model with L2 regularization having C parameter inversely proportional related to the strength of regularization. This is done to decrease the overfitting of the model. To select the optimal C parameter, the input data were trained over a range of C values and the one that produced the highest accuracy was selected for the testing phase. Stochastic Average Gradient (SAG) descent was used as the solver as it generally convergences faster for large feature-sets such as ours.

The scikit learn's validation curve was used to find the optimal C parameter value. Fig.15. shows the accuracy validation curve as a function of C-Parameters. From the results, it is suggested to use 0.001 as the C parameter.

Figure 15

Logistic regression validation curve as a function of C-Parameters

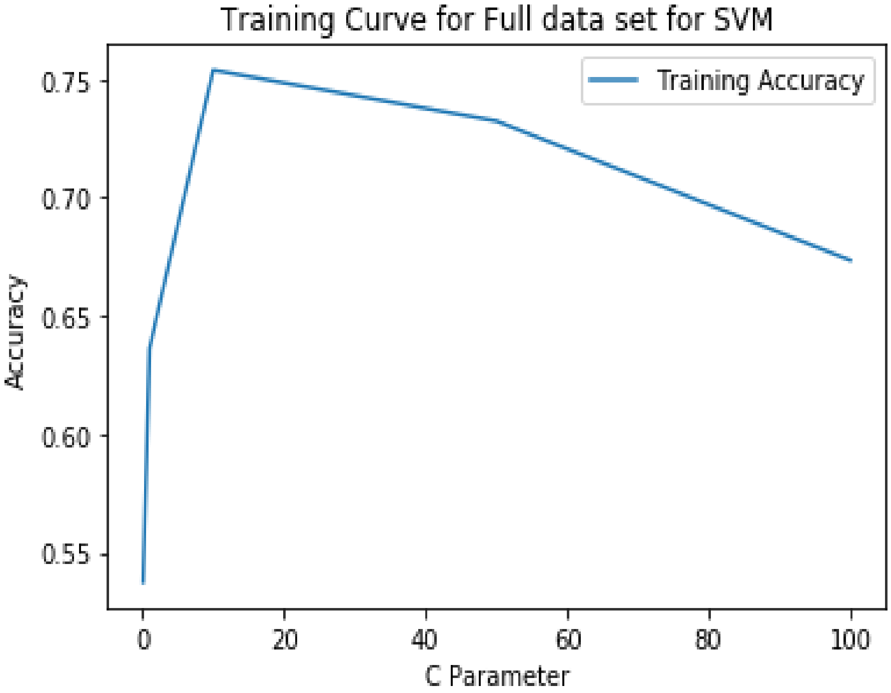


SUPPORT VECTOR MACHINE

We wanted to explore a nonlinear model to check if it can outperform logistic regression. Our natural choice was to use SVM as it is easy to implement because it requires only a few parameter tunings. When data is trained with SVM, the model creates hyperplanes to separate the training data into its respective classes. This splitting is done with the help of kernel function which maps the training data into a higher dimension so that the data can be separated linearly. Here we tried two kernel functions namely linear and radial basis function (RBF) to see which one that suits best for our purpose. We also experimented with different C parameters because selecting the optimal kernel and C parameter is essential when using SVM.

Figure 16

SVM validation curve as a function of C-Parameters



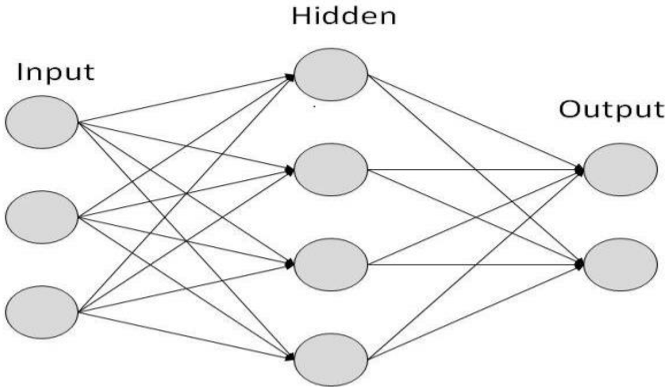
Scikit learn's validation curve function was used to tune the C parameter over a list of values [0.1, 1.0, 10.0, 50.0, 100.0]. From the accuracy validation curve shown in Fig. 16, we found that maximum accuracy is provided by the C parameter value of 10.0. Once the C parameter was fixed, we applied SVM to the training data and compared the accuracy obtained against one another.

DEEP NEURAL NETWORK

The deep neural network is a machine learning methodology that is commonly used in supervised learning problems. It is a feedforward neural network; that is, the information only moves forward and never backward, and there is no cycle formed by connections between nodes. Deep Learning consists of an input layer, an output layer, and one or more hidden layers. Non-linear activation functions are applied on all layers (except the input layer). The model learns through the process of backpropagation in which weights and biases of neurons in every layer are adjusted to reduce the error function.

