

Spring 5-21-2020

## **Probabilistic and Machine Learning Enhancement to CONN Toolbox**

Gayathri Hanuma Ravali Kuppachi

Follow this and additional works at: [https://scholarworks.sjsu.edu/etd\\_projects](https://scholarworks.sjsu.edu/etd_projects)



Part of the [Artificial Intelligence and Robotics Commons](#), and the [Other Computer Sciences Commons](#)

---

Probabilistic and Machine Learning Enhancement to CONN Toolbox

A Project Presented To  
The Faculty of Department of Computer Science  
San José State University

In Partial Fulfillment  
Of the Requirements for the Degree  
Master of Science

By  
Gayathri Hanuma Ravali Kuppachi

April 2020

© 2020

Gayathri Hanuma Ravali Kuppachi

ALL RIGHTS RESERVED

## **ACKNOWLEDGEMENTS**

I would like to offer my sincerest thanks to Prof. Leonard Wesley for his appropriate direction and coordinated effort all through the span of my project. I believe myself to be amazingly blessed to have had a chance to work with somebody as splendid as him.

I would additionally like to thank my committee members Dr. Robert Chun and Yijie Sui for guiding and mentoring me with their significant feedback.

## ABSTRACT

Clinical depression is a state of mind where the person suffers from persevering and overpowering sorrow. Existing examinations have exhibited that the course of action of arrangement in the brain of patients with clinical depression has a weird framework topology structure. In the earlier decade, resting-state images of the brain have been under the radar a. Specifically, the topological relationship of the brain aligned with graph hypothesis has discovered a strong connection in patients experiencing clinical depression. However, the systems to break down brain networks still have a couple of issues to be unwound. This paper attempts to give a machine-learning answer for the graph-based brain network investigations of resting state graphs analysis. This model attempts to determine a cost function for a given pair of nodes in the brain to check whether they are connected or not. It can be utilized by medicinal experts to treat patients experiencing clinical depression which helps in decision making of whether a node can be hit directly or not to cure the patient.

***Index terms: Resting state, functional connectivity, graph theory analysis, clinical depression***

## TABLE OF CONTENTS

1. INTRODUCTION .....	6
2. BACKGROUND .....	7
3. APPROACH AND METHOD .....	9
3.1 CONN .....	9
3.2 LOGISTIC REGRESSION.....	11
3.4 ARTIFICIAL NEURAL NETWORKS.....	15
4. EXPERIMENTAL METHOD.....	18
4.1 ENVIRONMENT .....	18
4.2 DATASET .....	18
4.3 FLOWCHART.....	22
5. EVALUATION OF RESULTS .....	38
6. CONCLUSION AND DISCUSSION .....	52
7. FUTURE WORK.....	52
LITERATURE REFERENCES.....	53

## 1. INTRODUCTION

Clinical depression is a weakening mental issue portrayed by unavoidable discouraged disposition, anhedonia, intellectual incapacity, and self-destructive propensity with death rates of 4%, influencing a wide range of population nearly in the millions across the world consistently [1], and it has a high pace of repeat, which causes growing social and monetary concerns [2].

As per the National Institute of Mental Health (NIMH), roughly 17.3 million grown-ups in the U.S. experienced at any rate one scene of clinical depression in 2017. This number likens to 7.1% of all U.S. grown-ups.

People suffering from clinical depression face symptoms like extreme sadness, insomnia, loss of appetite and irritability. It may also cause some physical ailments like body pains and headaches [3]. Individuals with clinical depression experience particular periods enduring at least fourteen days of trouble, low vitality, and loss of enthusiasm for things that they once delighted in. Individuals here and there allude to these periods as painful episodes. The experience of depression can change fundamentally among people. For instance, it might make a few people sleep in and others to rest practically nothing. Each sort of gloom can have particular side effects and impacts [4].

While past examinations have explored both basic and practical variations from the norm, depression is progressively comprehended as a confusion of abnormal interaction between various brain networks and regions [4]. In such manner, the examination of the atypical mind system may move our perception of the turmoil.

Resting-state functional connectivity analysis is a particularly valuable methodology since it tends to be utilized effectively in the various patient population to evaluate functional network availability as far as corresponded, unconstrained MR signal vacillations [5][6]. These examinations raise the intriguing possibility that functional analysis of the system could be used to recognize novel subtypes of depression with more grounded neurobiological associations that predict treatment responsiveness. [7][8].

## 2. BACKGROUND

Clinical depression is a genuine mental ailment described by the discouraged state of mind, and depression patients demonstrated loss of intrigue and joy, sleep deprivation, fixation challenges, weariness, and sentiments of uselessness. Although much efforts have been made in the treatments of depression, up to 80% of patients still suffer from a relapse. Past research has announced that people with clinical depression show decreased social reactivity and adjusted mind actuation in the anterior cingulate cortex (ACC), amygdala, and insula while performing goal-directed tasks. Research also suggests there is a direct proportionality between depression and the rest state. This is examined using resting state functional connectivity. Resting-state functional magnetic resonance imaging (R-fMRI) proportions of brain activity have pulled in extensive consideration. Considering changes in the blood oxygen level-subordinate sign, it offers a novel technique to review the mind's unconstrained development. In particular, the topological association of the brain network has been as of late concentrated with graph hypothesis. Existing investigations have demonstrated the composition of brain networks in patients with clinical depression has a strange system topology structure.

The human mind is sorted out into a perplexing system to satisfactorily process the compromise and detachment of information. Graph theory gives an amazing numerical structure to portraying the topological association of functional networks [12]. Despite the fact that past assessments have displayed disturbed system properties, including neighborhood and worldwide effectiveness, node degree, consonant centrality and node betweenness in sorrow patients [13], the results are much of the time clashing or in any event, restricting. These clashing revelations may be to a limited extent due to node choice or the grouped assortment of patient subtypes.

Various examinations have discovered contrasts in rest state FC in default mode organize identified with a self-referential emotional guideline, the central network associated with consideration and working memory, and other cortical or subcortical areas including basal ganglion, visual cortex, and cerebellum [11]. In any case, a large portion of these investigations accepted that practical availability is static all through



the whole output time frame and accordingly utilized the whole-time course to ascertain useful network. Such a methodology disregards the plausibility of various mental activities happening at various times focuses on time. Continuous assessments have found reoccurring network patterns among traditional frameworks in different disorders including schizophrenia and bipolar issue, which can't be recognized in static system examination [12]. Other examinations have exhibited that rest state of brain functional connections is profoundly dynamic. Past research has moreover demonstrated variation in disordered functional network properties in clinical depression [13]. Although dynamic graph-based networks have been a very useful and promising discovery in the field of cure of depression, there haven't been very strong studies that focus extensively on their relationships and this still looks like a big gap that needs to be bridged.

### 3. APPROACH AND METHOD

Given the absence of appropriate inspections and approaches that focus exclusively on dynamic graph-based brain system analysis using R-fMRI, the purpose of this project is to present a simple but efficient way to analyze dynamic graph-based networks with machine learning. This project will help provide an elegant machine learning solution to analyze R-fMRI images with dynamic graph theory which will help neurosurgeons to treat clinical depression. Although there have been static approaches of graph theory where cost function has been fixed, this approach may not work for a diverse subset of patients. There have also been previous works involving resting state graphs but a dynamic solution with a range of cost functions on resting state graphs with a machine learning approach has not been done before. This is going to be a new solution which in turn can go as an enhancement or an extension to the CONN toolbox which will be useful for a larger community. This section talks in detail about the basics of various methodologies used in the project.

#### 3.1 CONN

CONN is a Matlab-based cross-stage programming for the calculation, show, and examination of a useful network in fMRI (fcMRI). Availability measures incorporate seed-to-voxel network maps, ROI-to-ROI availability frameworks, chart properties of network systems, summed up psychophysiological communication models (gPPI), natural availability, neighborhood relationship and other voxel-to-voxel measures, free segment investigations (ICA), and dynamic segment examinations (Dyn-ICA). [14][15]

It is accessible for rest state information. It covers the whole pipeline from crude fMRI information to theory testing, including spatial coregistration, ART-based scouring, aCompCor methodology for control of physiological and development puzzles, first-level availability estimation, and second-level irregular impact examinations and theory testing.[14][15]

CONN can be installed by downloading and unzipping the conn.zip file which is available open-source for users to download. Minimum requirements for the toolbox are MATLAB R2008 or above with no additional toolboxes needed. Then path needs to be appended to the

appropriate MATLAB directory path. You can launch the tool by typing conn in the MATLAB window. To perform network examinations utilizing this tool stash one will require:

### *FUNCTIONAL DATA*

Either resting state or assignment tasks can be broken down. [14][15]

### *STRUCTURAL DATA*

At least one anatomical volume for each subject (this is utilized for the most part for plotting purposes yet additionally to determine the dim/white/CSF covers utilized in the aCompCor confusion removal technique) [14]

### *ROI DEFINITIONS*

A progression of documents characterizing seeds of defined ROIs can be characterized by veil pictures, content documents characterizing a rundown of MNI positions, or numerous name pictures. The tool stash additionally gives a progression of default pre-characterized districts of intrigue that that will be stacked consequently. These incorporate a progression of seed zones helpful for exploring resting state availability – districts portraying DMN, dorsal consideration arrange, official control organize, and so forth.- , just as a total mind parcellation including 91 cortical regions and 15 subcortical zones from the FSL Harvard-Oxford Atlas just as 26 cerebral territories from the AAL chartbook. The tool employs four steps or stages to process all rfMRI images.[15]

### *SETUP*

Defines fundamental analysis data, information areas, districts of intrigue (seeds), transient covariates, and second-level models. Alternatively, perform useful and anatomical preprocessing steps if fundamental, including realignment, cut planning rectification, coregistration/standardization, division, exception distinguishing proof, and smoothing. [14]

### *DENOISING*

Define, investigate, and expel conceivable puzzles in the BOLD sign, including movement, physiological and other clamor sources [14]

*ANALYSES*

Perform first-level examinations. Characterize the seeds of intrigue and investigate the useful network of various sources independently for each subject. Characterize ICA, voxel-to-voxel investigations, dynamic examinations, and so on [15]

*RESULTS*

Perform second-level examinations. Characterize bunch investigations and perform populace level inductions from the subsequent network proportions of every first-level investigation. Every one of these steps can be characterized intuitively utilizing the tool compartment GUI or automatically utilizing contents and conn batch usefulness. Also, when utilizing the GUI, the entirety of the investigations can be performed locally on a PC, or they can be dispersed among different PCs in a cluster. At present CONN supports multiple frameworks like Sun Grid Engine (SGE), Open Grid Scheduler (OGS/GE) and other Grid Engine executions, PBS/Torque and LSF [14][15]

### 3.2 LOGISTIC REGRESSION

Multinomial logistic regression [16] is utilized when the target variable being referred to is ostensible (proportionately all out, implying that it falls into any of a lot of classes that can't be requested in any important manner) and for which there are multiple classifications. A few models would be like:

- a) Which blood classification does an individual have, given the aftereffects of different analytic tests?
- b) Which nation will a firm find an office in, given the attributes of the firm and the different applicant nations?

These are for the most part measurable grouping issues. They all share for all intents and purpose a reliant variable to be anticipated that originates from one of a restricted arrangement of things that can't be definitively requested, just as a lot of autonomous factors (otherwise called highlights, explanators, and so on.), which are utilized to foresee the target variable. Multinomial logistic regression is a specific answer for characterization issues that utilization a direct mix of the watched highlights and some issue explicit parameters to

evaluate the likelihood of every specific estimation of the needy variable. The best estimations of the parameters for a given issue are typically decided from some preparation information (for example a few people for whom both the symptomatic test outcomes and blood classifications are known, or a few instances of realized words being expressed). The principal approach – called additionally one versus rest – is the most straightforward classifier and comes legitimately from the paired logistic regression. The One-versus-all model is incredibly straightforward and is actually what you anticipate of the words: you train a classifier for each class.

For instance, consider a sky object to be characterized can be one of three classes: Star, Galaxy or Quasar [17]. For this model that we are utilizing the first model is trained that says the positive class will be all the Stars and the negative class will be everything else (in our model: cosmic systems and quasars); it is simply attempting to gain proficiency with a classifier that isolates the stars from the other sky objects. Specifically – on the star side – the likelihood that  $y$  is equivalent to a star given the information  $x_i$  and the parameter  $w$  will be more noteworthy than 0.5 and on the opposite side – the rest – the likelihood that  $y$  is equivalent to a star given the information  $x_i$  and the parameters  $w$  will be under 0.5. Next, become familiar with a model for every last one of different cases. In this way, it will be 1 versus for every single one of the classes. This becomes familiar with a model that attempts to think about and separate systems from stars and quasars. Furthermore, in conclusion, there is a model for the quasars forecast. The weights fitted for each model will be unique, as should be obvious by the lines being dissimilar. When these models are prepared, whatever class has the most elevated likelihood, it wins [17].

