Deep Learning based classification approach to predicting Parkinson disease using distinctive features of audio.

Tejaswini Rajendra Kale
10514273@mydbs.ie

Dissertation submitted in partial fulfilment of the requirements for the degree of

M.Sc. Data Analytics

at Dublin Business School

Supervisor: Terri Hoare

January 2020

Declaration

I declare that this dissertation that I have submitted to Dublin Business School for the award of M.Sc. Data Analytics is the result of my own investigations, except where otherwise stated, where it is clearly acknowledged by references. Furthermore, this work has not been submitted for any other degree.

Signed: Tejaswini Rajendra Kale
Student Number: 10514273
Date: 6 Jan 2020

Acknowledgment

I would with express my gratitude to Ms. Terri Hoare, my supervisor for valuable guidance that she provided during the complete tenure of this research work. I would like to thank her and Dublin Business School for support in getting access to critical health care data. I would like to extend my thanks to Synapse for providing access to medical confidential data.

Abstract

Vocal disorder is present in more than 90% of people suffering from Parkinson's disease. Speech impairment can be considered for early detection of Parkinson's disease. This research focuses on prediction of Parkinson's disease using audio features. Various features such as formant, jitter, shimmer, MFCC are extracted form audio files using PRAAT and Librosa python libraries. Deep learning models are used to differentiate Parkinson's disease patients from healthy people. Deep learning models are trained and tested on data obtained from the mPower synapse database containing audio files of 5826 people. The research includes a comparative study of classifiers trained using deep learning versus traditional machine learning algorithms. Results obtained show that deep neural network classifiers outperform traditional machine learning algorithms including KNN, logistic regression, GLM, decision tree, and random forest. The best classifier accuracy of 86.12% is obtained using a state-of-the-art H2O deep learning algorithm.

Table of Contents

List of Figures

List of Equations

List of Tables

List of Abbreviations

PD – Parkinson’s Disease

SVM – Support Vector Machine

KNN – K Nearest Neighbour

ANN – Artificial Neural Network

CNN – Convolution Neural Network

DNN – Deep Neural Network

AUC – Area Under Cover

Chapter 1: Introduction

1.1 Parkinson

More than 5.9 million people are diagnosed with Parkinson's disease (PD) worldwide, and was the cause of death for more than 100,000 people in 2013 (Krishnan, 2017). PDs signature syndromes embrace resting tremor, slowness of movement, and muscle rigidity.

Parkinson's disease is an enduring neurological degenerative disease affecting the central nervous system responsible for progressive evolution movement disorders. (A. Bourouhou, 2016). Cause of Parkinson disease is unknown, according to researchers Parkinson patients shows degradation of dopaminergic neurons which reduces the production of dopamine. Dopamine is organic chemical which function as hormone as well as neurotransmitter. Thus, it is used by brain to control body movements, decline in production of dopamine in Parkinson's patients make it difficult to control body movements. As direct cause of reduced control on motor neuron in central nervous system causes difficulty in articulation (Can, 2013).

It is extremely difficult to diagnose Parkinson disease as it is identified by movement analysis. PD develops for extended period being clinically silent hence motor movement are well suited to identify PD. (M. Wodzinski, 2019). Speech impairment is extremely common in people suffering with Parkinson disease. About 70% to 90% to percentage of people suffering with Parkinson disease suffers from voice impairment. According to (Ho, 1998) study conducted on 200 PD patients and found that 74% of PD patients were affected by different levels of speech impairment, from mild to profound.

1.1.1 Symptoms

- Shaking or instability of libs and finger when it is at rest
- Speech becomes monotonous and soft rather than usual tones.
- Slowness caused in movements of body; simple regular works becomes difficult to complete.
- Writing changes due to instability
- May get balancing problem

1.1.2 Effect of Parkinson on Voice

Nerves in brain and body are damaged by PD. Substantia nigra pars compacta is part of brain which damaged by PD that causes significant problems in speech of person suffering from Parkinson. (Downward, 2017). Speech impairments are caused by laryngeal function deficit, impaired mimicry, reduced lung life capacity and decreased speech force. Such changes lead to appearance of numerous deficits in voice and speech such as: reduction of loudness, lowering the tone of the voice, limited modulation (monotonous speech), difficulties with changes in loudness, voltage reduction of vocal folds, rough and hoarse tone, as well as improper articulation (speech becomes indistinct) and change in speech pace (M. Wodzinski, 2019).

1.2 Research Questions

1. How audio is affected in people suffering from Parkinson disease?

2. How to extract features of audio file in image format and in numeric(.csv) format?

3. Which algorithms will work better in classification; Traditional or Deep Learning algorithms?

4. Is there be scope of growth in audio processing for Parkinson classification?

5. How can we apply this work in real-time: health care institutes?

1.3 Hypothesis

1. Hypothesis

   Build Deep Learning models would work better in accordance with accuracy, precision, specificity, and sensitivity when compared to other models described in below sections.

1.4Research Flowchart

Flow chart of research was created to have clear and well-defined path for research depending on research questions.