Figure 17

Structure of Artificial Neural Network

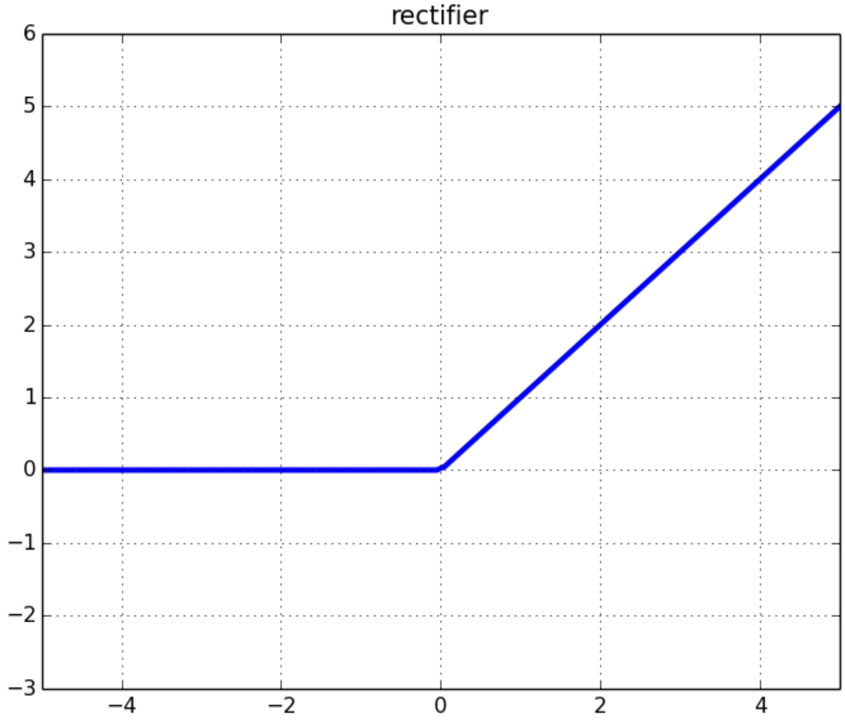


The implementation is carried out using the Keras library. This is done by creating a sequential model and adding dense layers - an input layer, hidden layers, and an output layer. Experimented were conducted by training with 3 hidden layers and 4 hidden layers. The model with 3 hidden layers performed much better than 4 hidden layers as it is now able to capture much more complex information. We have not considered more than 4 hidden layers as our research showed that a neural network with 4 hidden layers can represent functions of any shape.

ReLu function was applied as the activation function on the output from the input layer and the hidden layers. Since this is a binary classification problem, the Sigmoid activation function is applied to the output layer.

Figure 18

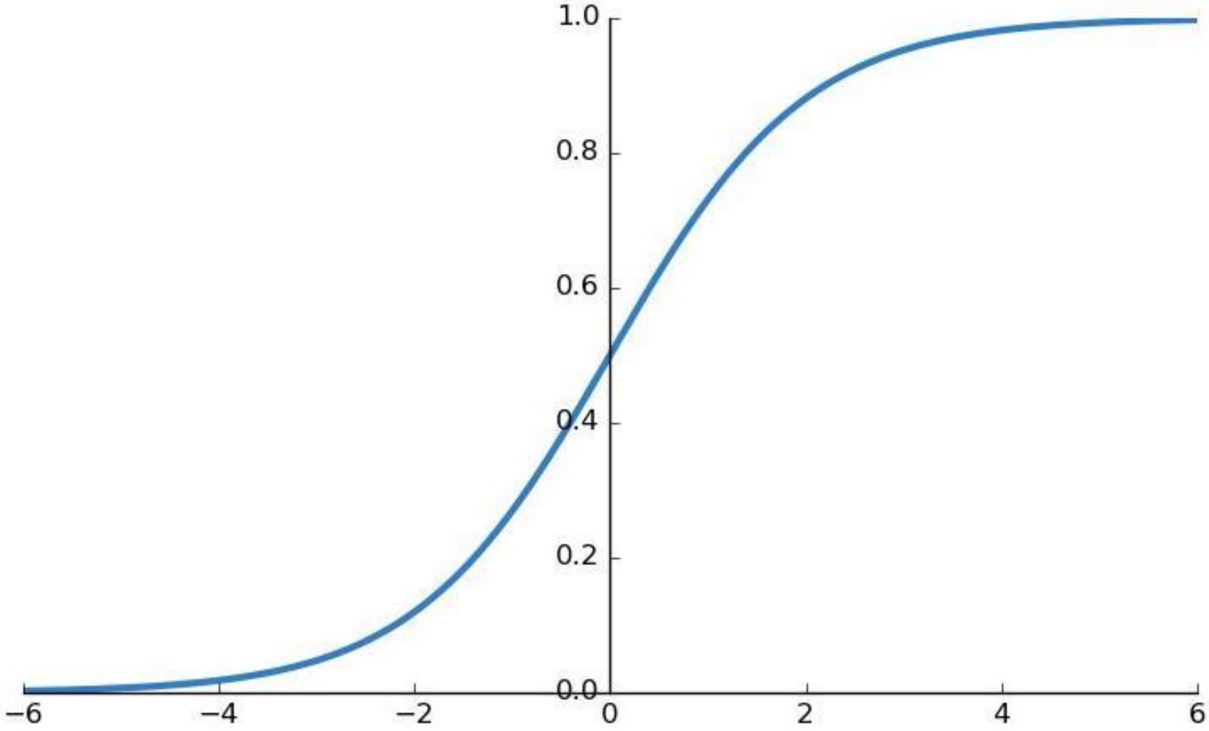
ReLU Curve



The sigmoid function has a characteristic 'S' shaped curved and is widely used in binary classification. The function generates a probability of real numbers in the range [0,1]. Again, as this is a binary classification problem, categorical cross-entropy is used as the loss function.