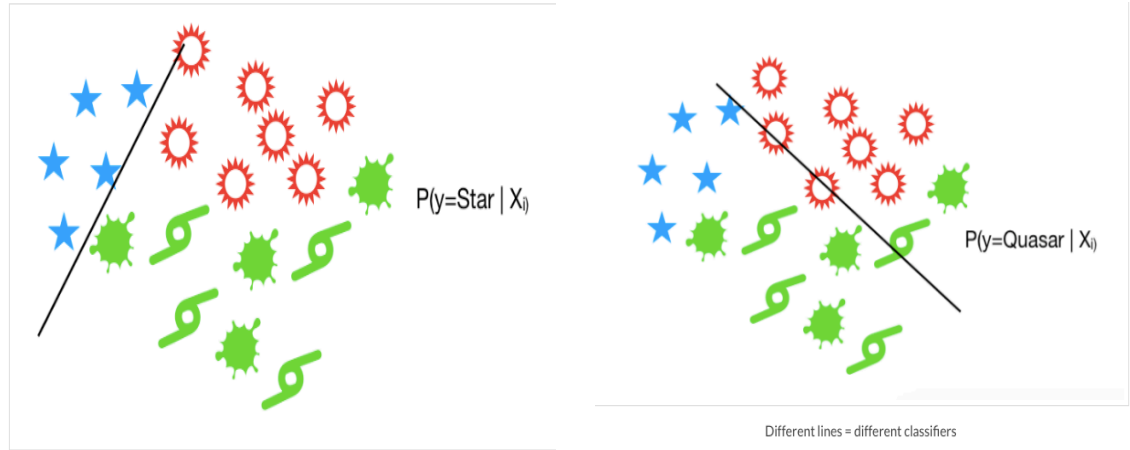


Fig 1. Logistic Regression Multiclass classification

### 3.3 RANDOM FOREST

Random forest is a gathering of decision trees, regularly arranged with the "bagging" strategy. The general idea of the bagging method is that a mix of learning models constructs the general result. Precisely, it is a structure of diverse decision trees and joins them to get an undeniably accurate and stable estimate. [18]

One major significance of this algorithm is that it in general will be used for both classification and regression, which structure the majority of the present AI systems. Below figure is a portrayal of a random forest comprised of two trees:

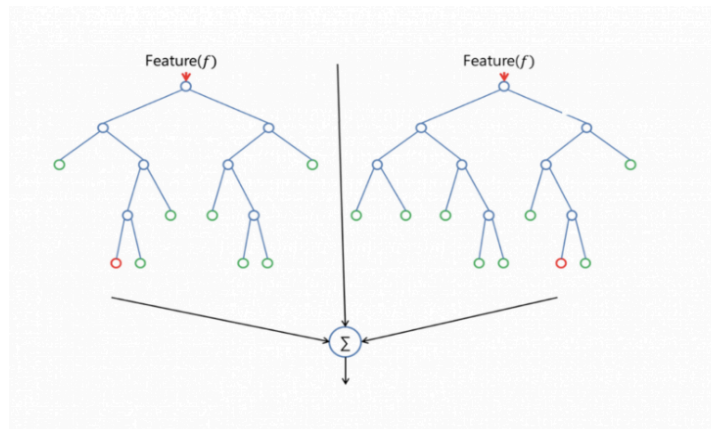


Fig 2. Random Forest

This algorithm adds additional randomness to the model while building up the trees. As opposed to filtering for the most noteworthy component while separating a center point, it searches for the best component among a random subset of trees. These results in a wide fitting assortment that generally realizes a better model. Subsequently, only a random subset of the features is thought about by the figuring for separating a center. You can even make trees continuously random by likewise using random cutoff points for every component rather than examining for the best edges (like an average decision tree does).

Another improbable nature of the algorithm is that it is extraordinarily easy to measure the significance of every part. Sklearn gives an exceptional tool to this that checks the importance of features by looking at how much the tree nodes contribute to lessening impurity influence over all trees in the forest. It enrolls this score normally for every part or node of the tree and scales the results, so the aggregate is equal to one [18].

By looking at the component centrality you can pick which features to possibly drop since they don't contribute enough (or to a great extent nothing in any way shape or form) to the desire method. This is noteworthy in light of the fact that a general rule in AI is that the more features you have the more plausible your model will encounter the evil impacts of overfitting and the opposite way around.

The hyperparameters in the algorithm are either used to grow the perceptive force of the model or to make the model speedier. The following is the viewpoint of how hyperparameters of sklearn work in limit. [18]

There is `n_estimators` hyperparameter, which is just the amount of trees the computation works before taking the most extraordinary majority rule or taking the midpoints of conjectures. A higher number of trees grows the presentation and makes the gauges progressively consistent, yet it in like manner blocks the count. Another huge hyperparameter is `max_features`, which is the most extraordinary number of features random forest thinks about splitting a center point. Sklearn gives a couple of decisions, all delineated in the documentation. The last noteworthy hyperparameter is `min_sample_leaf`. This chooses the base number of leaves required to section an inward center point. [18]

The `n_jobs` hyperparameter mentions to the engine what number of processors it is allowed to use. In the event that it has an estimation of one, it can simply use one

processor. An estimation of "- 1" suggests that there is no limitation. The `random_state` hyperparameter makes the model's yield replicable. The model will reliably make comparative results when it has a positive estimation of `random_state` and in case it has been given a comparative hyperparameter and comparable planning data.

Random forests are also hard to beat execution time. One can apparently reliably find a model that can perform better, like a neural network for example, yet these by and large put aside more exertion to develop, in any case, they can manage an assortment of feature types, like clear cut, paired, and numerical. By and large, the random forest is a (generally) speedy, clear and versatile instrument, anyway not without certain impediments.[18]

### 3.4 ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANN) are multi-layer totally connected neural nets that looks like the figure below. They include an input layer, output layer, and many hidden layers. Every node in a layer has an association with one another node in the accompanying layer. We make the framework progressively significant or more profound by extending the quantity of hidden layers.

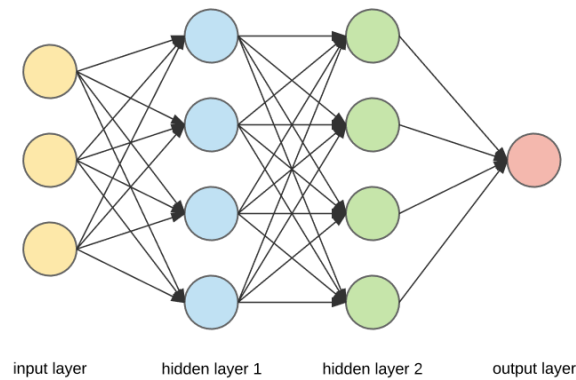


Fig 3. Artificial Neural Network (ANN)

ANN relies upon an arrangement of related networks called artificial neurons, which openly model the neurons in a characteristic psyche. Each affiliation, like the synapses in



an organic psyche, can transmit a sign to various neurons. An artificial neuron that gets a sign by then structures it and can hail neurons related with it.

Here, the "signal" at an affiliation is a veritable number, and the output of each neuron is calculated by a nonlinear boundary of the sum of its input. The affiliations are called edges. Neurons and edges usually have an importance that changes as learning proceeds. The weight increases or decreases the nature of the sign at an affiliation. Neurons may have a breaking point with the ultimate objective that a sign is conferred just if the absolute sign crosses that edge.

Regularly, neurons are gathered into layers. Different layers may perform different changes in their input of data. Signs travel from the principle layer (the data layer) to the last layer (the yield layer), possibly in the intersection the layers on various events. A given neuron takes the weighted aggregate of its data sources and experiences a non-linear activation function. This is the yield of the hub, which by then transforms into the commitment of another input to the accompanying layer. The processing happens from left to right, and the last yield is dictated by playing out this framework for all the neurons. Setting up this significant neural system infers learning the loads identified with all the edges.

Below is a figure of representation of the network with weights.

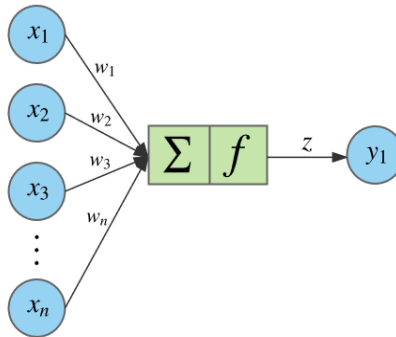


Fig 4. ANN with weighted sums

The condition for a given neuron/node looks as follows. The weighted aggregate of its info sources experiences a non-direct initiation work. It might be addressed as a dab result of vectors, where  $n$  is the quantity of contributions for the hubs.

Bias is a contribution to all the hubs and consistently has the worth of 1. It permits to move of the outcome of the activation function to one side or right. It likewise encourages the model to prepare when all the inputs are 0. Below is the formula used.

$$z = f(b + x \cdot w) = f\left(b + \sum_{i=1}^n x_i w_i\right)$$

$$x \in d_{1 \times n}, w \in d_{n \times 1}, b \in d_{1 \times 1}, z \in d_{1 \times 1}$$

In any case, the first model needs to be prepared to become familiar with the inputs, and the training process fills in as follows. Arbitrarily instate the loads for all the nodes. For each preparation model, play out a forward pass utilizing the present inputs, and ascertain the yield of every node going from left to right. The last input is the estimation of the last node. In contrast the last node and the real target in the training data, and measure the error utilizing a loss function. Perform backward pass from output to left and proliferate the blunder to each node utilizing backpropagation. Compute each weight's contribution and alter the inputs in a like manner utilizing gradient descent. Proliferate the error back beginning from the last layer. Given this introduction about various methodologies, the next sections delve deeper into the design and implementation of how these are used in the current project.

## 4. EXPERIMENTAL METHOD

This section covers the design and implementation of this project. It talks in detail about the method, techniques used and also the various experiments and algorithms applied to the dataset.

### 4.1 ENVIRONMENT

Some of the important tools and technologies used for this project are as follows:

- Python - The programming language used for this project as it has a good compilation of machine learning libraries support which is extensively used throughout building the models. The version used is 3.6
- CONN Toolbox - CONN Toolbox is a MATLAB software that is used to pre-process images of depression patients by analyzing resting state graphs. Version 18b is used.
- Scikit-Learn - Scikit Learn is a popular package of Python which enables many machine learning application programming interface (API) which can directly be used on the dataset.
- Keras and TensorFlow - Keras and TensorFlow packages are used for deep learning machine learning model's usage to train the model.

### 4.2 DATASET

This project was originally intended to work on real patient data obtained from Palo Alto (PA) hospital but due to security clearances and time constraints, this could not be achieved. Hence, an opensource Functional Connectome Project (FCP) dataset has been used for this project. Resting state images of 40 patients from Beijing in the age group between 18 to 26 are extracted and fed into the CONN toolbox. These images represent the

brain of depression patients in the resting state. These images have 33 slices and 225 timepoints captured of the brain cortex region. Two datasets are generated, and their visualization is as follows for experiments.

#### DATASET 1:

In this dataset, cost functions of 0.1, 0.2, 0.3 and 0.4 are used as 4 different classes. They are labeled as Cost 1, Cost 2, Cost 3 and Cost 4 respectively. Four algorithms have been used on this dataset. The size of the dataset is 54k records. Data is evenly distributed.

After standardization, the data looks like below on training data:

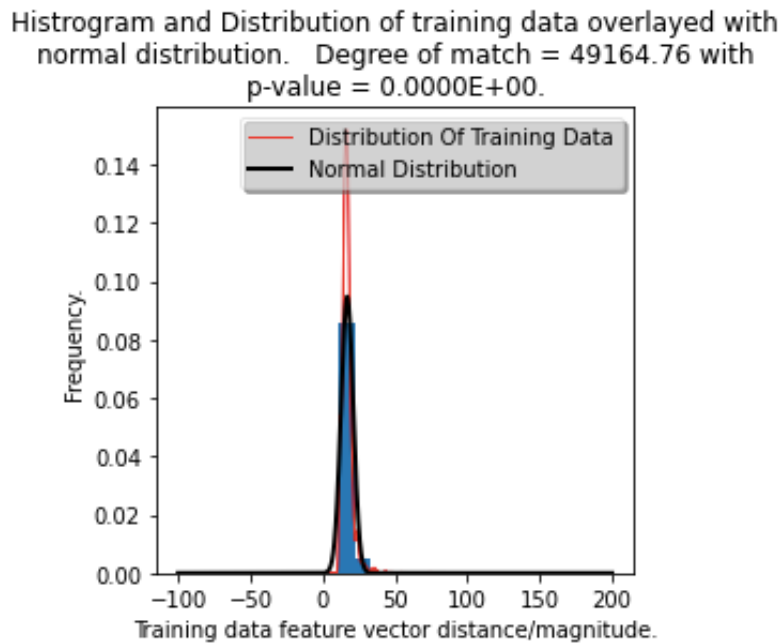


Fig 5. Histogram of Training Data

After standardization, the data looks like below on testing data:

Histogram and Distribution of test data overlayed with normal distribution. Degree of match = 12745.83 with p-value = 0.0000E+00.