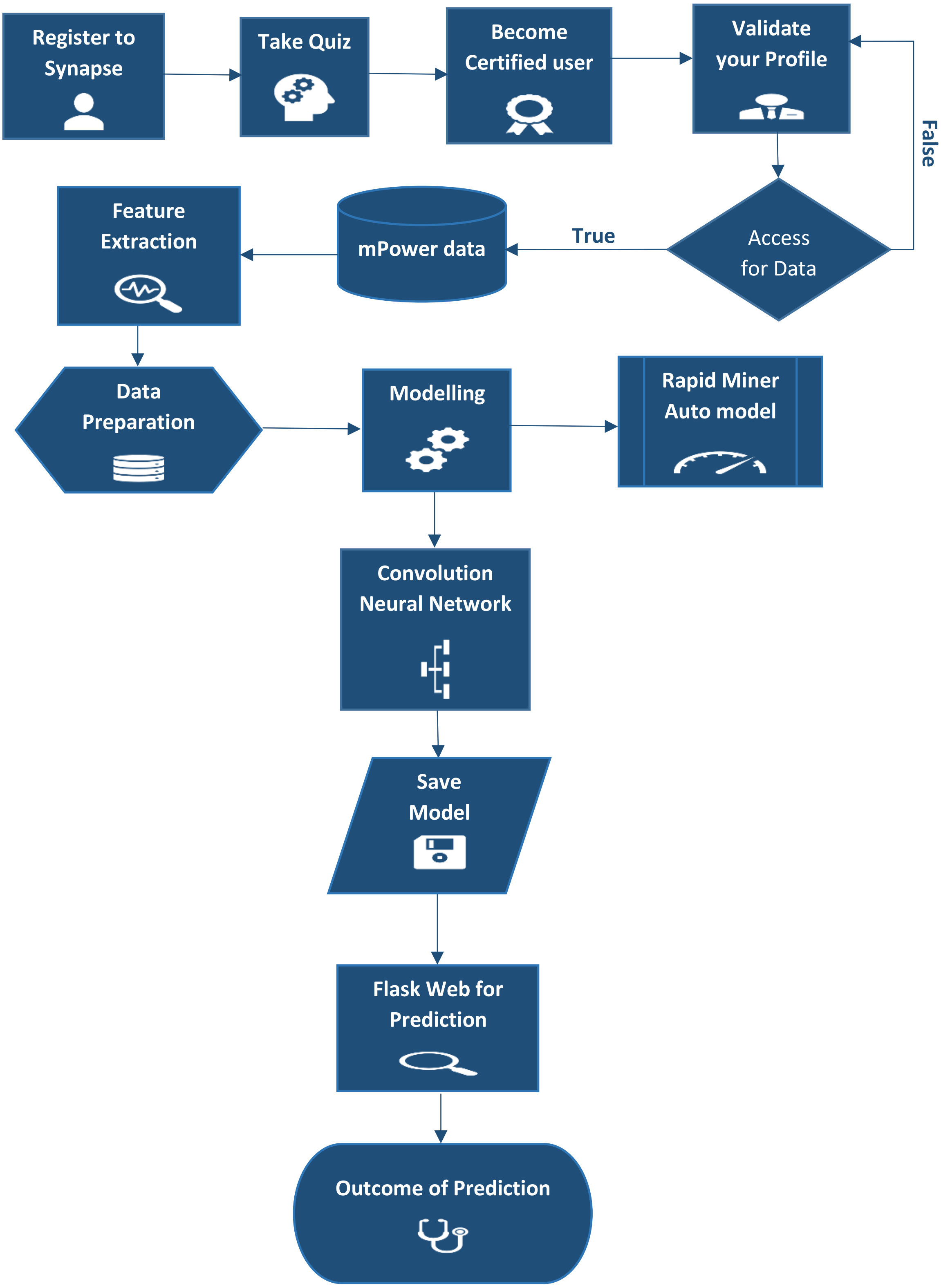


Figure 1:1 Research Flowchart

Chapter 2: Literature Review

2.1Introduction

Currently research is done to identify which machine algorithms performs best on audio data for analysis and classification of disease. It is challenging to conclude which algorithm performs best (traditional machine learning algorithms or deep learning) for classification of Parkinson's disease using audio recordings. For better understanding the concept and various model working on audio data for classification of Parkinson's disease following papers are referred.

2.2 Literature Review

According to (Can, 2013) paper, data used for study is a voice recording originally done at University of Oxford by M.A. Little. This data consists of 195 audio recordings extracted from 31 people whom 23 were suffering of Parkinson's disease. Data used was not balanced data which lead to no single class with true positive rate." The designed neural network system is boosted by filtering, and this causes a significant increase of robustness. It is also shown that by majority voting of eleven parallel networks, recognition rates reached to > 90 despite 3:1 imbalanced class distribution of the Parkinson's disease data set".

In this (A. Bourouhou, 2016) paper they have tried to apply different classification algorithms database which is available on University California Irvine (UCI) machine learning repository website. Database consist of voice recordings of 20 healthy people and 20 people who are suffering with PD. On this database they applied three classifiers for comparison; k-near neighbour "k-NN", the Naive Bayes "NB" and support vector machines "SVM". K-NN have a good detection performance with 70%, and high Support Vector Machines detection performance with an accuracy of 80% correct detection rate. But Naïve Bayes has a low quality of detection performance 65%.

The authors of (M. Wodzinski, 2019) created a system based on modifies ResNet algorithm to detect Parkinson disease by converting audio records of 100 people into imaged based representation of frequency features. Algorithm was able to classify data with 90% of accuracy.

In paper (Bolat, 2016) author discuss about structure and model building using Multi-Layer Perceptron (MLP) and Generalized Regression Neural Networks (GRNN) algorithm. Database used for classification contain 40 subjects, 20 of them were suffering from Parkinson's disease and remaining were healthy. Accuracy acquired by models are 57.5% for MLP.

According to analysis of (T. J. Wroge, 2018) Artificial Neural Network (ANN) works better on mPower data which consist of 60000 audio recording of 5826 people who donated their audio for research. Data used for modelling was subset of complete data. Audio-Visual Emotion recognition Challenge (AVEC) and Geneva Minimalistic Acoustic Parameter Set (GeMaps) using the openSMILE toolkit are used to extract features from audio recordings. Investigations suggest that model trained on AVEC extracted features outperformed on model trained on GeMaps extracted features as AVEC has more features than GeMaps. Model trained on AVEC features achieved accuracy of 86%.

In paper (Zhijing Xu, 2018) author specific that audio of vowel 'u' has discriminative information than other vowels for detection of Parkinson's disease using machine learning. According to research DNN provides highest accuracy of 89.5% on data of 20 healthy people and 20 Parkinson patients. They have used MFCC feature of voice to train and test DNN model.

Diagnosis of Parkinson’s disease in (Frid, 2016)is done using Convolution neural network consisting of one convolution layer, pooling layer and fully connected layer ending with softmax activation function. Data set used in this research consist of 43 patients, 9 were healthy. Rainbow passage is used to record audios of patients. Convolution neural network is trained on 85% of data and 15% is used for validation acquiring average accuracy of 80.5%.

(Ozkan, 2016) depicts comparisons of commonly used six machine learning algorithms with cross validation of 2 folds, 5 folds, 10 folds for classification of Parkinson’s disease. According to research, combination of principle component analysis and KNN algorithm along with cross validation 10 folds provide accuracy of 99.1% on dataset which consist of 195 audio records of 31 people.

Dataset created by Max Little of the University of Oxford is used in (Das, 2010) research for classification of Parkinson’s disease. Database consist of 197 rows and 23 columns extracted from audio of 31 people, 21 people having Parkinson’s disease. Research illustrate comparison of three machine learning models decision tree, neural network, and logistic regression and DMneural for classification. Neural network outperforms over other three algorithms with 92.9% accuracy.

(S. Aich, 2019) have reduces original feature set of audios to Genetic Algorithm (GA) and PCA using feature selection technique. Model is trained on audio records collected from 31 people, 23 were having Parkinson’s disease. They compared performance of various machine learning models on GA features and PCA features. Research conclude that GA features outperform over other algorithms with accuracy of 97.57%.

(Frid, 2014) have uses data of 43 patients according to their grade of disease, they have used Rainbow passage reading during examination. Result of using machine learning SVM algorithm is 81% accuracy.

(E. J. Alqahtani, 2018) depict how new technique nearest neighbour graph (**NNG**) works on data consisting of audio recording of 31 individual, 23 having Parkinson’s disease. NNG is rule based machine learning classifier. Using NNG classifier on WEKA for Parkinson’s classification acquired 96.30% accuracy.

Dataset created by Max little University Oxford and National Centre for Voice and Speech, Denver, Colorado is used in (S. Aich, 2018) research. Audio data of 31 people was collected out of which 23 had Parkinson’s disease. They compared performance of various machine learning models on original extracted features and PCA features. Research concluded that random forest with PCA features outperform over other algorithms with accuracy of 96.87%.

(H. Hazan, 2012) uses data from two different location USA and Germany and apply machine learning algorithm SVM on it producing conclusion that set of features required are language dependent and model trained with one country can be used in other country for testing. SVM acquires 90% accuracy on both data.

2.3Take away form review

It is evident from literature review that current research is conducted on audio record of limited people as accessing medical data due to confidentiality is challenging task. Many researchers have compared different traditional machine learning algorithms for classification of audio data, results depict that SVM outperform among traditional models. Some of researchers have explored area of deep learning for audio classification and had showed reliable results for audio classification but data used for classification is small.