Figure 19

Sigmoid Curve



We experimented by using different optimizers (RMSprop, SGD, and Adam), and decided to use RMSprop as it converges much faster than Adam and SGD. Finally, dropout regularizations were added to reduce the issue of overfitting. The dropout layer does this by randomly dropping a fraction of neurons (setting their weights to zero), thus reducing the complexity of the model.

ROC CURVES

Receiver Operating Characteristic (ROC) curve gives the performance measure of the classifier by computing ratio of TPR to TNR. The ROC curve evaluates how well the classifier is able to separate the two classes of test instances. The Area Under the Curve (AUC) gives the measure of the separability in the ROC curve. If the AUC for ROC is closer to 1, it means that the classifier is able to distinguish the classes very well while if the AUC is 0.5, it means that the classifier is not able to separate the classes at all. Since classes in the dataset are balanced, we have used accuracy as a metric instead of AUC-ROC. However, ROC curves provide a good way of visualizing the performance of our dataset.

RESULTS AND EVALUATIONS

DECISION TREE CLASSIFIER

Using a maximum depth of 90, the decision tree achieved a test accuracy of 0.66. It is known that decision trees usually overfit training data causing high training accuracy and low-test accuracy. Yet, in our decision tree model, the test accuracy was closer to training accuracy. As this was highly irregular, we investigated the cause by checking for duplicates in the data and randomly reordering the dataset. However, the test accuracy remained the same, we concluded that the high accuracy is because the dataset is very clean and noise-less, causing the test-set to be perfectly matched with the training set. Also, the decision tree misses information, when it groups features with continuous values. This might be another reason for low accuracy. Further, the model is as good as the number of nodes in the tree. Hence, the low accuracy can be improved by having large dataset with more features. The ROC curve and Confusion Matrix (for depth = 90) are shown in Fig.20. and Fig.21. respectively.

Figure 20

ROC Curve for Ordinary Decision Tree

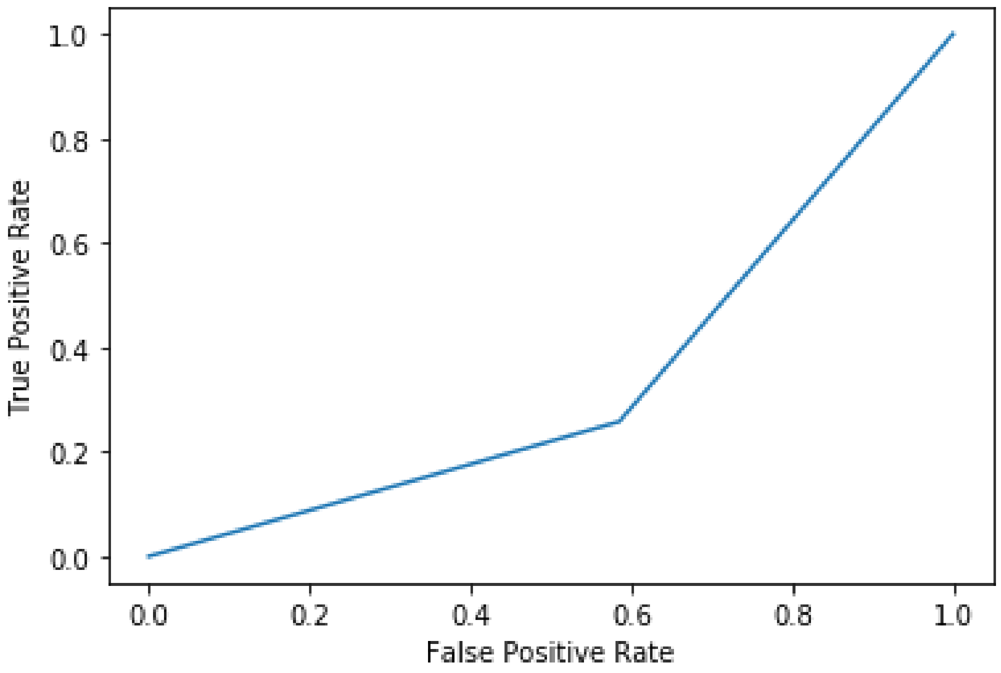
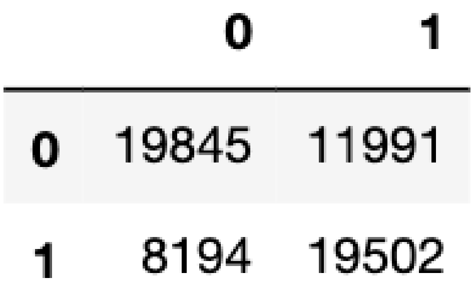


Figure 21

Confusion Matrix for Ordinary Decision Tree (depth = 90)



	0	1
0	19845	11991
1	8194	19502

ADA BOOST

Using a maximum depth of tree as 10, the full dataset achieved an accuracy of 0.65 100 weak learners. The ROC curve and Confusion Matrix are shown in Fig.22. and Fig.23. respectively.