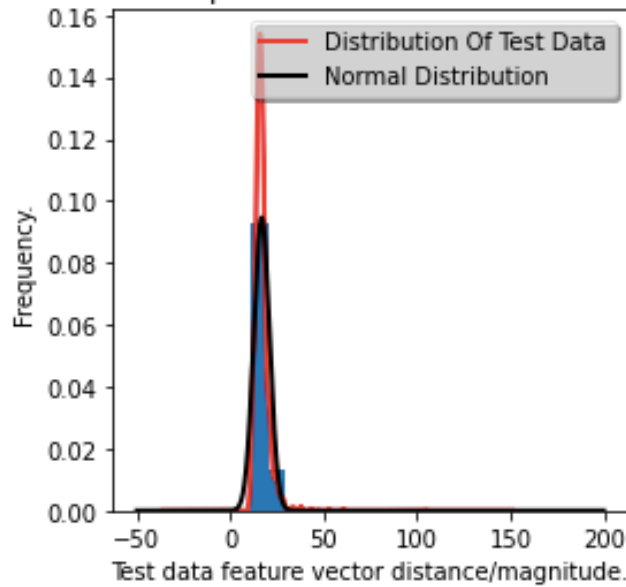


Fig 6. Histogram of Testing data

## DATASET 2:

The cost range is bucketed or equally distributed into 4 classes on bigger data to see how the experiment performs. 4 cost functions are defined where  $[0.1, 0.15, 0.2, 0.25]$ ,  $[0.3, 0.35, 0.4, 0.45]$ ,  $[0.5, 0.55, 0.6, 0.65, 0.7]$ ,  $[0.75, 0.8, 0.85, 0.9]$  are categorized as 4 classes of Costs. The same machine learning algorithms are applied to this dataset and below are the results. After standardization, the data looks like below on training data:

Histogram and Distribution of training data overlayed with normal distribution. Degree of match = 219155.46 with p-value = 0.0000E+00.

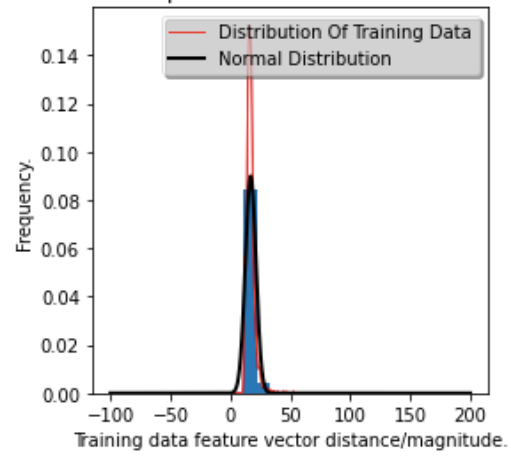


Fig 7. Histogram of Training data

Histogram and Distribution of test data overlayed with normal distribution. Degree of match = 59202.87 with p-value = 0.0000E+00.

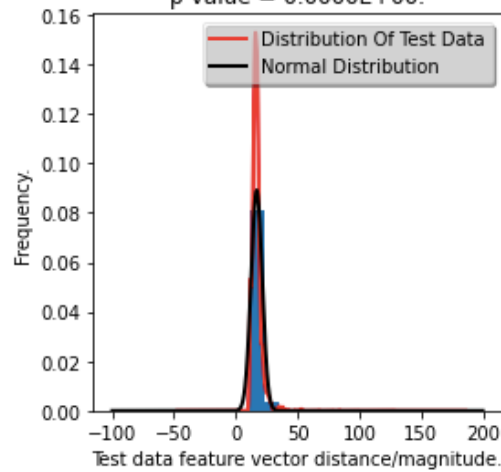


Fig 8. Histogram of Testing data

### 4.3 FLOWCHART

This section talks in detail about the different steps employed in building machine learning. Below is the flowchart for the dataset generated.

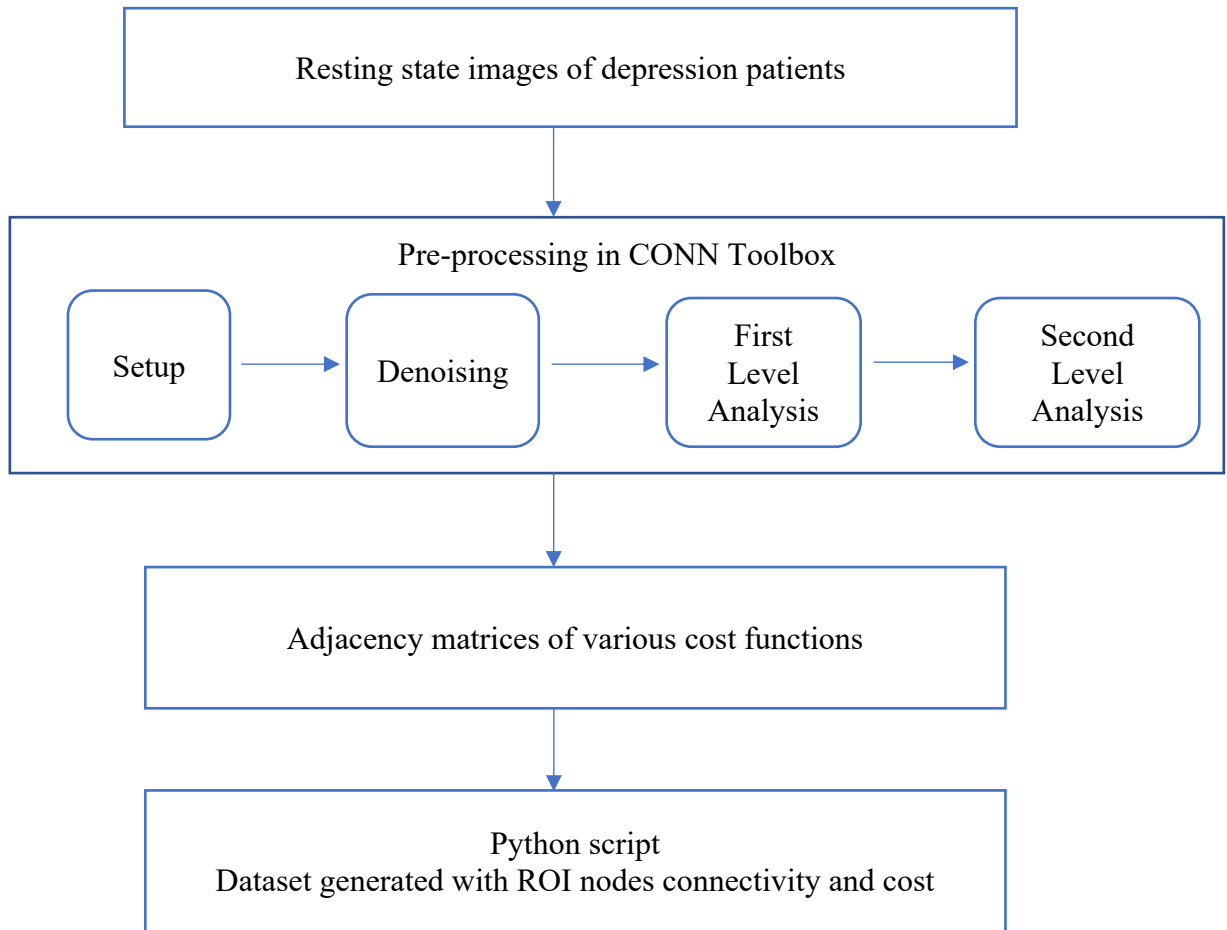


Fig 9. Dataset Generation

Resting state images are sent into the CONN toolbox for pre-processing. It undergoes 4 phases to complete preprocessing which are as follows.

#### *SETUP*

Inside the CONN toolbox, Setup phase undergoes few sub-steps which are as follows:

The basic experimental info setup is where 40 subjects are created and assigned initially by creating a new conn project. Other parameters are optional like the number of sessions runs, repetition time and acquisition type are optional and can be created with default values. The creation of the project and setup is as shown in the figure below.

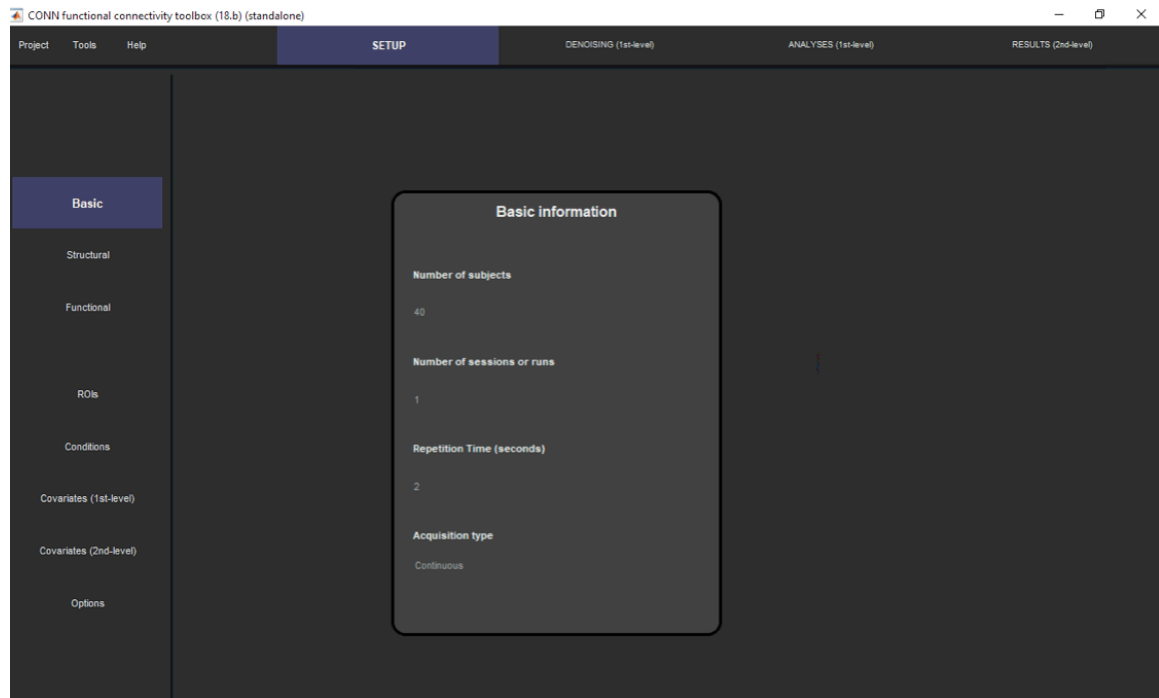


Fig 10. Basic experimental info setup

In the structural file's setup, each subject id is processed sequentially and select the related anatomical picture (one anatomical volume for every subject). On the off chance that you have various anatomical volumes per subject (for example one for every session/scan), select Session-specific structural data and afterward enter the relating session explicit anatomical volumes. Anatomical volumes ought to normally, be coregistered to the practical and ROI volumes for each subject (for example on the off chance that utilizing MNI-space normalized volumes you ought to enter here the standardized anatomical volume). The structural file setup of the image after processing and assignment is as shown in the figure below.



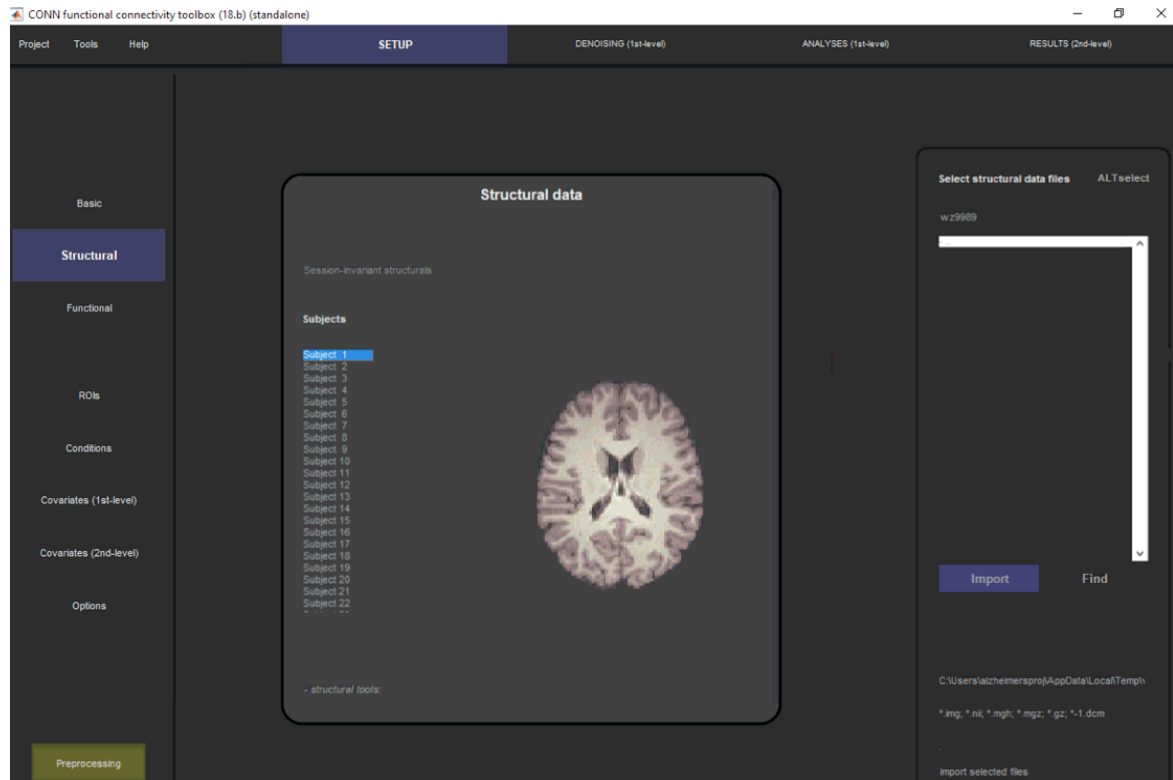


Fig 11. Structural files setup

In the next step, the brain is shown in the "Functional files setup" window shows the first (left) and last (right) examine for the chosen subject/meeting. The functional images are required to be now pre-handled (realigned and smoothed), just as coregistered with the structural and ROI volumes. On the off chance that they are not, you may choose 'functional tools: individual preprocessing step' or 'Preprocessing' to play out the fitting preprocessing steps. This will establish the coregistration between the functional and structural volumes for the chosen subjects/sessions. As a matter, of course, the toolbox will characterize an alternative 'dataset 1' highlighting the unsmoothed volumes (utilizing the SPM-show for unsmoothed volumes, same documents without the underlying 's' in the filename) and this dataset will be the one utilized except if indicated in any case while extricating ROI-level BOLD sign time series. The functional file setup is as shown in the figure below.