From existing research in this field, following models are used to compare results on large data set

- Artificial Neural Network
- Deep Neural Network

- Convolution Neural Network
- Traditional Machine learning algorithms

Chapter 3: Research Methodology

Research is done using CRISP-DM methodology. CRISP-DM methodology helps for better understanding and faster process of research. It consists of six phases which aid in better understanding of process and provide research road map for planning and execution of research.

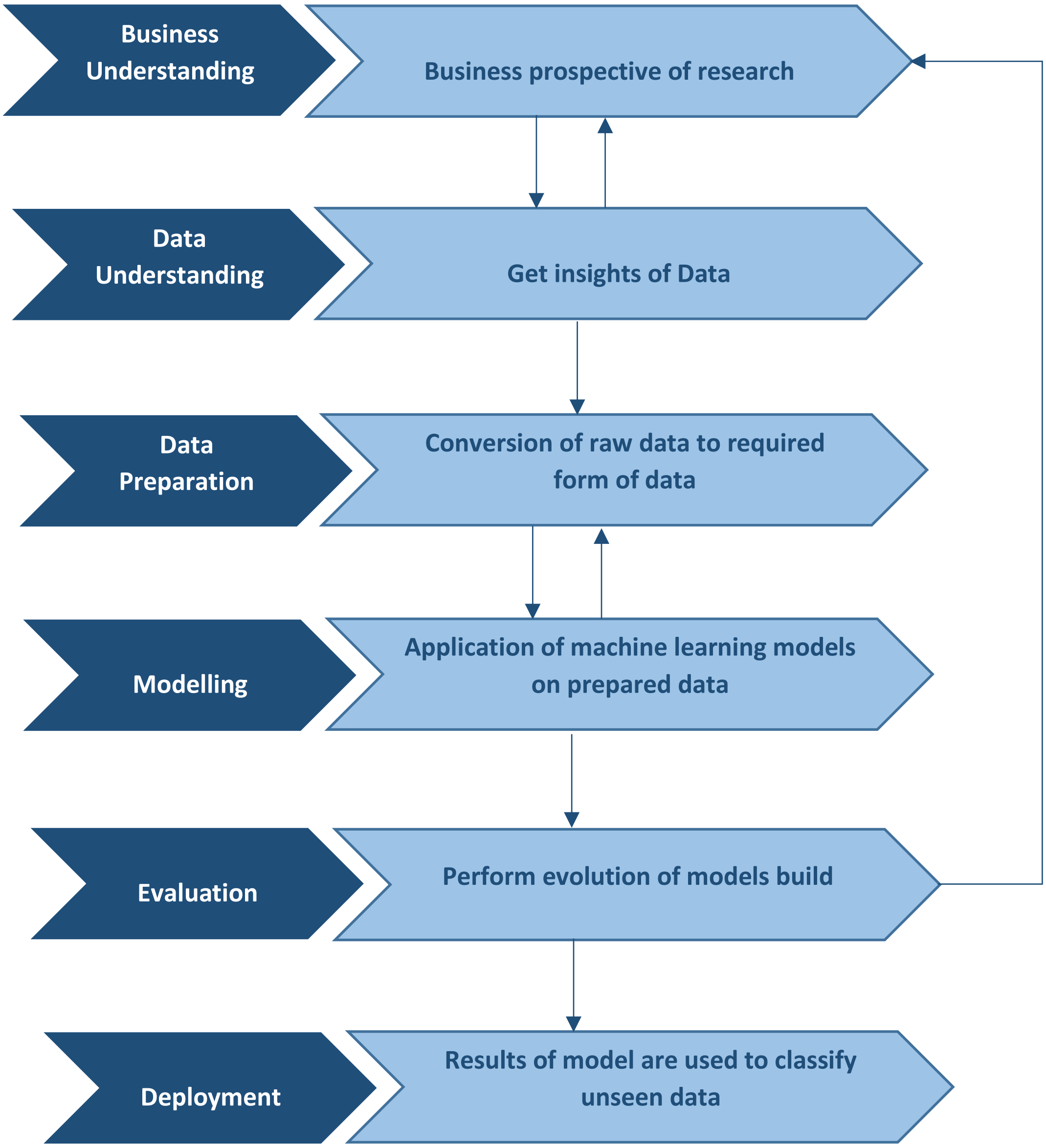

Business Understanding
Business prospective of research
Data Understanding
Get insights of Data
Data Preparation
Conversion of raw data to required form of data
Modelling
Application of machine learning models on prepared data
Evaluation
Perform evolution of models build
Deployment
Results of model are used to classify unseen data

Figure 3:1 CRISP-DM Methodology

3.1Business Understanding

Business understanding is first and primary phase of CRISP-DM model. Aim of this phase is to understand research objectives from business prospective. This business perspective is converted to research problem and plan is created to achieve this objective.

Main objective of this research is to diagnose if person having Parkinson's disease from audio data. Currently PET-CT imaging test is used to determine reduction in dopaminergic neurons this test is expensive and hence reduces diagnosis rate. Thus, this thesis focuses on building machine model on audio data which will easy the process of diagnosis and will be inexpensive to use, this will increase efficiency of treatment.

3.2 Data Understanding

Data understanding is second phase of CRISP-DM model which begins with collection of data. After data is collected it is explored to get insight of data.

In mPower data collection 5826 people participated in donation of voice recording multiple times and 6805 people attended survey question which are used to label audio data.

Table Name	Number of Participants	Description
syn5511444(Voice Activity)	5826	• Participants record themselves saying 'aaaah' for 10 seconds using mPower mobile application and microphone. • Multiple records are collected from unique participants thus number of voice recordings in database are about 60000
syn5511429(Demographics Survey)	6805	• Demographic Survey include questions that has multiple choice as well as some have integer as input. • This table is used for labelling audio records depending on health history.

Table 3:1 Database Description

Flow Chart of labelling audio files

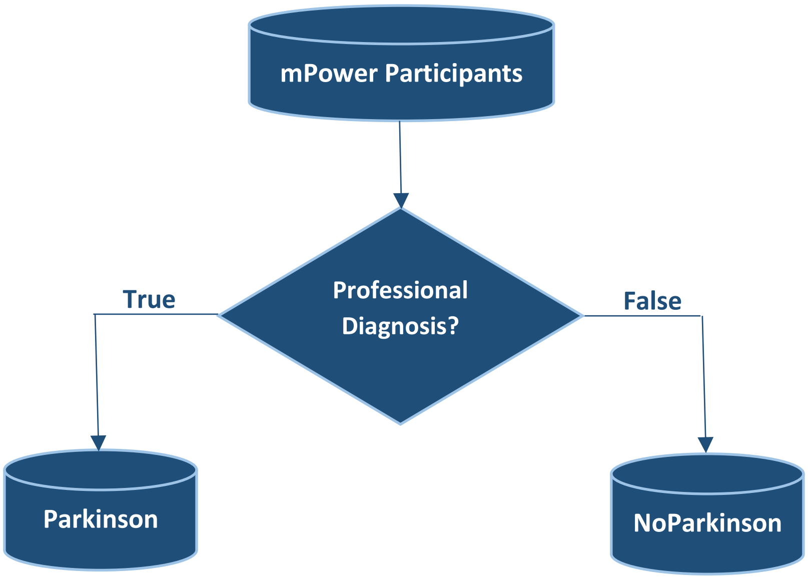


Figure 3:2 Data Segregation Flowchart

3.2.1 Data Access

Accessing data was challenging task due government rule of GDPR and confidentiality medical data. Synapse is research platform which permits people to share data, projects, and analysis of data, mPower is public research portal for accessing Parkinson disease data. It has donated records of thousands of people which can be used for research and development in healthcare sector.