Figure 22

ROC Curve for Full Dataset Using AdaBoost

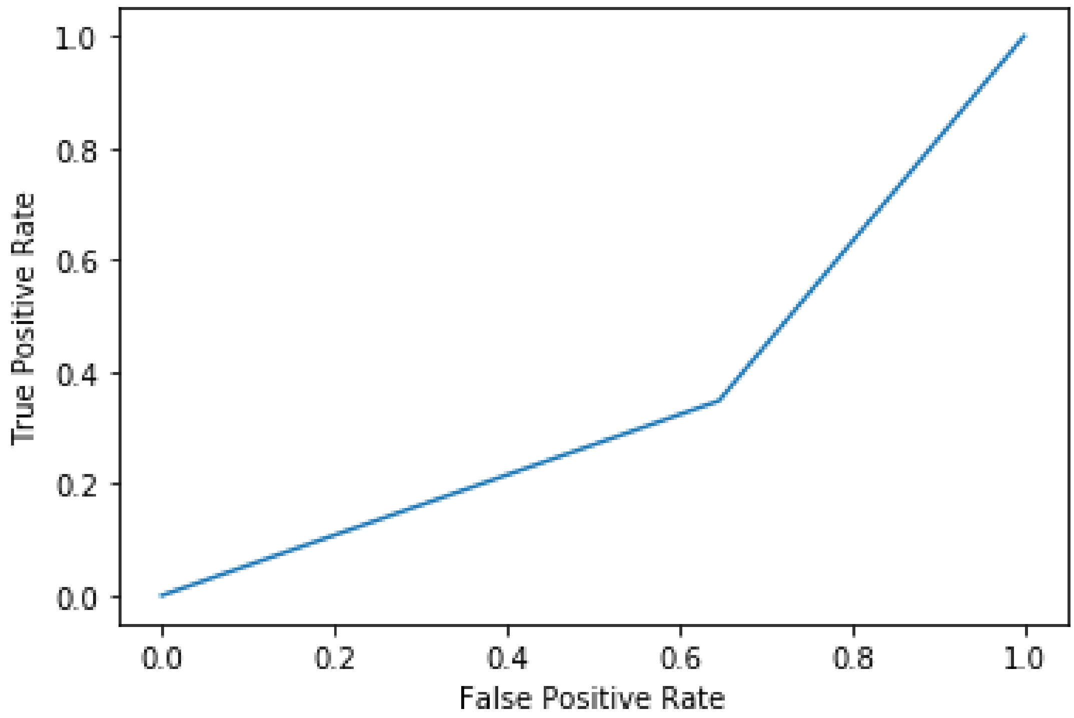
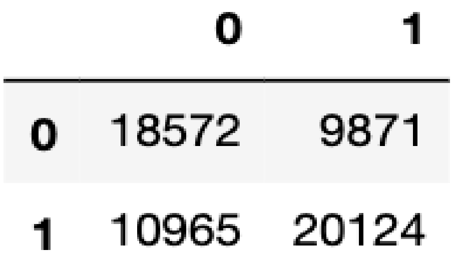


Figure 23

Confusion Matrix for AdaBoost on Subset of Full Dataset



	0	1
0	18572	9871
1	10965	20124

RANDOM FOREST

Using 100 decision trees with each having a maximum depth of 50, we achieved a test accuracy of 0.61. The ROC curve and Confusion Matrix are shown in Fig.24. and Fig.25. respectively.