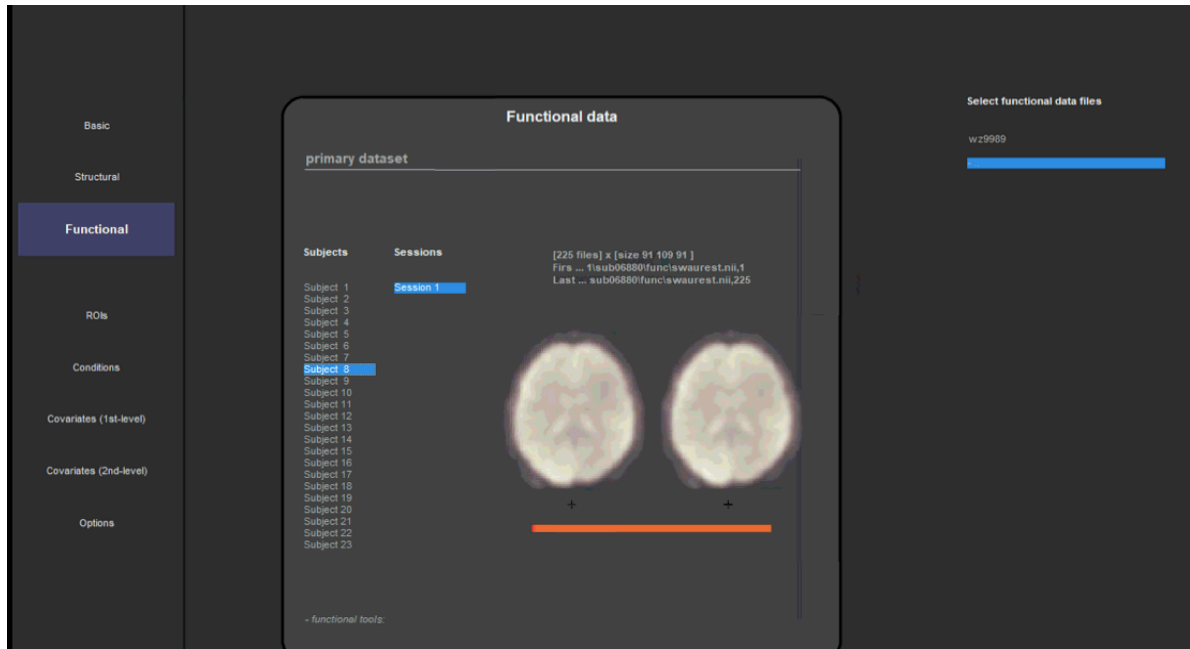


Fig 12. Functional files setup

Next Step is ROI Files Setup where Region of Interest (ROI) files can be assigned independently to each subject (for subject-explicit ROIs) or generally over all subjects (standardized space ROIs). From every ROI, CONN may separate a normal BOLD signal time series inside the ROI voxels (remove normal time-series alternative), numerous time-series from a Principal Component decomposition of the BOLD signal inside ROI voxels (extract Principal Component Analysis (PCA) decay alternative or a weighted aggregate of the BOLD signal inside ROI voxels (weighting each voxel time series by an alternate factor; extract weighted aggregate time series alternative). Additionally, you may show which dataset, among conceivably numerous elective utilitarian datasets, every ROI BOLD signal is to be processed. Running the structural or functional division Preprocessing step will naturally fill in the first three ROIs in this tab (Gray/White/CSF) with the comparing subject-explicit veils. As a matter, of course, all new extends in CONN are set up to extract a Principal Component disintegration from both the White and CSF districts which can later be utilized during the Denoising venture to actualize aCompCor (evacuation of White/CSF noisy parts). ROI Setup is as shown in the Figure below.

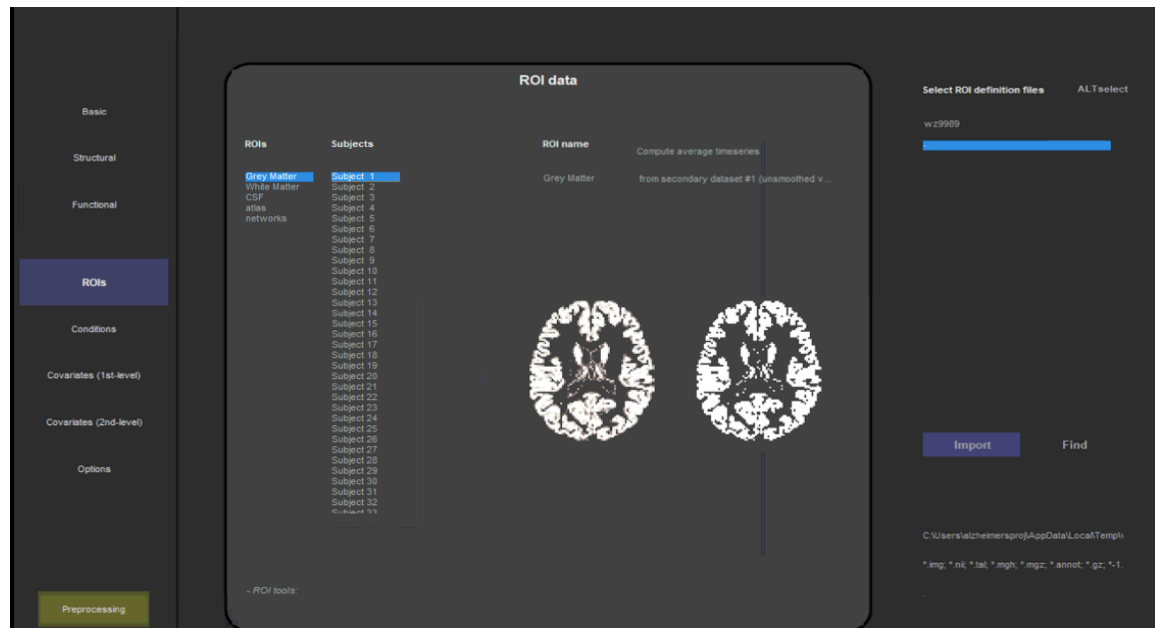


Fig 13. ROI files setup

The other options are set to default values, and the Preprocessing button is clicked. After nearly 3-4 hours, the next step is advanced which is Denoising. The details of this process are described in the next step.

## DENOISING

In this step, linear regression and band-pass sifting are applied to evacuate undesirable movement, physiological, and other artifactual impacts from the BOLD sign previously figuring network measures. As a matter, of course, the framework will begin with three distinct wellsprings of conceivable confounders: 1) BOLD sign from the white issue and CSF covers (5 measurements every); 2) any beforehand characterized inside subject covariate (realignment and cleaning parameters), and 3) the fundamental condition impacts (condition squares convolved). For every one of the chosen conceivable realignment you may change the number of measurements (indicating what number of worldly parts are being utilized), and the subordinate's request (determining what number of progressive requests of worldly subsidiaries are remembered for the model). For

instance, the realignment puzzle (got from the assessed subject movement parameters) is characterized as a matter of course by 6 measurements. One can change the subsidiary request to 1 showing that what's more the first-request transient subordinate of the movement parameters ought to likewise be utilized as covariates. So also, the White Matter confound is characterized of course by 5 measurements and 0 subsidiary requests (showing that 5 PCA fleeting segments are being utilized, with not extra transient subsidiary terms). Naturally and except if indicated in any case every single new venture that utilization the default CONN preprocessing steps will be consequently set up to utilize a blend of aCompCor (White and CSF ROIs, 5 parts each), cleaning (the same number of regressors as recognized invalid sweeps), movement regression (12 regressors: 6 movement parameters + 6 first-request worldly subordinates) and separating in the Denoising step. Also, for task plans the primary impact of assignment (direct BOLD sign changes related to the nearness/nonappearance of an errand) are additionally relapsed out during this progression except if indicated something else. Extra/elective denoising steps can be determined by including/subbing impacts recorded in the 'frustrates' list. Denoising setup is as shown in the figure below.

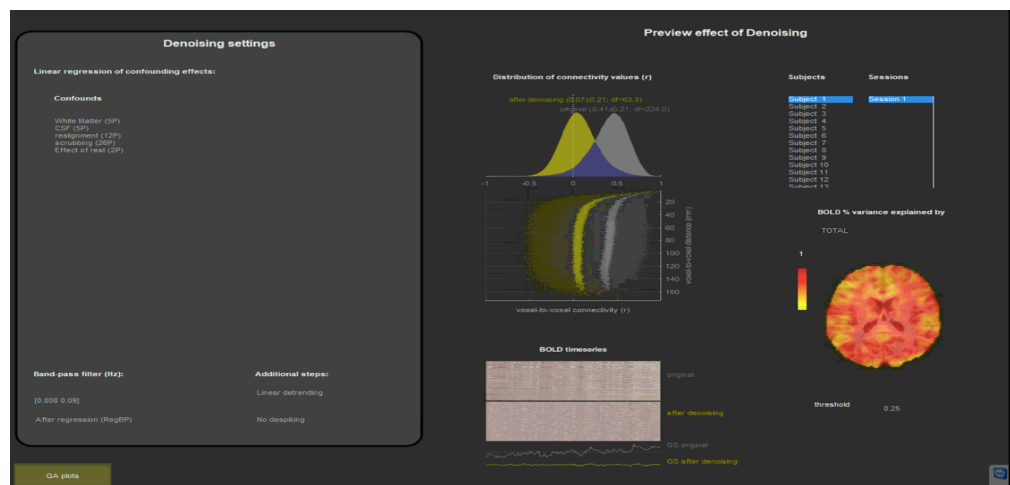


Fig 14. Denoising setup

Once the denoising setup is completed, it progresses to the next step i.e.; first-level analysis which is explained in detail below.

## *FIRST LEVEL ANALYSIS*

The first step is to set up Seed to Voxel and ROI Analyses Phase. In the Analysis type, you may choose the kind of the first-level model to be utilized. The default conduct for useful availability investigations is to utilize a General Linear Model with weights for relapse/connection proportions of the explicit relationship between the BOLD time series seeds and each target ROI BOLD time series. This model is fitting both for resting state just as errand related structures. When utilizing weighted GLM click on Weights to change the general weighting of the outputs inside each condition (as a matter, of course, condition-explicit loads are characterized by the hrf-convolved squares/occasions for each condition). Notwithstanding the standard bivariate relationship measure for functional connectivity network examinations, you may likewise choose relapse measures, just as characterize whether you need to process bivariate measures – investigating singular seed/sources independently, or semi partial/multivariate – where the entirety of the sources/ROIs are entered mutually into the general direct model to assess their extraordinary commitments.

Leave the rest of the values to default.

Select Analyses to describe the sort of assessments to be performed (ROI-to-ROI, seed-to-voxel, or both). Right when portraying/researching the system assessments is finished, press Done. This will play out the portrayed assessments for all subjects, building seed-to-voxel organize maps for each picked source (as well as complete ROI-to-ROI organize structures for these hotspots) for each subject and each condition. First-level results (beta maps and association maps when appropriate) are also exchanged as nifti volumes (one for each Subject/Condition/Source mix) in the results/first level envelope.