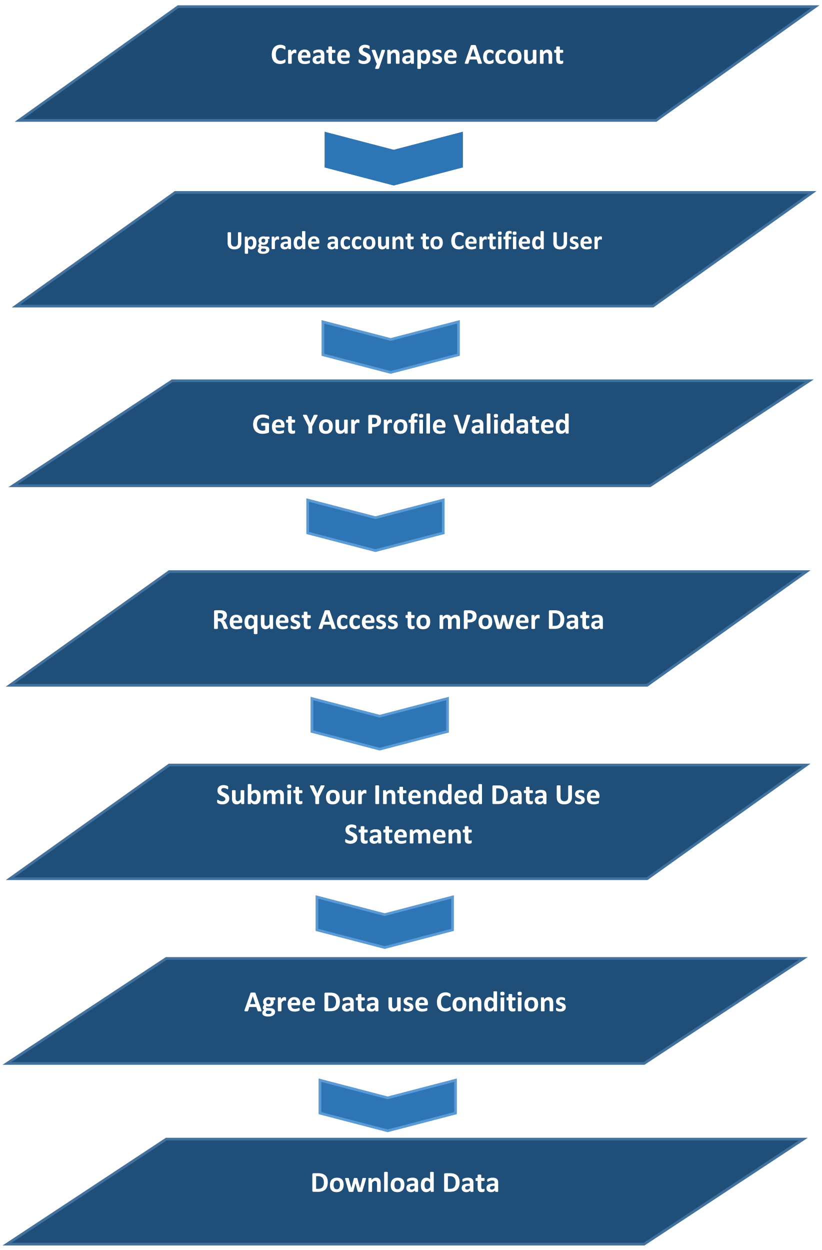

Create Synapse Account
Upgrade account to Certified User
Get Your Profile Validated
Request Access to mPower Data
Submit Your Intended Data Use Statement
Agree Data use Conditions
Download Data

Figure 3:3 Steps to Access Data

3.2.1.1 Account Creation

Anyone can access public data from synapse who is registered with them. For that you must create account with email address, and you will receive verification mail to complete the process of registration.

3.2.1.2 Become Certified User

To access or upload data on synapse website you need to demonstrate awareness of security and privacy issue. For that you need to take a quiz in which you must score above 90% to become certified user.

You can complete this by taking a Certification Quiz.

3.2.1.3 Profile Validation

User who is certified can apply to get profile validated. Follow below process to get your profile validated.

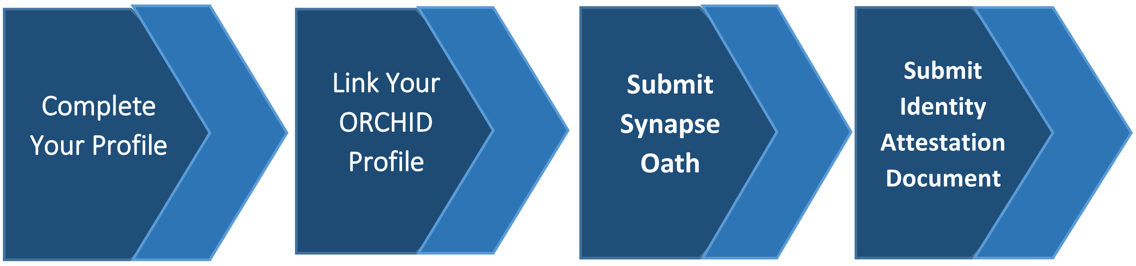


Figure 3:4 Profile Validation Steps

3.2.1.4 Request Access to mPower

To request access to the mPower data and submit your Intended Data Use Statement, please click this link: https://www.synapse.org/#!AccessRequirements:ID=syn5608426&TYPE=ENTITY and click "request access".

3.2.1.5 Agree to Conditions of data use

To access data from mPower you must accept their conditions of data usage

3.2.1.6 Python Code to Download Data

Synapse Client is python library to directly access data from synapse database using python code.

- **Install synapseclient library**

```
! pip install synapseclient
```

- **Login to Synapse Database**

```
syn = synapseclient.Synapse()
syn.login(UserName, 'Password')
Welcome, Tejaswini Kale!
```

- **Download data From Synapse Database**

```
def downloadAll():
  results = syn.tableQuery("SELECT  * FROM syn5511444 ")
  demographic = syn.tableQuery('SELECT* FROM syn5511429')
  df=results.asDataFrame(rowIdAndVersionInIndex=True)
  df.set_index(['recordId'], inplace = False)
  demographicdf=demographic.asDataFrame(rowIdAndVersionInIndex=True)
  colList=['healthCode','professional-diagnosis','age','gender']
  demographicdf=demographicdf[colList]
  mergedDf = pd.merge(df, demographicdf, how='inner',on=['healthCode'])
  mergedDf = mergedDf.set_index("audio_audio.m4a", drop = False)
  return mergedDf
```

3.3 Data Preparation

For data cleaning and consolidation Python code is used.

3.3.1 Transform

Transform is a process of adding new column or removing columns from existing file.

Pandas is a library in python which is used for transformation of data. First it reads ‘csv files’ in panda’s data frame and then use built-in functions to transform data.