Figure 24

ROC Curve for Full Dataset Using Random Forest

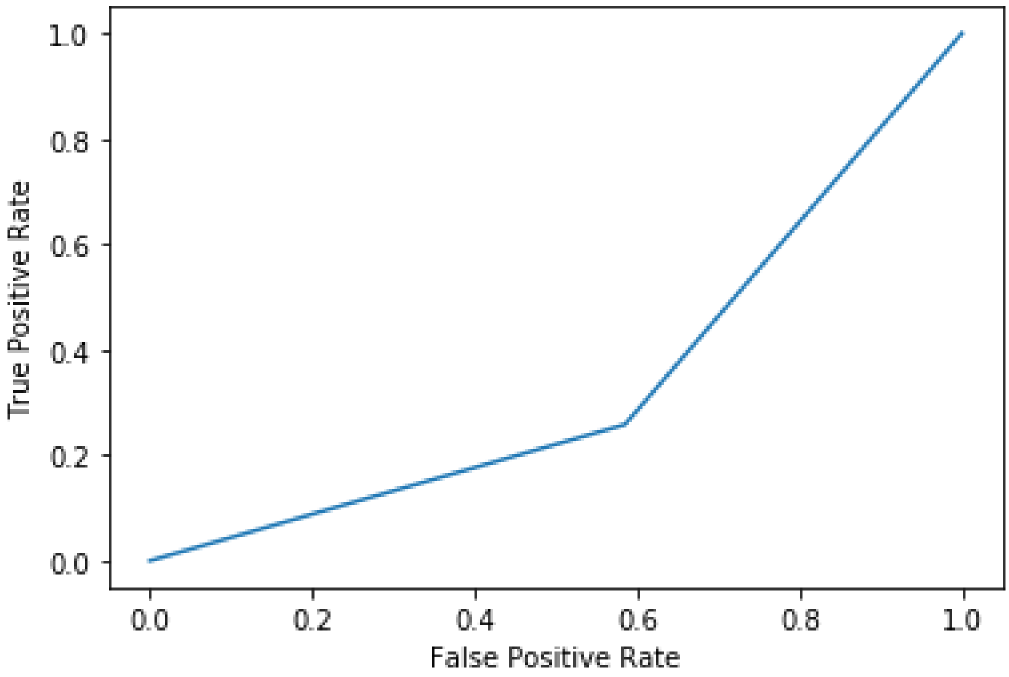
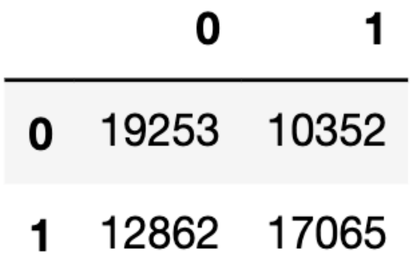


Figure 25

Confusion Matrix for Random Forest on Full Dataset



	0	1
0	19253	10352
1	12862	17065

LOGISTIC REGRESSION

With the chosen C value of 0.001, solver as 'sag' and penalty as 'l2', the model achieved an accuracy of 0.52. Since the accuracy was very less, we tried training the model with the 'l1' regularization parameter, but it showed us a decrease in the accuracy. The ROC curve and confusion Matrix of the best model is shown in Fig.26. and Fig.27. respectively.

Figure 26

ROC Curve for the Dataset Using Logistic Regression

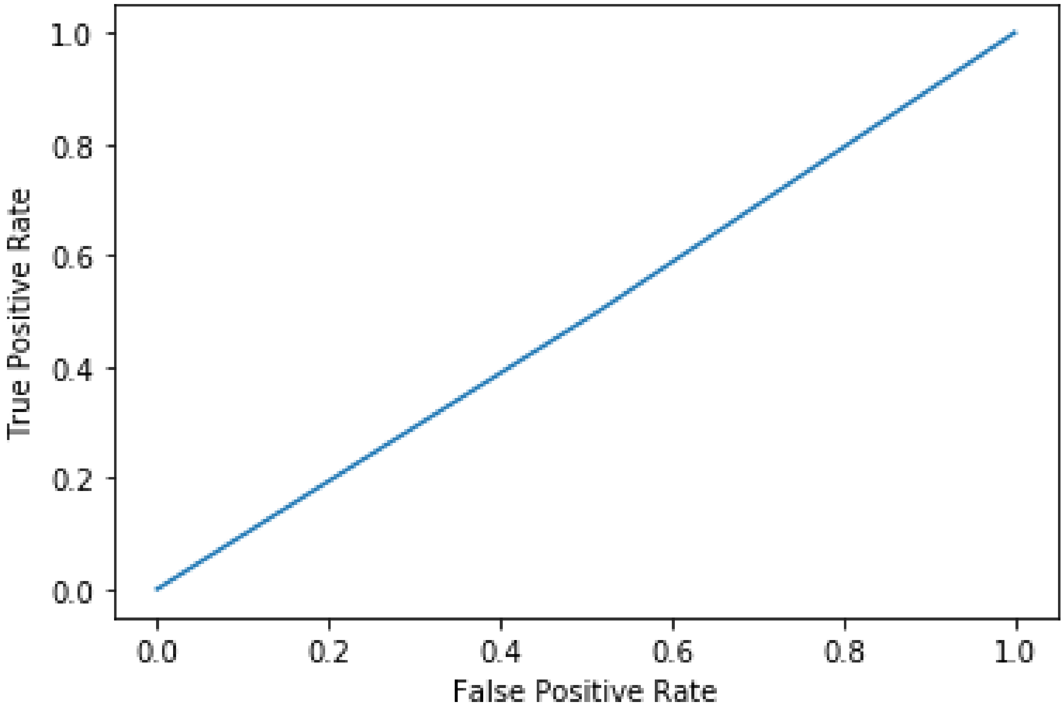
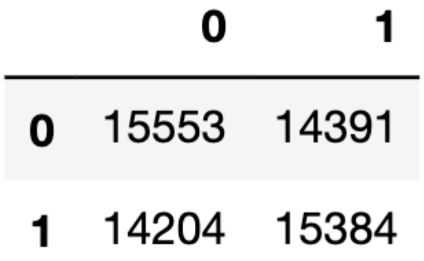