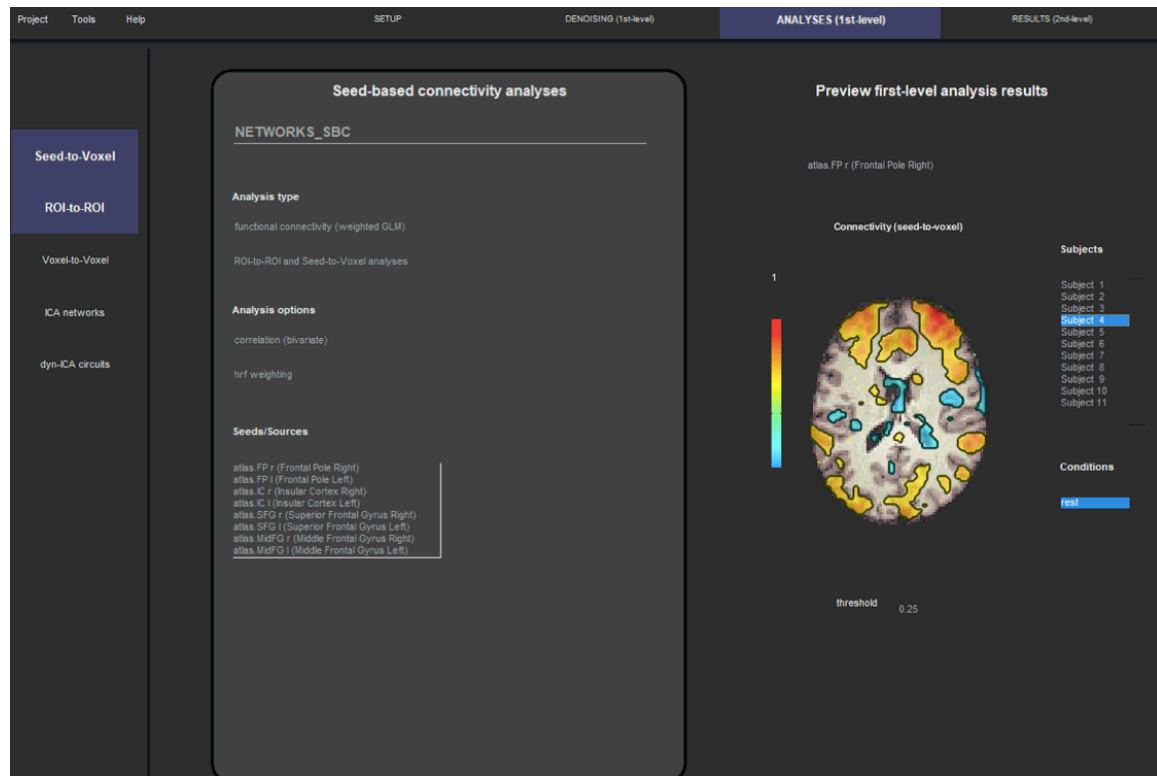


Fig 15. Seed to Voxel and ROI Analyses

The next step is Voxel to Voxel and ICA Analyses phase. Voxel-to-voxel examinations are investigations that consider the whole grid of voxel-to-voxel availability values. These are helpful when you would prefer not to confine your investigations to one or a few from the earlier seeds/ROIs and rather need to examine conceivable network contrasts over the whole cerebrum. The executed investigations/measures are group ICA, several elements, group PCA, group MPVA. At the point when got done with characterizing/investigating the availability examinations press Done. This will play out the characterized examinations for all subjects, assessing the subsequent maps from the voxel-to-voxel network grid for each subject and each condition. First-level outcomes (beta maps) are additionally traded as nifti volumes (one for each Subject/Condition/Measure mix) in the results/first level organizer.

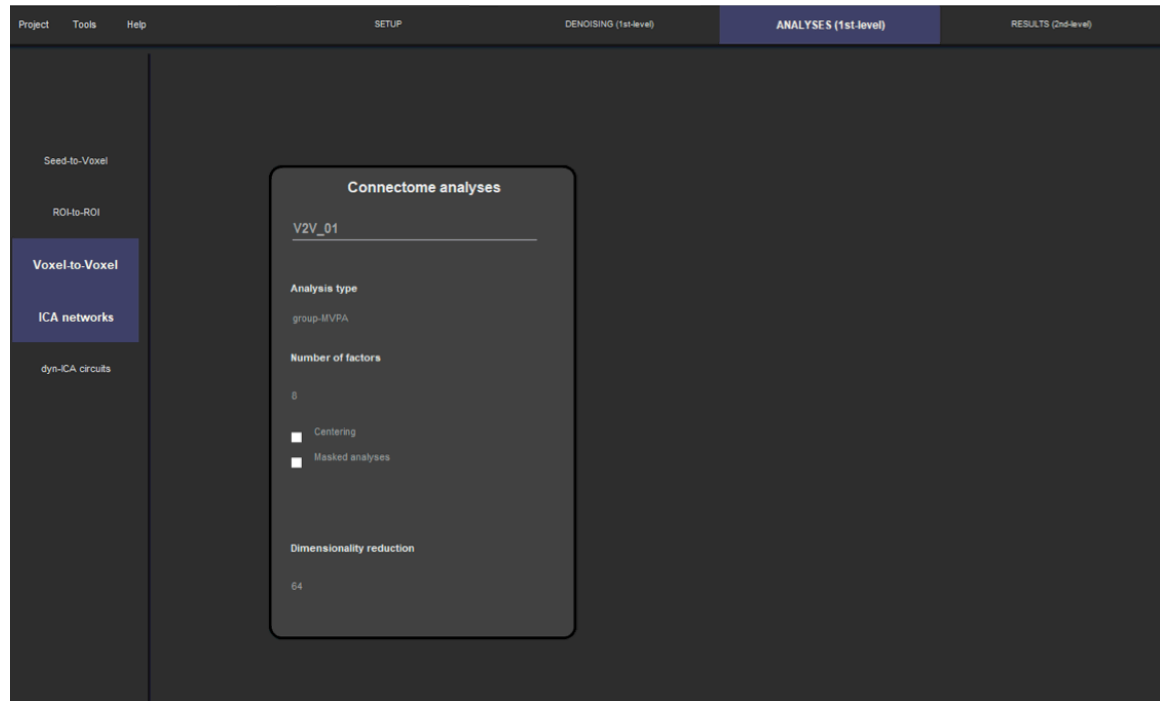


Fig 16. Voxel to Voxel &amp; ICA Analyses

## SECOND LEVEL ANALYSIS

This is the final stage where results and pre-processing are completed. This project is concerned with only ROI-ROI phase which is explained in detail below.

In the ROI-to-ROI Analyses phase, while choosing ROI-to-ROI investigations (from the furthest left tab in the figure), the mind shows at the privilege shows a hub perspective on the return for capital invested to-return for capital invested second-level investigation results evaluated continuously for the chose second-level model and complexities. These outcomes can have a threshold at the ideal p-esteem edge, utilizing uncorrected p-qualities or FDR-remedied p-values, and for a couple sided inductions. The dropdown menu over this figure permits you to alter the rundown of target ROIs being shown (this decision influences the FDR corrected p-estimations of the outcomes; FDR rectification is applied over the arrangement of target ROIs picked). Pictures are shown in the neurological arrangement. Select Display 3d to see a 3d-rendered show of the supra-

edge return on initial capital investment level outcomes. For each target ROI, the rundown at the base of the figure shows the network different impact sizes (between the chose source – or a straight mix of sources-and each focus), just as T/F/X-values, uncorrected p-qualities, and FDR-revised p-values for the predefined second-level examination. Right-tapping on this table permits you to send out this table to a .txt or .csv document. Select Display Values or on the other hand Import Values to show/import the evaluated ROI-to-ROI network esteems for each subject and each chose condition between the chose seed and target ROIs.

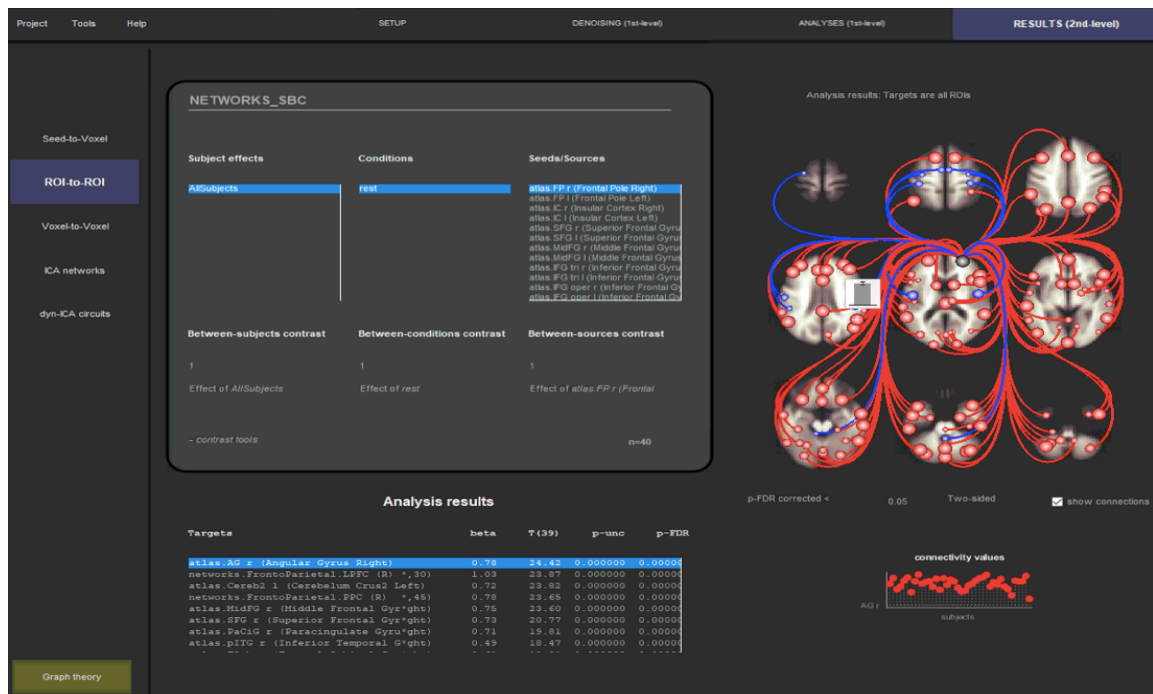


Fig 17. ROI to ROI Analyses

Lastly by clicking diagram hypothesis, a graphical showcase is appeared permitting the clients to test (for the choice among subjects and between-condition contrasts) proportions of proficiency, centrality, and cost/degree, related with an ROI-to-ROI availability arrange. Close on 'System hubs' to restrain the ROI-to-ROI organize broke down to that characterized by a subset of ROIs. The 'System edges' alternative permits the meaning of the availability edge above which two ROIs are viewed as associated, and it very well may be characterized dependent on connection scores, z-scores, or cost esteems. The below figure shows the comparing measure impact



size (worldwide effectiveness, nearby efficacy, or cost), just as T values, uncorrected p-qualities, and FDR-amended p-values for the determined second-level examination. The graphical showcase and second-level results can have a threshold utilizing either uncorrected or FDR corrected p-qualities, and it tends to be set to show uneven or two-sided results. Right-tapping on the mind show shows extra show alternatives, including 3d-rendered perspectives on the, broke down system of the network.

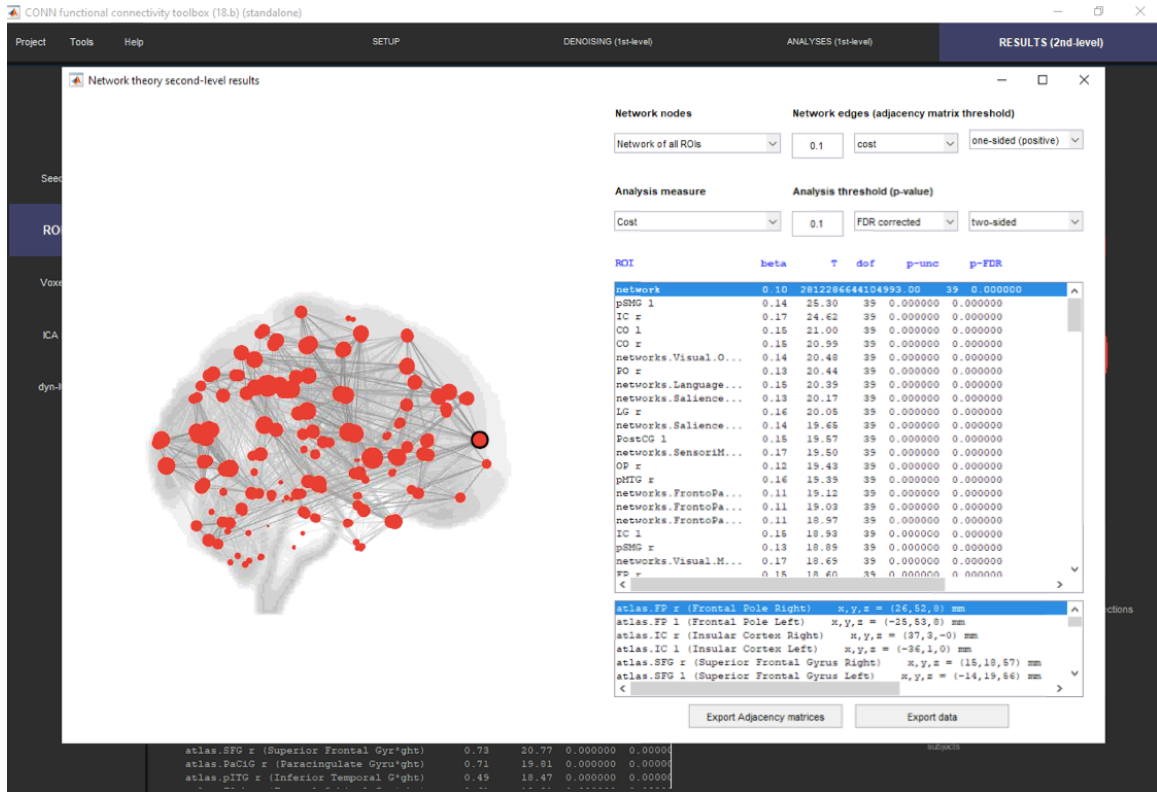


Fig 18. Graph Theory

From the above image, we select 18 cost functions [ 0.1, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.5, 0.55, 0.6, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95] in the following range bracket. We enter them in the textboxes of cost as shown in the above figure. Therefore 18 .mat files with adjacency matrices are exported which look like below.