There are some columns in feature extracted file which are not necessary for building model such as audioId, recordId, healthCode are dropped using following code.

Following code is used to read ‘csv file’ in data frame.

```
data=pd.read_csv ("/content/dataCheck.csv", usecols=range (0, 46))
```

Pandas data frame has function called drop (), which can be used to delete unwanted columns from file.

```
data=data.drop(['audioId',"recordId",'healthCode'], axis=1)
```

3.3.2 Merge

Merge can be defined as append or join operation used combine two files together. Pandas data frame has function called as merge (), which can be used for joining two data frames into one depending on types of join used.

There are two different data frames, one holding data of survey questions and other have extracted features from audio. These two files are merged using inner join.

Following code is for merging two data frames using inner join

```
mergedDf = pd.merge(df, demographicdf, how='inner', on=['healthCode'])
```

3.3.3 Clean

Clean is process of removing missing values from data. Feature extracted from audio may have some missing values as all attributes may not be present at specific time frame. Thus, it is important to drop such data to avoid anomalies and to obtain correct the data.

Pandas data frame have function dropna () to remove all missing values from data frame.

```
data = data.dropna(how='any’, axis=0)
```

Same operations can be performed in RapidMiner, when used for cross validation.

3.3.4 Feature extraction

3.3.4.1 Background

Extracting feature features from audio file is challenging task. Audio files which are available in synapse database are with ‘.tmp’ extension and thus required to be changed to ‘.m4a’ files for accessing audio files in its original format. It was necessary to extract information, which can explain data in numeric format, for extracting this numeric information from audio file python libraries are used; Librosa and PRAAT. Pydub python library is used to export these .m4a audio file to ‘.wav’ audio file as PRAAT takes only .wav files as input.

Extracted Features

MFCC, Pitch, Jitter, Shimmer, Formant

3.3.4.2 Python Library Dependencies

- Parselmouth Praat

  “Parselmouth is python library for PRAAT software” (Jadou, 2019). It has praat as packages which is used to extract various features from audio such as jitter, shimmer, formant.
- Librosa

  Librosa is python library for audio analysis and processing. It is used to extract features like MFCC, chroma_stft, rmse, spectral_centroid, spectral_bandwidth
- Pydub

  Pydub is python library used to manipulate audio files. Pydub is used to convert .m4a to .wav in this thesis.
- OS

  OS python library supports numerous operating system reliant functionalities like listdir () to list all files in directory.
- NumPy

NumPy is python library used for scientific computation.

- Pandas

  Pandas offers user-friendly data structure and data analysis tools. It provides various functionalities for data manipulation and handling through panda data frame.

- Matplotlib.plotly

  This is python library used for visualization of various features to understand data in depth.

3.3.4.3 Spectrum Analysis

Spectrogram is graphical interpretation of potency of various frequency components with respect to time. Spectrogram of audio is also called sonogram; voice gram and it is also called as waterfall when represented in 3D plot. Spectrogram is used in various field such as radar, audio analysis, speech processing.

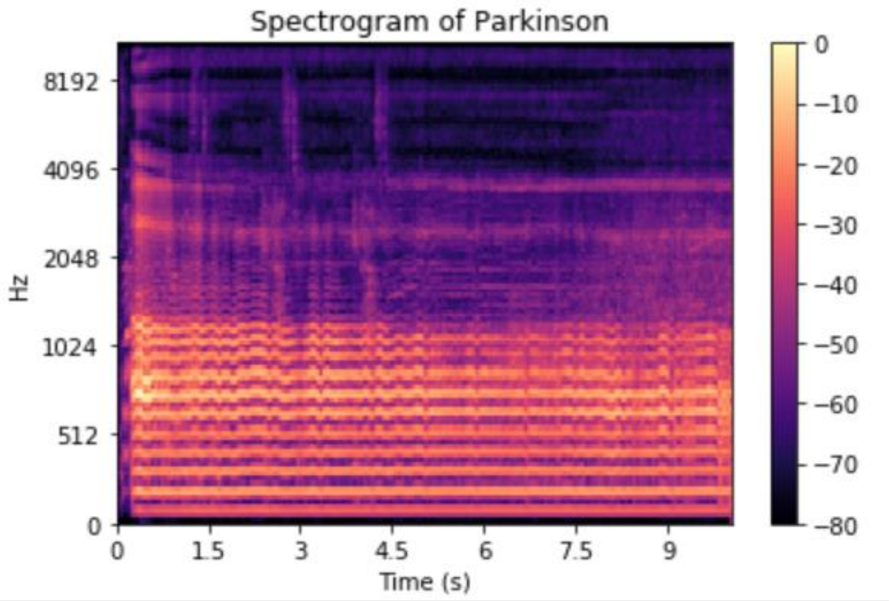


Figure 3:5 Spectrogram

Above figures represents all frequencies that are present in audio at given time. In diagram Y-axis signifies frequency and X-axis signifies time, level of colour is used to describe intensity of energy in signal at given frequency and time. Darker regions in graph has low frequency intensity, orange and yellow represents high frequency intensity in audio.

3.3.4.4 WAVOSAUR Version 1.3.0.0

It was crucial to have deep understanding of features involved in audio file for analysis of prediction of Parkinson disease. WAVOSAUR is a free Windows audio editor that supports VST plugins. The main reason of exploiting it in research is to better understand the meaning of attributes of audio file by translating the.wav file into a 3D spectrogram, audio sonogram, spectrum analysis.