Figure 27

Confusion Matrix for Logistic Regression



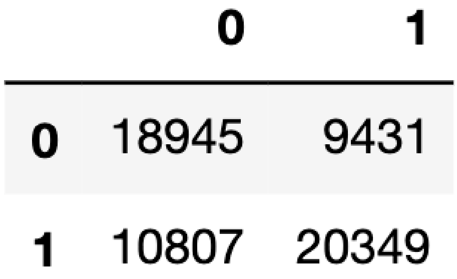
	0	1
0	15553	14391
1	14204	15384

SUPPORT VECTOR MACHINE

Using a C value of 10.0 and Radial Basis (RBF) kernel function, the full data set produced an accuracy of about 0.66. As the C value increase model becomes more costly to train and predict. This is because a larger amount of support vectors is required to be stored. Hence, for C=10.0 we considered only 60% of training data around 180000 samples. The Confusion Matrix and ROC curve are shown in Fig.28. and Fig.29 respectively.

Figure 28

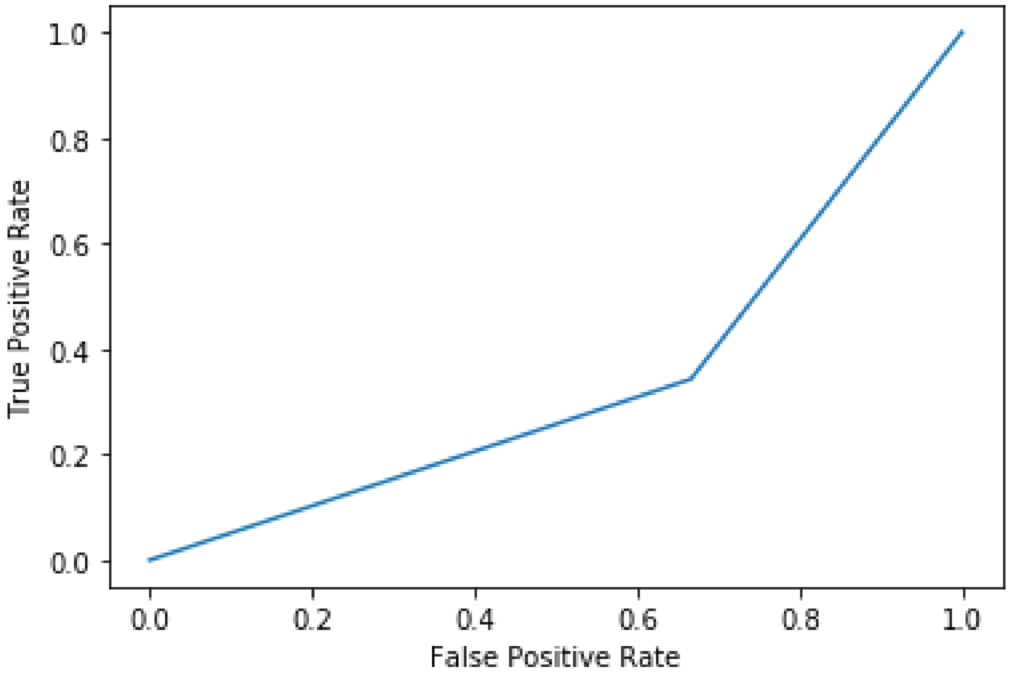
Confusion Matrix for SVM on Subset of Full Dataset



	0	1
0	18945	9431
1	10807	20349

Figure 29

ROC Curve for Full Dataset Using SVM



DEEP NEURAL NETWORK

A deep neural network with two hidden layers gave a good result with accuracy 0.71 on the test data. The model converges much faster than SVM, making it a better choice considering the small accuracy trade-off for the large difference in execution time. The ROC curve for neural networks is given in Fig.30. Comparing the area under the ROC curve for different classes shows that the classification is better for some classes as compared to others.