The output of these pre-processed images is an adjacency matrix generated for different Region of Interests (ROI) of the brain with different costs associated. Based on the target cost functions and ROI, the dataset is constructed based on the adjacency matrix. Since

there is 40 subject's probability/average of two ROI's connected in all 40 subjects is calculated. This is done by the following formula:

$$Probability(A, B) = \frac{\text{Sum}(\text{ROI} - A, \text{ROI} - B)}{\text{no. of subjects}}$$

The above formula is used to compute the probability of every ROI with each other which is extracted from the adjacency matrix. For this dataset, there is 164 ROI's present which has been used and target cost functions are used to form the entire dataset.

These cost functions are used as the target variables. Two sets of datasets are used as part of experiments on this project. Initially, a reduced range of cost functions [0.1, 0.2, 0.3, 0.4] was used as the bigger range was not giving results as expected. This improved the results but was still unsatisfactory. Sample Dataset 1 is shown in Fig. Details will be discussed in the Results section.

	Cost	ROI-A		ROI-B	Probability
51656	0.4	atlas.Putamen r		networks.Salience.ACC	0.85
17446	0.2	Inferior Temporal Gyrus, anterior division Right	Inferior Temporal Gyrus, temporooccipital part...		0.05
36887	0.3	Parietal Operculum Cortex Right		Cerebellum 4 5 Left	0.15
43422	0.4	Superior Temporal Gyrus, posterior division Right	Temporal Occipital Fusiform Cortex Right		0.78
37167	0.3	Planum Polare Left		networks.Salience.RPFC	0.30
31072	0.3	Inferior Temporal Gyrus, anterior division Right		Vermis 6	0.08
40324	0.3	networks.Salience.SMG	networks.FrontoParietal.LPFC		0.22
34115	0.3	Subcallosal Cortex	Temporal Fusiform Cortex, posterior division Left		0.55
40973	0.4	Insular Cortex Right	Parahippocampal Gyrus, anterior division Left		0.15
49117	0.4	Lingual Gyrus Right	Parietal Operculum Cortex Right		0.35

Fig 19. Sample Dataset 1

The second dataset employed a strategy of bucketing the cost functions. 18 Cost functions mentioned above are bucketed into 4 classes by equally distributing the above costs into these 4 buckets [0, 1, 2, 3]. Sample Dataset 2 is shown in below Fig.

	Cost	ROI-A	ROI-B	Probability
197239	3.0	Cuneal Cortex Left	Central Opercular Cortex Left	0.875
189400	3.0	networks.Salience.AInsula	networks.Salience.AInsula	0.000
136327	2.0	Middle Frontal Gyrus Right	Temporal Occipital Fusiform Cortex Left	0.375
146135	2.0	atlas.Thalamus I	Cerebelum 8 Right	0.825
168843	2.0	Intracalcarine Cortex Right	Temporal Fusiform Cortex, anterior division Left	0.850
12207	0.0	Cerebelum 6 Right	networks.DefaultMode.LP	0.000
135902	2.0	Insular Cortex Left	Cerebelum 9 Right	0.100
125376	2.0	Middle Temporal Gyrus, posterior division Left	networks.DefaultMode.PCC	0.750
203433	3.0	Insular Cortex Right	networks.Cerebellar.Anterior	0.850
149783	2.0	Superior Frontal Gyrus Left	networks.FrontoParietal.PPC	1.000

Fig 20. Sample Dataset 2

This ends the phase of dataset generation and creation. The next phase is applying machine learning models to this dataset. The steps are as follows which can be represented using a flowchart.

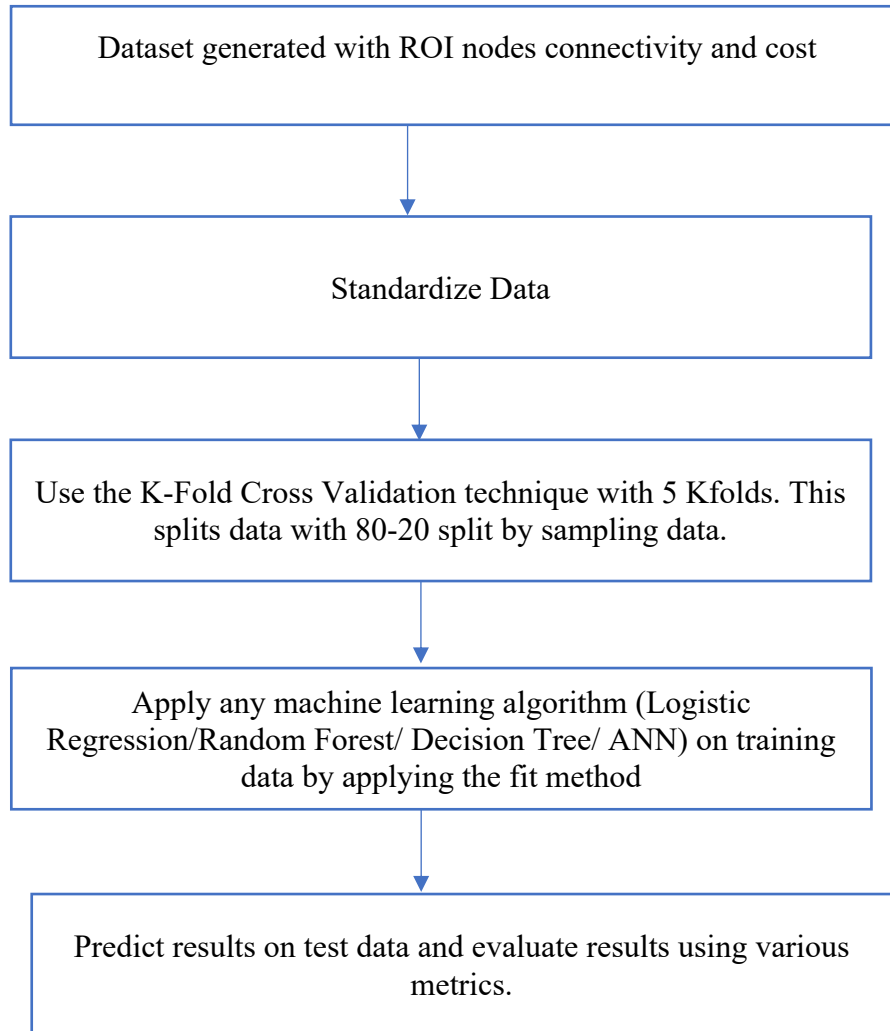


Fig 21. Model Training and Prediction

Using the above basic hierarchy or flow, all machine learning models are trained and tested on both the datasets. Steps are described in detail below:

### 1) STANDARDIZE DATA:

Sklearn offers packages to standardize or normalize data. This step is mandatory so that there can be good results achieved. Standard Scaler is the process where we subtract each data point from mean and divide it by standard deviation. We can fix the ranges of data points to be between -1 to 1 or 0 to 1. These parameters can be tweaked based on the result obtained from training and testing data.

MinMaxScaler is the process where each data point is subtracted from the minimum value of that feature and is divided by the difference of max and min of that column. This is another way of scaling data within a range specified. Depending on the plots of data after applying this, we can take a decision on which works better. For this dataset, Standard Scaler worked well.

### 2) K-FOLD CROSS VALIDATION:

This is the technique where you split data into train and test with an 80-20 split but is done in K-Folds or K different iterations which builds more trust in the performance of the model because the data is shuffled each time among train-test-splits. For this project, 5 folds have been used

### 3) MACHINE LEARNING ALGORITHMS:

For this project, four algorithms have been explored as mentioned above. Logistic Regression, random forest, Decision Tree and ANN have been used. Details of parameters are mentioned below.

#### LOGISTIC REGRESSION:

This is a fast and effective supervised classification algorithm. Since, the scope of this project is multiclass classification, solver = 'lbfgs' is a parameter that is used while training and testing the data. K-fold of 5 and Standard Scaler has been applied on data with 80-20 split then trained and tested.

### DECISION TREE:

A decision tree is a tree-based supervised learning algorithm. It supports multiclass classification and is tested for k-folds where  $k=5$  after scaling the data or standardizing the data with a train-test split of 80-20 just like the other algorithms.

### RANDOM FOREST:

This is a tree-based algorithm which is a bagging technique of multiple decision trees. Random Forest by default supports multiclass. So only various parameters within the algorithm have been tweaked and tested. Best results were obtained for the following parameters where  $n\_estimators=100$ ,  $min\_samples\_leaf=3$  which means it has 100 iterations and 3 levels of the tree.

### ARTIFICIAL NEURAL NETWORK:

This is an unsupervised learning algorithm that was used to improve the results of the model. This has experimented with both the datasets. This project uses the Keras model of Neural Network. It is initialized using Sequential. It uses 100 neurons on the first hidden layer with 305 input parameters. This seemed to give better results where neurons and the number of layers have experimented. A batch size of 100 is set with 50 epochs gives good results. A dropout layer is added to avoid overfitting or noise in the data.

#### 4) EVALUATE RESULTS:

This step will be covered in the next section with appropriate graphs, results, and samples.

## 5. EVALUATION OF RESULTS

In this section, details of all the results obtained from both the datasets are reported.

### DATASET 1:

Once, the data is standardized, machine learning algorithms are applied as below.

### LOGISTIC REGRESSION:

An accuracy of 39% was achieved which was very low. Accuracy scores after applying 5-fold cross-validation are as shown below. One of the reasons this algorithm performed poorly might be because Logistic Regression works best for binary classification rather multiclass classification. Also, the sample of data is too less with just 13k records for each class. Another reason is cost functions of cost 2 and cost 3 are mostly misclassifying as cost 1 and cost 3 respectively which impacts the overall accuracy.

Logistic Regression scores:

```
[0.3924612 0.39079823 0.38562454 0.39661863 0.39138871]
```

Logistic Regression scores mean: 0.39137825921487857

Below is the classification report of data.

---

#### Logistic Regression Classification Report

	precision	recall	f1-score	support
0.0	0.44	0.79	0.56	13529
1.0	0.22	0.13	0.16	13530
2.0	0.28	0.13	0.18	13530
3.0	0.45	0.51	0.48	13530
accuracy			0.39	54119
macro avg	0.35	0.39	0.35	54119
weighted avg	0.35	0.39	0.35	54119

Fig 22. Logistic Regression Classification Report

Below is the confusion matrix before with and without the normalization of data.

```
Logistic Regression Confusion matrix, without normalization
[[10737 1141  593 1058]
 [ 7848 1741 1374 2567]
 [ 4189 2819 1762 4760]
 [ 1837 2151 2601 6941]]
```

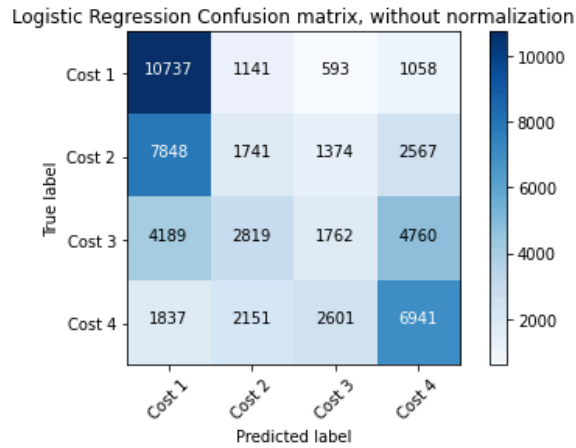


Fig 23. Logistic Regression Confusion Matrix without normalization

```
Logistic Regression Normalized confusion matrix
[[0.79 0.08 0.04 0.08]
 [0.58 0.13 0.1 0.19]
 [0.31 0.21 0.13 0.35]
 [0.14 0.16 0.19 0.51]]
```

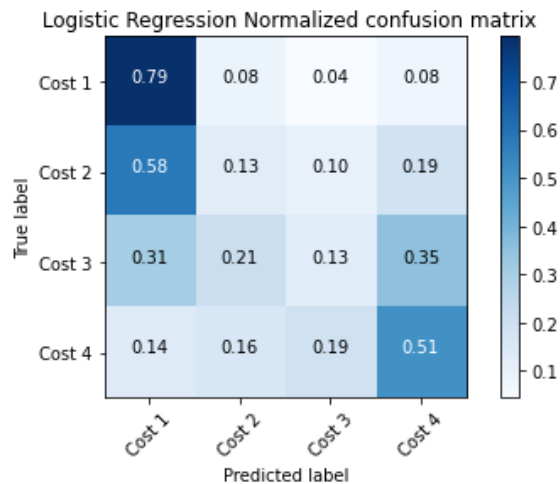




Fig 24. Logistic Regression Confusion Matrix with normalization

**DECISION TREE:**

An accuracy of 37% is reported which means it performed poorly than Logistic Regression. Decision Trees might have performed poorly because of lack of data. Another reason is cost functions of cost 2 and cost 3 are mostly misclassifying as cost 1 and cost 3 respectively which impacts the overall accuracy.

```
Decision Tree scores: [0.36 0.37 0.38 0.36 0.37]
Decision Tree scores mean: 0.36911253879617323
```

Classification report and Confusion matrix are as shown below:

```
Decision Tree Classification Report
```

	precision	recall	f1-score	support
0.0	0.50	0.51	0.50	13529
1.0	0.27	0.27	0.27	13530
2.0	0.28	0.28	0.28	13530
3.0	0.42	0.41	0.42	13530
accuracy			0.37	54119
macro avg	0.37	0.37	0.37	54119
weighted avg	0.37	0.37	0.37	54119

Fig 25. Decision Tree Classification Report

Below is the confusion matrix before with and without the normalization of data.