3.3.4.4.1 3D Spectrogram

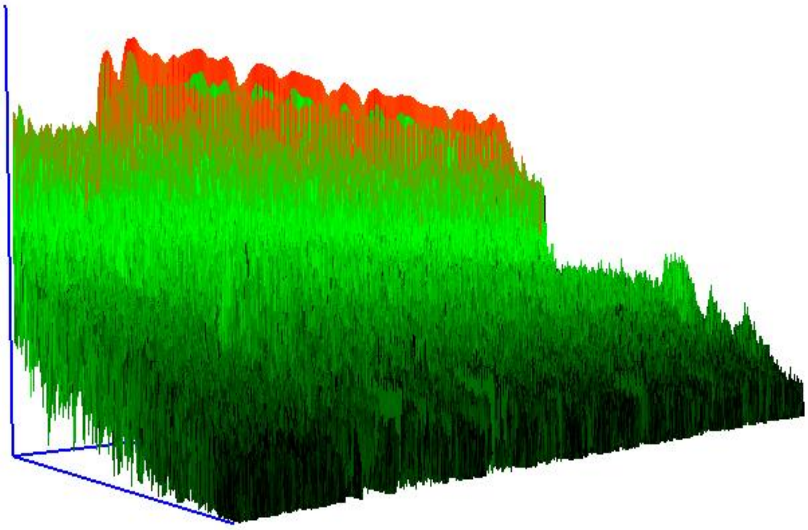


Figure 3:6 Parkinson 3D Spectrogram

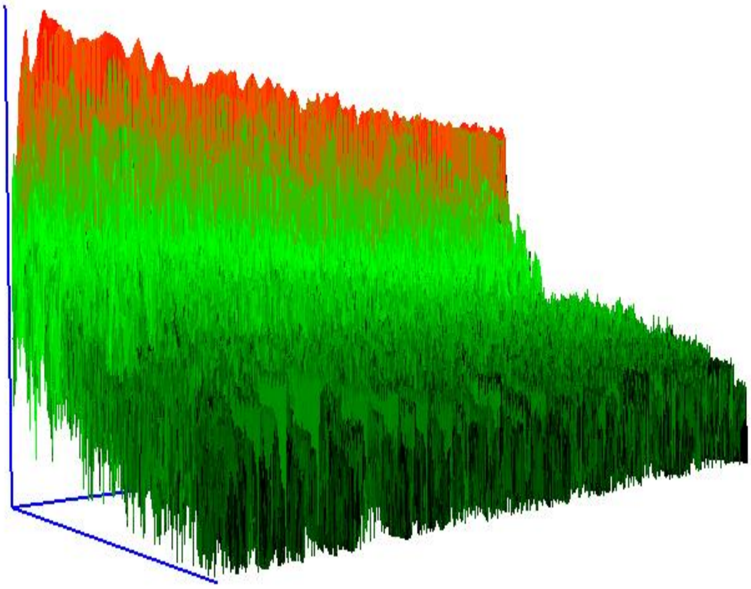


Figure 3:7 Healthy 3D Spectrogram

Audio file is represented in form of time and frequency. Amplitude is represented by height. Higher amplitude is symbolized by brighter colour (i.e. Red Orange) whereas lower amplitude is depicted by darker colour (i.e. Black-Dark Green)

Configuration of WAVOSAUR while creating 3D Spectrogram:

- FFT Size: 4096 Points
- Data: 256 Buffer
- Windowing: Hamming

3.3.4.4.2 Audio Sonogram

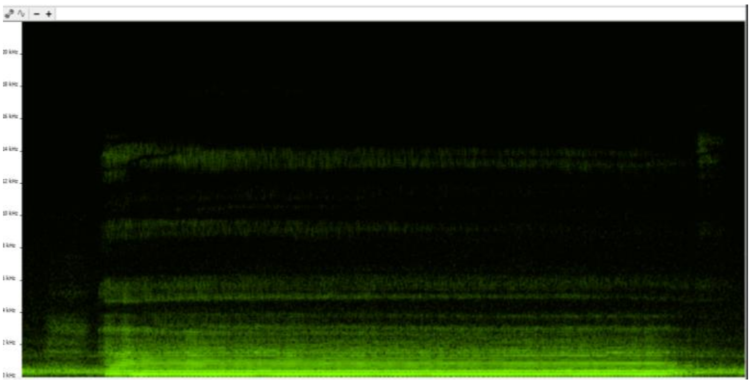


Figure 3:8 Audio Sonogram of Person suffering from Parkinson

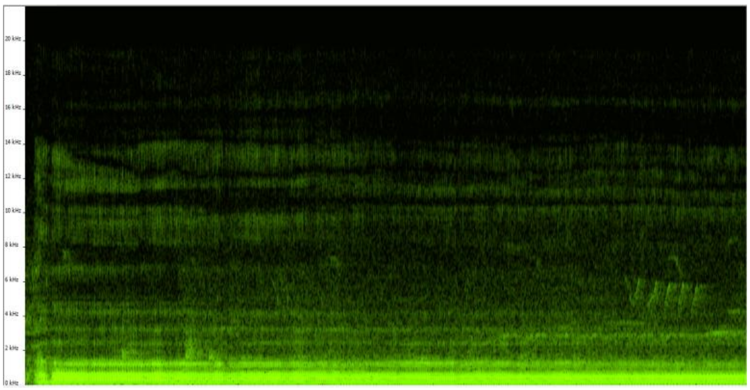


Figure 3:9 Audio Sonogram of Healthy Person

Sonogram can be explained as visual illustrations of a sound's spectral quality, that is, the distribution of signal energy over frequency and how this distribution of energy develops over time. High energy is represented by brighter colour (bright green) whereas low energy is represented by dark colour (Dark Green Black)

3.3.4.4.3 Audio Spectrum Analysis

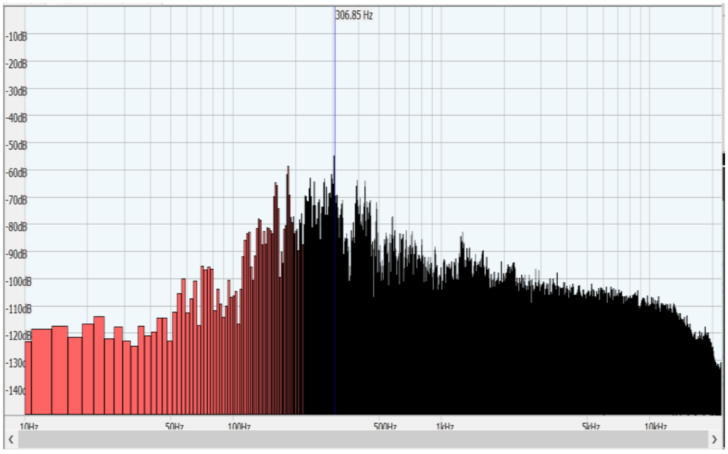


Figure 3:10 Audio Spectrum of Person suffering from Parkinson

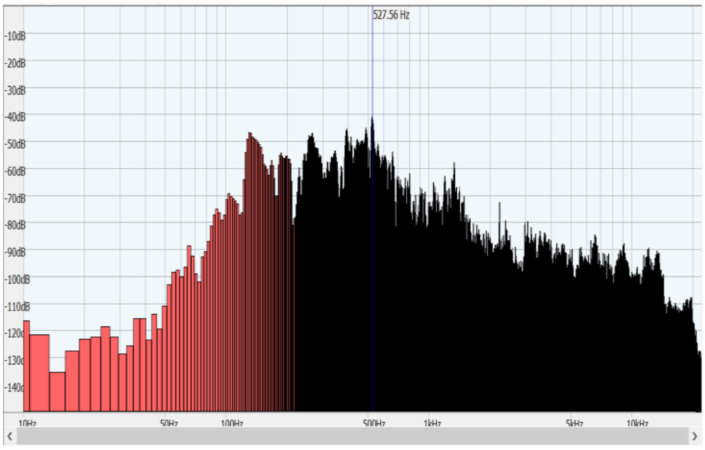


Figure 3:11 Audio Spectrum of Healthy Person

Above diagrams signifies spectrogram of pitch in accordance with frequency. Higher is frequency represents dark colour whereas lower frequency represents lighter colour i.e. pink.