Figure 30

ROC Curve Using Deep Learning Classifier

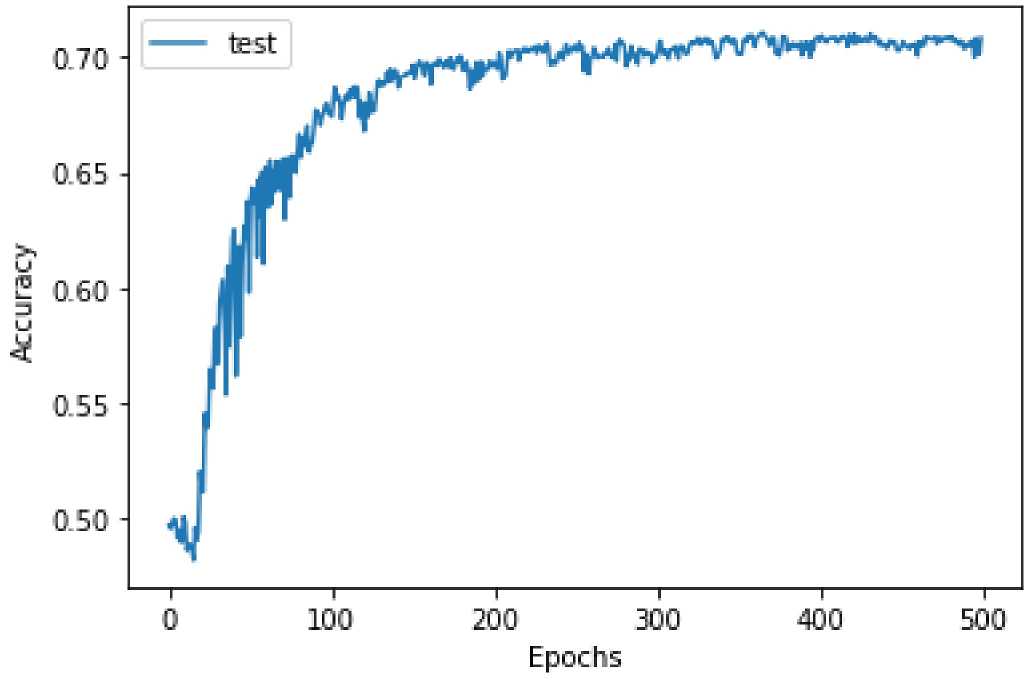
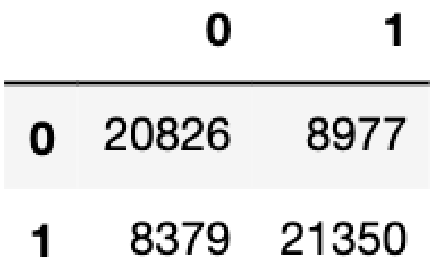


Figure 31

Confusion Matrix for Deep Learning Classifier



	0	1
0	20826	8977
1	8379	21350

CONCLUSION

All the models produced an average result ranging from 0.60 to 0.72 except logistic regression. The decision tree classifier performed the same for both training and test data. After experimenting further, we conclude that this is because the test data is highly representative of the training dataset. The decision tree misses information, when it groups features with continuous values. This might be another reason for low accuracy. Further, the model is only as good as the number of nodes in the tree. Hence, the low accuracy can be improved by having large dataset with more features.

The plan to decrease overfitting by ensemble techniques like random forest and Adaboost didn't improve the accuracy. With just 10 subjects and 4 features, we have a comfortable training sample wherein even decision tree may get low accuracy. If the accuracy of trees is less, then all sub models in the ensemble returns nearly the same accuracy and the predictions of these models will be lesser than or equivalent to that of each single tree. Thus, not only will the overall performance be the same, it will have the same number of data that are predicted correctly and wrongly, respectively. In this project, the Adaboost model performed slightly better than the Random Forest.

With 10 subjects the probabilities of belonging to one of the two target classes will be based on very few observations which most likely will lead to huge fluctuations in out predictions (for example extreme overfitting). Thus, it is really not surprising to see that naiver models perform well here because a random forest in general demands a enormous amount of observations to function admirably.

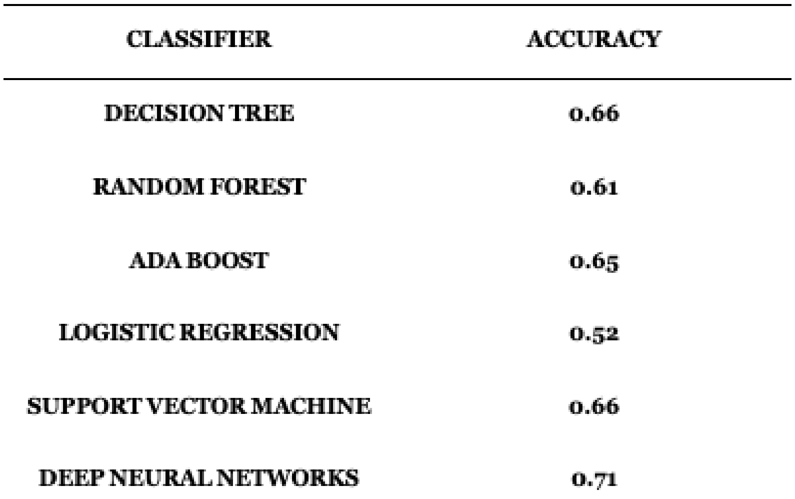
Logistic regression showed the worst performance when compared with the other models. The low accuracy in the case of logistic regression could be due to preprocessing which removed some of the essential data that might be helpful for the prediction of the model. We verified this by training the model on non-preprocessed data and found that the model with non-preprocessed data gave better accuracy than that of the preprocessed data. Though we got better results, the non-preprocessed data is bounded with noise and unclean data hence using it to train a machine learning model is not a good practice.

On the other hand, SVM provided fairly better accuracy but it required very large execution time. They are notable for their efficiency in high dimensional spaces, where the quantity of features is more than the quantity of observations. Having extra features would have given some discriminate power on the training data. SVM will utilize these features to get a handle on some unreal pattern, increasing its generalization error. Furthermore, in the event that we include numerous features, then the likelihood that these extra features collectively happen to characterize the two target classes will be high. Also, if we had a lot of subjects, then the probability that a feature happens to distinguish the two classes will be high.