Decision Tree Confusion matrix, without normalization  
[[6893 4097 1449 1090]  
[4105 3719 3564 2142]  
[1759 3523 3810 4438]  
[1090 2284 4581 5575]]

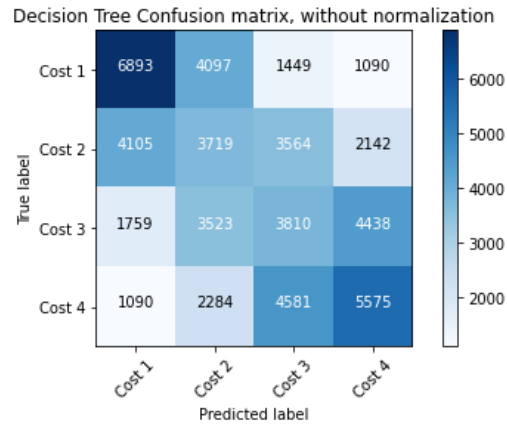


Fig 26. Decision Tree Confusion Matrix without normalization

Decision Tree Normalized confusion matrix  
[[0.51 0.3 0.11 0.08]  
[0.3 0.27 0.26 0.16]  
[0.13 0.26 0.28 0.33]  
[0.08 0.17 0.34 0.41]]

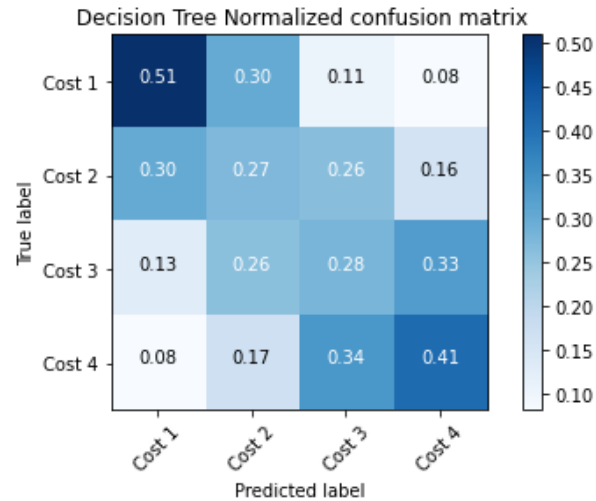


Fig 27. Decision Tree Confusion Matrix without normalization

## RANDOM FOREST

An ensemble of decision trees using the bagging technique has been applied. An accuracy of 43% was reported which is an improvement on the other two algorithms.

Random Forest has an improvement in scores because it uses bagging technique and eliminates weak classifiers in decision tree. It still performs poorly because of insufficient data. Accuracy scores after 5-fold cross-validation are as follows:

```
Random Forest scores: [0.42 0.43 0.44 0.46 0.43]
Random Forest scores mean: 0.43639008321701905
```

Classification Report and Confusion Matrix are as shown below.

Random Forest Classification Report					
	precision	recall	f1-score	support	
0.0	0.44	0.79	0.56	13529	
1.0	0.22	0.13	0.16	13530	
2.0	0.28	0.13	0.18	13530	
3.0	0.45	0.51	0.48	13530	
accuracy			0.39	54119	
macro avg	0.35	0.39	0.35	54119	
weighted avg	0.35	0.39	0.35	54119	

Fig 28. Random Forest Classification Report

```
Random Forest Confusion matrix, without normalization
[[10737 1141 593 1058]
 [ 7848 1741 1374 2567]
 [ 4189 2819 1762 4760]
 [ 1837 2151 2601 6941]]
```

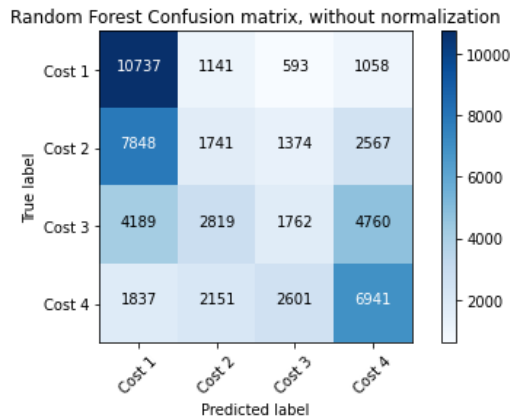


Fig 29. Random Forest Confusion Matrix without normalization

Random Forest Normalized confusion matrix  
[[0.79 0.08 0.04 0.08]  
[0.58 0.13 0.1 0.19]  
[0.31 0.21 0.13 0.35]  
[0.14 0.16 0.19 0.51]]

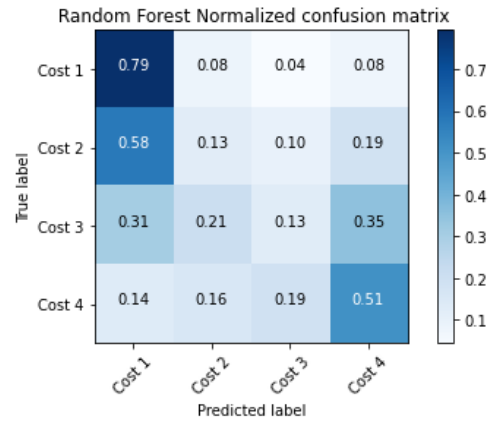


Fig 30. Random Forest Confusion Matrix with normalization

## ARTIFICIAL NEURAL NETWORKS (ANN)

As expected, ANN outperforms the other algorithms, with an accuracy of 65%.

Below is the classification report of ANN

	precision	recall	f1-score	support
0	0.76	0.70	0.73	2759
1	0.55	0.58	0.57	2649
micro avg	0.65	0.64	0.65	5408
macro avg	0.66	0.64	0.65	5408
weighted avg	0.66	0.64	0.65	5408

Fig 31. ANN Classification Report

Below is the confusion matrix for ANN with and without normalization

ANN Confusion matrix, without normalization

```
[[1937 686 96 40]
 [ 493 1547 499 110]
 [ 88 470 1518 658]
 [ 39 89 687 1867]]
```

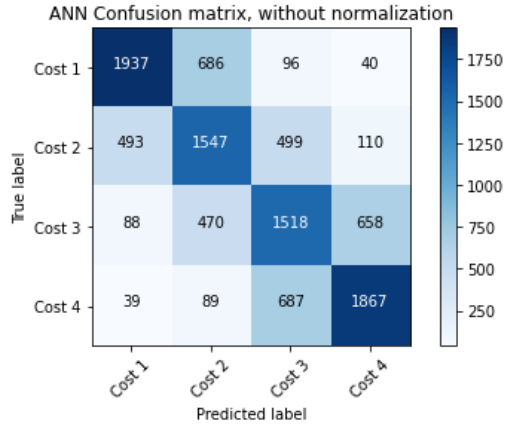


Fig 32. ANN Confusion Matrix without normalization

ANN Normalized confusion matrix

```
[[0.7 0.25 0.03 0.01]
 [0.19 0.58 0.19 0.04]
 [0.03 0.17 0.56 0.24]
 [0.01 0.03 0.26 0.7 ]]
```

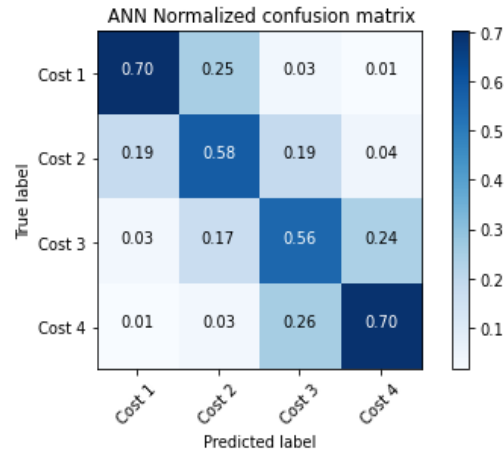


Fig 33. ANN Confusion Matrix with normalization

Since this dataset results are not impressive and accurate, a strategy of grouping costs together was an alternative which has been explained below.

**DATASET 2:**

Results of using a dataset with bucketed costs are as follows:

**LOGISTIC REGRESSION:**

An accuracy of 53% was achieved which was a major improvement compared to previous experiments but is still low. After bucketing the costs there is a vast improvement in performance from Approach 1 but it still performs poorly because Logistic Regression is used for binary classification but has been used in this project because it is fast. Another reason for low accuracy is misclassification problem and hence if there were just two classes, this algorithm might have performed better. Accuracy scores after applying 5-fold cross-validation are as shown below.

Logistic Regression scores:

```
[0.57600394 0.57618872 0.57204155 0.57347869 0.57117047]
```

Logistic Regression scores mean: 0.5737766739363895

Classification Report of Logistic Regression is as shown below:

Logistic Regression Classification Report				
	precision	recall	f1-score	support
0.0	0.59	0.78	0.67	54119
1.0	0.44	0.22	0.29	54120
2.0	0.52	0.46	0.49	67650
3.0	0.65	0.80	0.72	67650
accuracy			0.57	243539
macro avg	0.55	0.57	0.54	243539
weighted avg	0.55	0.57	0.55	243539

Fig 34. Logistic Regression Classification Report

Confusion Matrix without normalization is as shown below.

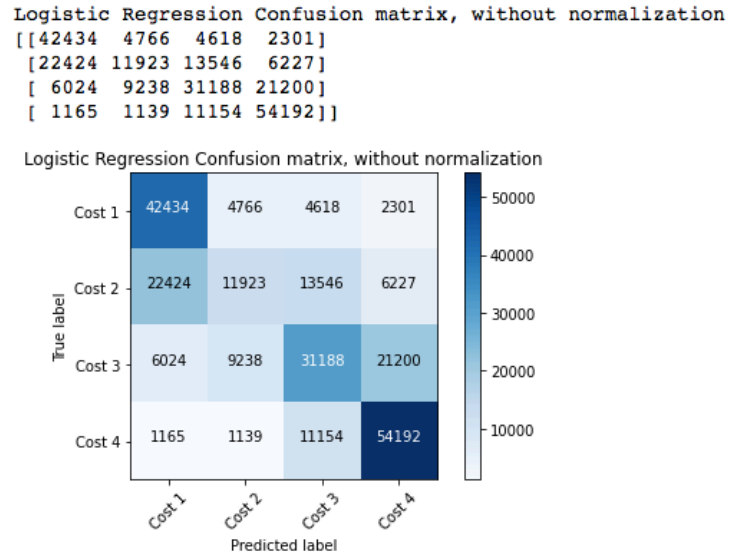


Fig 35. Logistic Regression Confusion Matrix without normalization

Logistic Regression Normalized confusion matrix

```
[[0.78 0.09 0.09 0.04]
 [0.41 0.22 0.25 0.12]
 [0.09 0.14 0.46 0.31]
 [0.02 0.02 0.16 0.8 ]]
```

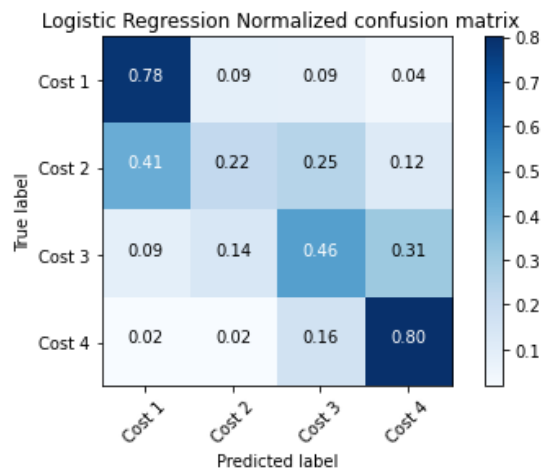


Fig 36. Logistic Regression Confusion Matrix with normalization

## RANDOM FOREST

An accuracy of 71% was achieved which was a major improvement compared to previous experiments and also Logistic Regression from the previous model. Accuracy scores after applying 5-fold cross-validation are as shown below. Model starts performing pretty well among all supervised algorithms.