3.3.5 Extracted Features and Description

3.3.5.1 Pitch

Pitch of voice is defined as rate at which vocal folds vibrates. With change in vibration of vocal fold changes the sound of voice, higher the rate of vibration, high is the pitch of voice and thus sound. Thus, pitch can also be defined as highness or lowness of tone perceived by ears. (Reiman, 2019)

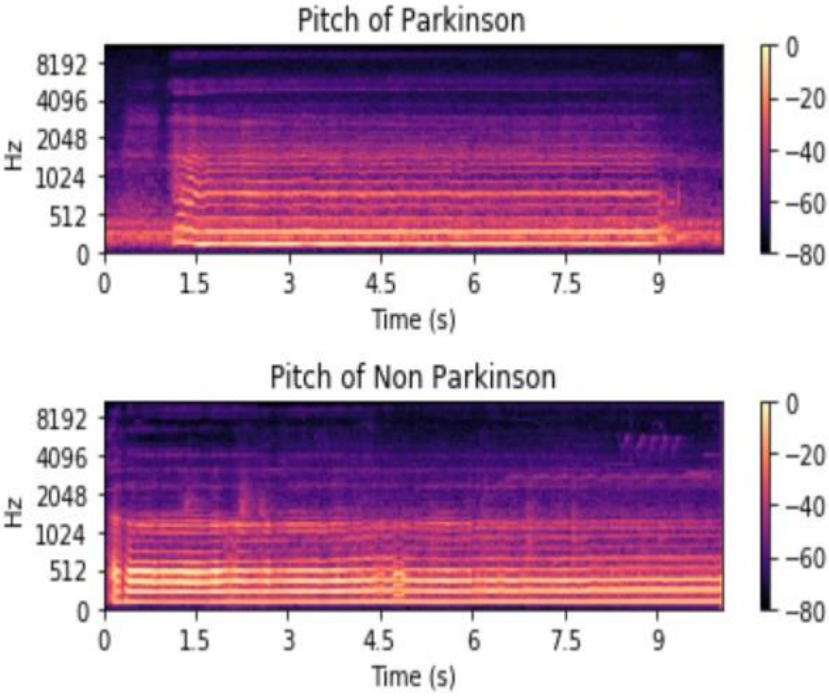


Figure 3:12 Pitch Analysis

As we can see above diagram for healthy people pitch is stable whereas for person suffering from Parkinson pitch is unstable.

3.3.5.2 Jitter and Shimmer

Vocal characteristics can be defined by Jitter and Shimmer of audio signal. Jitter is the cycle-by-cycle frequency variance parameter and Shimmer is the difference in the amplitude of the sound wave (I.C, 2016; João Paulo Teixeira*, 2013). Primary difference between jitter and shimmers is; Jitter consider time cycle whereas shimmer consider maximum peak amplitude of audio signal. Such parameters could be evaluated in a steady voice that constantly generates a vowel. Lack of accountability on vocal cord majorly affects jitter, pathological patients voice has observed to have high jitter. Decreased glottal resilience and mass defects on the vocal cords causes variation in shimmer which is associated with noise production and respiration (João Paulo Teixeira*, 2013).

Jitter (local, absolute): It is average absolute variation among two successive periods also called as jitta (Guimarães, 2007; João Paulo Teixeira*, 2013).

$$\boldsymbol{jitta} = \frac{\mathbf{1}}{\boldsymbol{N-1}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} |\boldsymbol{T_i} - \boldsymbol{T_{i-1}}|$$

Equation 3:1 Jitter (local, absolute) Calculation

Jitter (local): It signifies average variation among two successive periods divided by average time (João Paulo Teixeira*, 2013).

$$jitt = \frac{jitta}{\frac{1}{N}\sum_{i=1}^{N} T_i} \times 100$$

Equation 3:2 Jitter(local) Calculation

Jitter (rap): It illustrate average absolute difference between one period and its two neighbouring periods, divided by an average period (João Paulo Teixeira*, 2013).

$$rap = \frac{\frac{1}{N-1}\sum_{i=1}^{N-1} | T_i - \left(\frac{1}{3}\sum_{n=i-1}^{i+1} T_n\right) |}{\frac{1}{N}\sum_{i=1}^{N} T_i}$$

Equation 3:3 Jitter(rap) Calculation

Where, T_i duration of time for each period, N is number of periods.

Shimmer(local): It illustrate the average absolute variance among the amplitudes of two successive periods, divided by the average amplitude. It is called a shim as well (João Paulo Teixeira*, 2013).

$$shim = \frac{\frac{1}{N-1}\sum_{i=1}^{N-1} |A_i - A_{i-1}|}{\frac{1}{N}\sum_{i=1}^{N} A_i}$$

Equation 3:4 Shimmer(local) Calculation

Shimmer (local, dB): It depicts the median absolute variation of the base 10 logarithm of the disparity between two successive periods and it is also called as ShdB (João Paulo Teixeira*, 2013).

$$shdb = \frac{1}{N-1}\sum_{i=1}^{N-1} \left|20 \times log\left(\frac{A_{i-1}}{A_i}\right)\right|$$

Equation 3:5 Shimmer (local, db.) Calculation

Shimmer (apq3): It illustrate the average fundamental change among the mean amplitudes of its two neighbours and the amplitude of a period, divided by the average amplitude (João Paulo Teixeira*, 2013).

$$apq3 = \frac{\frac{1}{N-1}\sum_{i=1}^{N-1} |\, _i - \left(\frac{1}{3}\sum_{n=i-1}^{i+1} A_n\right) |}{\frac{1}{N}\sum_{i=1}^{N} A_i}$$

Equation 3:6 Shimmer Calculation

3.3.6 Python Function Description

Python script is developed to extract various features from audio using inbuild functions of python library.

- downloadAll ()

  This function is used to downloads survey data and required fields for downloading audio files from synapse database, recordId is one of the columns that act as primary key for downloading audio files from synapse database. Professional diagnoses column is used to label each audio file as Parkinson or NoParkinson.

  Data extracted form downloadAll () function is stored in python panda data frame and for loop is iterated over this data frame to download one audio file at each iteration using downloadByRecordId(recordId) function.

  Various functions are defined to extract features from audio file; librosaMeasures (), pitchHrnMeasures (), jitterShimmerMeasures (), formantMeasures ()

- librosaMeasures ()

  This defined function extract features using Librosa python library inbuild functions. From this function various features were extracted;

- pitchHrnMeasures ()

  Function is defined to return features using praat library. Pitch mean is mean of all pitch values in single with respect to time, in analogous way PitchStdev is pitch standard deviation.

- jitterShimmerMeasures ()

  Jitter and shimmer are crucial factors of audio for analysis of pathological voice. Defined function extract features such as localJitter, localabsoluteJitter, rapJitter, ppq5Jitter, ddpJitter, localShimmer, localdbShimmer using praat python library; explained in section 3.2.5

- Final Script

  This script downloads one audio file for each iteration and extract all features of this file. These extracted features are stored in dataCheck.csv file. As soon as features are extracted and stored in .csv file, downloaded audio file is deleted.

Final script aided to download and extract features of 30000 audio files and store it in .csv file with 30000 rows and 49 columns.

3.4 Modelling: Deep Learning

Based on architecture, neural network can be divided into two categories feed forward network and recurrent network. In feed forward network data is passed only from input node to output nodes, there is no procedure of feedback interconnection available whereas recurrent network contains feedback loop.

Neural network can be built in three stages the two of which are finding the number of input and output units, are quite simple, whereas the third, specification of the number of hidden neurons can become crucial to accuracy of obtained classification results.

Third factor of specification specifies that if number of neurons are unnecessarily high then network will easily learns but poorly generalizes on new data this is called remembering than learning, and if number of neurons are too less there is possibility that neuron in hidden network will never learn relationship between input data. Since there is no precise indicator how many neurons should be used in the construction of a network, it is a common practice to build a network with some initial number of units and when it learns poorly this number is either increased or decreased as required. Obtained solutions are usually task-dependant.

3.4.1 Cross Validation

Validation of model is always important for checking stability of machine learning model. It is necessary to make sure that model got all the patterns from correct data and it is not picking up too much on the noise, or in other words it is low on bias and variance.