Deep learning model exhibited the best accuracy when compared to the other traditional machine learning models. The reason could be that deep learning models attempts to acquire high-level features from data in an incremental approach. This reduces the necessity of domain knowledge and extreme feature extraction. When there is absence of domain knowledge for feature contemplation, deep Learning methods outperforms other traditional models as you have to be less concerned about feature engineering. The accuracy of all accuracy is shown in Table. 1.

Table 1

Test Results for all Methods



CLASSIFIER	ACCURACY
DECISION TREE	0.66
RANDOM FOREST	0.61
ADA BOOST	0.65
LOGISTIC REGRESSION	0.52
SUPPORT VECTOR MACHINE	0.66
DEEP NEURAL NETWORKS	0.71

FUTURE WORKS

In the future, we plan to explore the performance of the same machine learning models used in this project with large number of patients each with more features. We would like to explore more complicated machine learning methods to investigate the further scope for improvement. Specifically, we would like to implement CNN (Convolution Neural Network), K-Nearest Neighbors Classifier, and SGD Classifier as it could provide similar or improved performance with considerably less memory usage. Apart from this, we would also like to test the performance of our models on a different dataset collected from many Alzheimer’s patients through the Palo Alto Medical hospital in California.

REFERENCES

[1] Bokde AL, Lopez-Bayo P, Meindl T, Pechler S, Born C, et al. (2006) Functional connectivity of the fusiform gyrus during a face-matching task in subjects with mild cognitive impairment. *Brain* 129: 1113–1124.

[2] Petersen R. C. (2000b) Mild cognitive impairment: transition between aging and Alzheimer's disease. *Neurologia* 15, 93– 101.DOI: 10.1159/000020983.

[3] Farlow, M. R. (2009). Treatment of mild cognitive impairment (MCI). Curr. Alzheimer Res. 6, 362–367. doi: 10.2174/156720509788929282.

[4] Misra C, Fan Y, Davatzikos C (2009) Baseline and longitudinal patterns of brain atrophy in MCI patients, and their use in prediction of short-term conversion to AD: results from ADNI. *NeuroImage* 44: 1415–1422.

[5] Eskildsen, S. F., Coupé, P., & García-Lorenzo, D. (2013). Prediction of Alzheimer's disease in subjects with mild cognitive impairment from the ADNI cohort using patterns of cortical thinning. *NeuroImage* 65, 511-521.

[6] Khazaee, A., Ebrahimzadeh, A., & Babajani-Feremi, A. (2015). Identifying patients with Alzheimer’s disease using resting-state fMRI and graph theory. *Clinical Neurophysiology*, *126*(11), 2132–2141. doi: 10.1016/j.clinph.2015.02.060.

[7] Supekar, K., Menon, V., & Rubin, D. (2008). Network Analysis of Intrinsic Functional Brain Connectivity in Alzheimer’s Disease. *PLoS Computational Biology*, *4*(6). doi: 10.1371/journal.pcbi.1000100.

[8] Armstrong, CC., Moody, TD., Feusner, JD., McCracken, JT., Chang, S., Levitt, JG., Piacentini, JC., & O'Neill, J., (2016) Graph-theoretical analysis of resting-state fMRI in pediatric obsessive-compulsive disorder. *Journal of Affective Disorders* 193, 175–184.

[9] Mascali, D., DiNuzzo, M., Gili, T., Moraschi, M., Fratini, M., & Maraviglia, B., (2015) Resting-state fMRI in dementia patients. *Harvard Dataverse*. Retrieved from April 23, 2020, from http://doi.org/10.7910/DVN/29352.

[10] Mckhann, G., Drachman, D., Folstein, M., Katzman, R., Price, D., & Stadlan, E. M. (1984). Clinical diagnosis of Alzheimer's disease: Report of the NINCDS-ADRDA Work Group* under the auspices of Department of Health and Human Services Task Force on Alzheimer's Disease. *Neurology, 34(7)*, 939–939. doi: 10.1212/wnl.34.7.939.

[11] Petersen, RC., Doody, R., Kurz, A., Mohs, RC., Morris, JC, Rabins, PV., Ritchie, K., Rossor, M., Thal, L., Winblad, B., (2001). Current concepts in mild cognitive impairment. *Archives of Neurology, 58(12*), 1985-1992.

[12] Friston, K. J., Williams, S., Howard, R., Frackowiak, R. S., & Turner, R. (1996). Movement-related effects in fMRI time-series. *Magnetic resonance in medicine*, *35*(3), 346-355.

[13] Behzadi, Y., Restom, K., Liau, J., & Liu, T. T. (2007). A component-based noise correction method (CompCor) for BOLD and perfusion-based fMRI. *Neuroimage*, *37*(1), 90-101.

[14] Power, J. D., Mitra, A., Laumann, T. O., Snyder, A. Z., Schlaggar, B. L., & Petersen, S. E. (2014). Methods to detect, characterize, and remove motion artifact in resting state fMRI. *Neuroimage*, *84*, 320-341.