Random Forest scores mean: 0.71

Classification Report is as shown below:

	precision	recall	f1-score	support
0.0	0.81	0.72	0.76	54119
1.0	0.62	0.58	0.60	54120
2.0	0.63	0.69	0.66	67650
3.0	0.78	0.80	0.79	67650
accuracy			0.71	243539
macro avg	0.71	0.70	0.70	243539
weighted avg	0.71	0.71	0.71	243539

Fig 37. Random Forest Classification Report

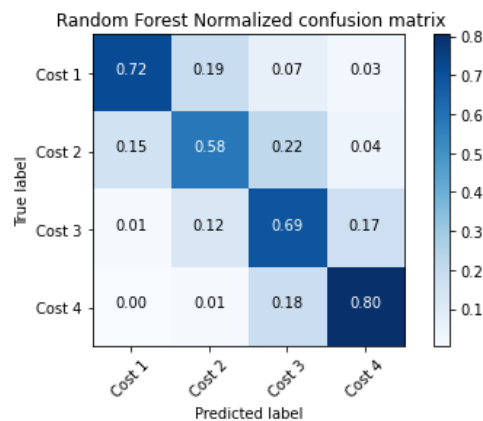


Fig 38. Random Forest Confusion Matrix without normalization



```
Random Forest Normalized confusion matrix
[[0.68 0.19 0.09 0.04]
 [0.2  0.42 0.28 0.1 ]
 [0.03 0.13 0.57 0.27]
 [0.01 0.01 0.18 0.8 ]]
```

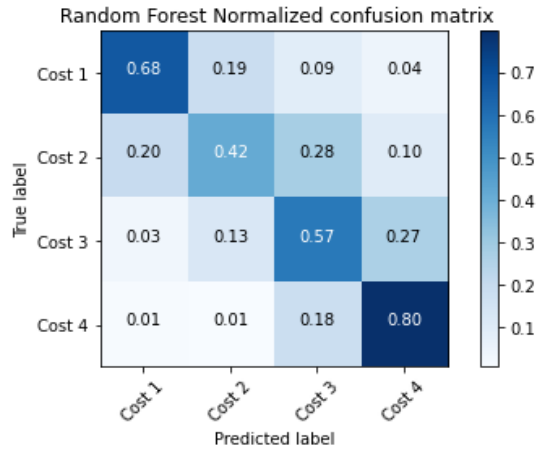


Fig 39. Random Forest Confusion Matrix with normalization

## DECISION TREES

An accuracy of 64% was achieved which was a major improvement compared to previous experiments and other models from this dataset. Accuracy scores after applying 5-fold cross-validation are as shown below. Since decision tree may be prone to overfitting or pruning problems, random forests is definitely a better choice to experiment with.

```
Decision Tree scores: [0.63 0.64 0.63 0.64 0.64]
Decision Tree scores mean: 0.6354546970971364
```

Classification Report is as shown below:

## Decision Tree Classification Report

	precision	recall	f1-score	support
0.0	0.73	0.68	0.70	54119
1.0	0.53	0.49	0.51	54120
2.0	0.55	0.59	0.57	67650
3.0	0.73	0.75	0.74	67650
accuracy			0.64	243539
macro avg	0.63	0.63	0.63	243539
weighted avg	0.64	0.64	0.63	243539

Fig 40. Decision Tree Classification Report

Confusion Matrix without normalization is as shown below:

## Decision Tree Confusion matrix, without normalization

```
[[36938 10626 4790 1765]
 [10881 26659 13349 3231]
 [ 2381 11121 40134 14014]
 [  700  1728 14217 51005]]
```

## Decision Tree Confusion matrix, without normalization

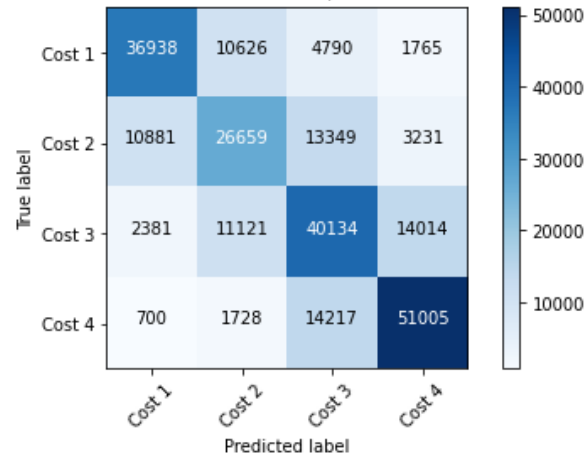


Fig 41. Decision Tree Confusion Matrix with normalization

```

Decision Tree Normalized confusion matrix
[[0.68 0.2  0.09 0.03]
 [0.2  0.49 0.25 0.06]
 [0.04 0.16 0.59 0.21]
 [0.01 0.03 0.21 0.75]]

```

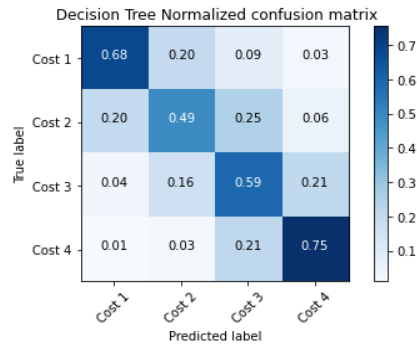


Fig 42. Decision Tree Confusion Matrix with normalization

## ARTIFICIAL NEURAL NETWORKS (ANN)

As expected, ANN outperforms the other algorithms, with an accuracy of 81%. This is the best model across all the algorithms that have been used. Costs can be associated with severity levels of depression ranging from lower to higher which means class 0 is lowest depression and class 3 is the highest depression. Based on the precision and recall obtained for this model, the model seems to be performing well as the people with the highest and lowest depression costs seem to be performing reasonably well so these costs would be ideal to be picked.

Below is the classification report of ANN

	precision	recall	f1-score	support
0	0.86	0.86	0.86	10947
1	0.75	0.70	0.72	10818
2	0.76	0.79	0.77	13509
3	0.86	0.87	0.87	13434
accuracy			0.81	48708
macro avg	0.81	0.80	0.81	48708
weighted avg	0.81	0.81	0.81	48708

Fig 43. ANN Classification Report

Confusion Matrix with and without normalization is as shown below.

```
ANN Confusion matrix, without normalization
[[ 9186 1386  160  44]
 [ 1408 7770 1622  96]
 [  108 1208 10551 1576]
 [   90   60 1719 11724]]
```

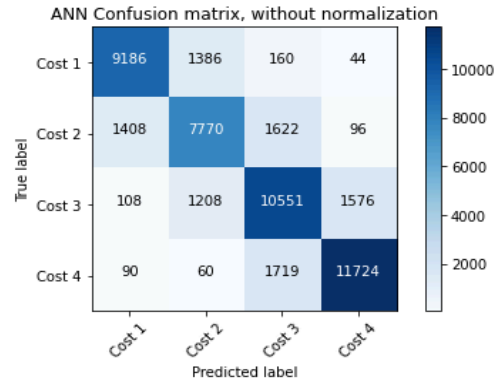


Fig 44. ANN Confusion matrix without normalization

```
ANN Normalized confusion matrix
[[0.85 0.13 0.01 0. ]
 [0.13 0.71 0.15 0.01]
 [0.01 0.09 0.78 0.12]
 [0.01 0.  0.13 0.86]]
```

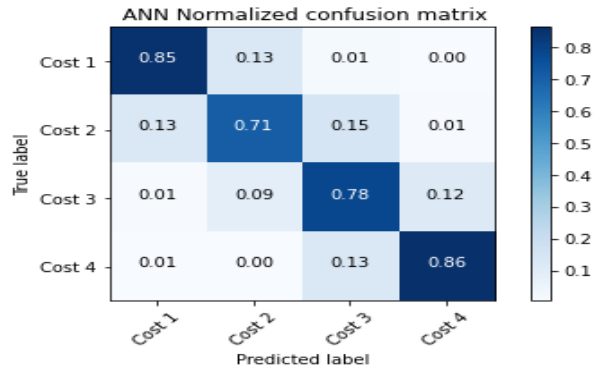


Fig 45. ANN Confusion matrix with normalization

## 6. CONCLUSION AND DISCUSSION

Based on the above results reported, a neural network seemed to have worked better on the above data. This indicates that ROI connectivity can be established by bucketing costs which indicate that, there is no bigger difference if you have a bucketed cost which is in closer range. This indicates that 4 patterns of costs can be clubbed together to identify whether a pair of ROIs are connected.

Based on this cost function, in the future you can apply to real patient data where you can identify which cost a patient belongs to and once the probability is also determined, this indicates whether a person has depression or not where higher cost indicates higher severity of depression. Based on the connectivity of ROI and cost, this information can be given to doctors to treat patients with depression as it helps in diagnosis and severity level of depression. This will further enable them to diagnose the patients and treat them with proper medication.

Another important observation found is after increasing dataset size from 50k to about 200k, all algorithms including Logistic Regression have performed extremely well. This indicates that if the data were increased more than all the algorithms will start performing well. Also 40 patient's data are too small to experiment with and increasing it to at least 300-400 patients will make a huge difference on the entire project.

## 7. FUTURE WORK

Although the original idea was to work on real patient data for PA hospitals, due to time constraints and security clearances this could not happen and had to work on the FCP dataset. However, this model would act as a baseline and can be trained on depression data of real patients and tests based on it. Regarding the model performance, there can be more improvements by taking a different dataset or increasing the size of a dataset as 40 patient's data might have been too low but could have experimented with lower cost functions. Also, another possible option would be of considering senior patients' data. The dataset belongs to a younger demographic of 18-40 whereas it is observed that senior

patients may have much more details in their topological structure of brain ROI and experiments with that dataset would have been interesting to compare.

## LITERATURE REFERENCES

- [1] American Psychiatric Association. (2002). *Diagnostic and statistical manual of mental disorders: Text revision*. Washington, DC
- [2] Diener, C., Kuehner, C., Brusniak, W., Ubl, B., Wessa, M., & Flor, H. (2012). A meta-analysis of neurofunctional imaging studies of emotion and cognition in major depression. *NeuroImage*, 61(3), 677–685. DOI: 10.1016/j.neuroimage.2012.04.005
- [3] Leonard, J. (2020, January 7). What are the types of depression? Retrieved from <https://www.medicalnewstoday.com/articles/327429>
- [4] Kessler, RC., Petukhova, M., Sampson, NA., Zaslavsky, AM., Wittchen, H. (2012). Twelve-month and lifetime prevalence and lifetime morbid risk of anxiety and mood disorders in the United States. *Int J Methods Psychiatr Res*, 21, 169–84. DOI: 10.1002/mpr.1359
- [5] Grecious MD, et al. Resting-state functional connectivity in major depression: abnormally increased contributions from subgenual cingulate cortex and thalamus. *Biol Psychiatry*. 2007; 62:429–437.
- [6] Knutson B, Bhanji JP, Cooney RE, Atlas LY, Gotlib IH. Neural responses to monetary incentives in major depression. *Biol Psychiatry*. 2008; 63:686–692.
- [7] Cook SC, Wellman CL. Chronic stress alters dendritic morphology in rat medial prefrontal cortex. *J Neurobiol*. 2004; 60:236–248.

- [8] Liston C, et al. Stress-induced alterations in prefrontal cortical dendritic morphology predict selective impairments in perceptual attentional set-shifting. *J Neurosci*. 2006; 26:7870–7874.
- [9] Drevets, WC., Price, JL., Furey, ML. (2008). Brain structural and functional abnormalities in mood disorders: implications for neurocircuitry models of depression. *Brain Struct Funct*, 213, 93–118. DOI: 10.1007/s00429-008-0189-x
- [10] Van Den Heuvel, MP., Hulshoff Pol, HE. (2010). Exploring the brain network: A review on resting-state fMRI functional connectivity. *European Neuropsychopharmacology*, 20, 519–34. DOI:10.1016/j.euroneuro.2010.03.008
- [11] Gaudio, S., Wiemerslage, L., Brooks, S. J., & Schiöth, H. B. (2016). A systematic review of resting-state functional-MRI studies in anorexia nervosa: Evidence for functional connectivity impairment in cognitive control and visuospatial and body-signal integration. *Neuroscience & Biobehavioral Reviews*, 71, 578–589. DOI: 10.1016/j.neubiorev.2016.09.032
- [12] Mulders, PC., Van Eijndhoven, PF., Schene, AH., Beckmann, CF., Tendolkar, I. (2015). Resting-state functional connectivity in major depressive disorder: A review. *Neuroscience & Biobehavioral Reviews*, 56, 330–44. DOI:10.1016/j.neubiorev.2015.07.014
- [13] CONN toolbox - Graphs (ROI-level). (n.d.). Retrieved from <https://sites.google.com/view/conn/definitions/connectivity-measures/graphs-roi-level>.
- [14] CONN toolbox (www.nitrc.org/projects/conn, RRID:SCR\_009550)

- [15] Whitfield-Gabrieli, S., & Nieto-Castanon, A. (2012). *Conn: A functional connectivity toolbox for correlated and anticorrelated brain networks*. *Brain connectivity*, 2(3), 125-141
- [16] Wikipedia contributors. (2020, April 18). Multinomial logistic regression. Retrieved from [https://en.wikipedia.org/wiki/Multinomial\\_logistic\\_regression](https://en.wikipedia.org/wiki/Multinomial_logistic_regression)
- [17] M. (2018, July 22). Multi-class logistic regression. Retrieved from <https://mashimo.wordpress.com/2018/07/22/multi-class-logistic-regression/>
- [18] Donges, Niklas. "A Complete Guide to the Random Forest Algorithm." *Built-In*, 16 June 2019, [builtin.com/data-science/random-forest-algorithm](https://builtin.com/data-science/random-forest-algorithm).