Cross validation is valuable technique to evaluate model, to access effectiveness of model, to alleviate overfitting and to determine hyper-parameters. (Gupta, 2017)

1. Split training data into k small subsets
2. Use k-1 subsets for training and remaining one for validation purpose
3. Test data is used for final evolution of model

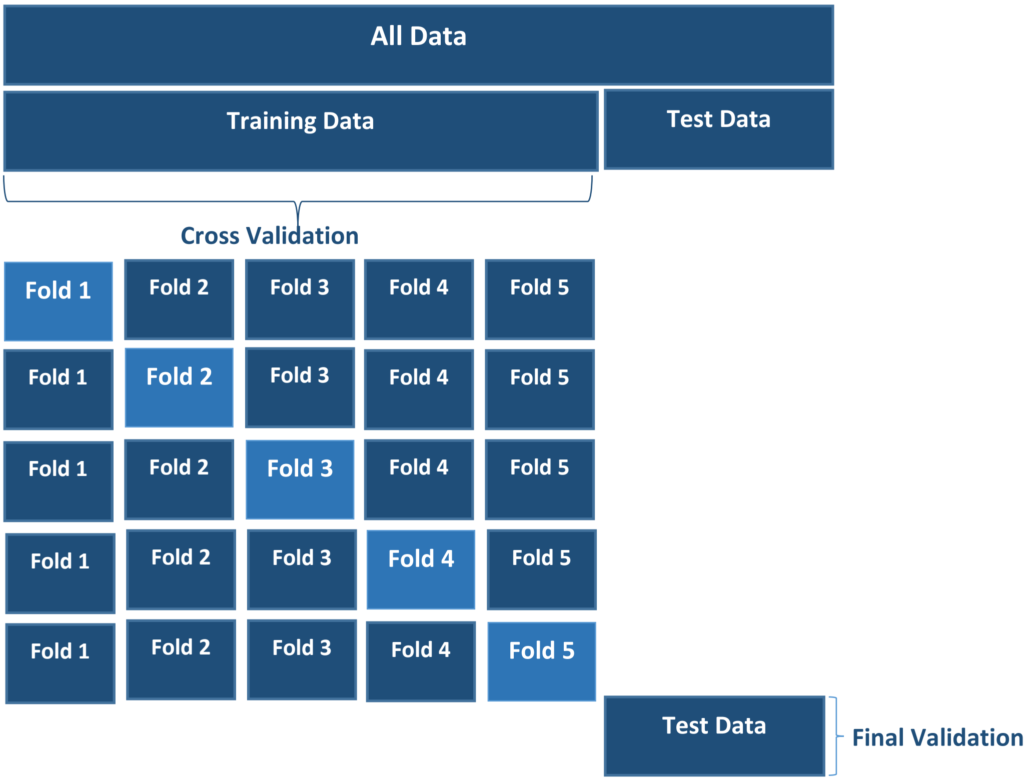


Figure 3:13 Cross Validation

Types of cross validation

Leave out one cross validation (LOOCV), k-fold cross validation, Stratified k-fold cross validation

3.4.2 K-Fold Cross Validation

K-Fold cross validation procedure

1. KFold divides complete data into k samples
2. K-1 samples are used for training
3. Remaining one set is used for testing.

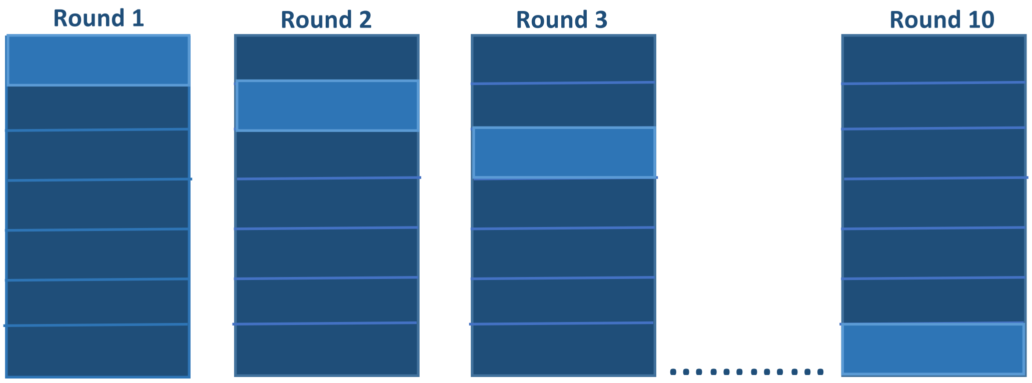


Figure 3:14 KFlod Validation

Sklearn library has one of the methods called cross_val_score which can be used for cross validation of model.

3.4.3 Artificial Neural Network

Artificial Neural Network represents working of human brain, which can be used to solve complex real time problem. It consists of three major layers; Input layer, hidden layers, output layer.

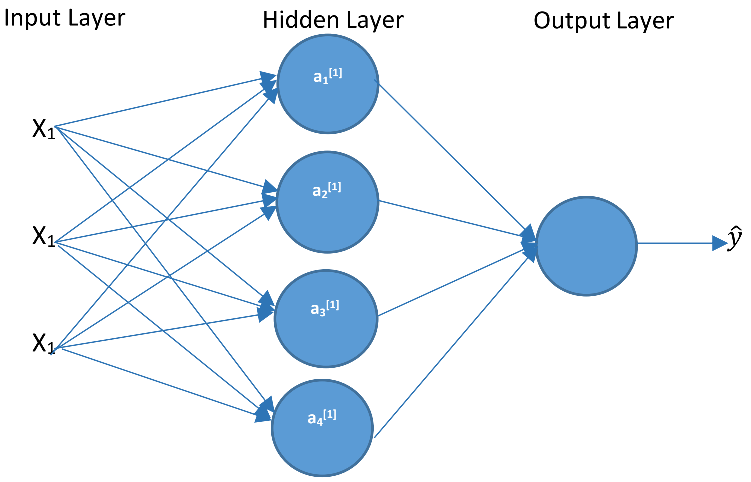


Figure 3:15 Basic Artificial Neural Network Structure

Input Layer: It consist of all the inputs which required to be fed to neural network to obtain desirable output.

Hidden Layer: It is always placed between input layer and output layer. The task of hidden layer is to transform input data into desirable output using weights and biases.

Output layer: This is a last layer of neural network which gives appropriate output in response to input data fed to model.

Each input fed to network is multiplied to corresponding weight matrix and then bias is added to it. To keep obtained values in suitable limit, it is passed to activation function.

$$Z_1^{[1]} = W_1^{[1]} * X + b_1^{[1]}, \qquad a_1^{[1]} = \sigma(Z_1^{[1]})$$

Equation 3:7 Artificial Neural Network Mathematical Representation

$$Where,\ X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix},\ W = Weight\ Matrix, \sigma = Activation\ Fuction$$

Before start to build a model using deep learning it is essential task to pre-process data by cleaning, transformation and feature selection, one of the important step of pre-processing data is scaling dataset by making mean of all input to zero and variance to one. This process of scaling makes sure that there is no bias due to different scales of input data. To scaling data, MinMaxScaler function of Sklearn pre-processing library of python is used.

3.4.3.1 Architecture of ANN

Let us understand how it works on features of audio for prediction of Parkinson disease.

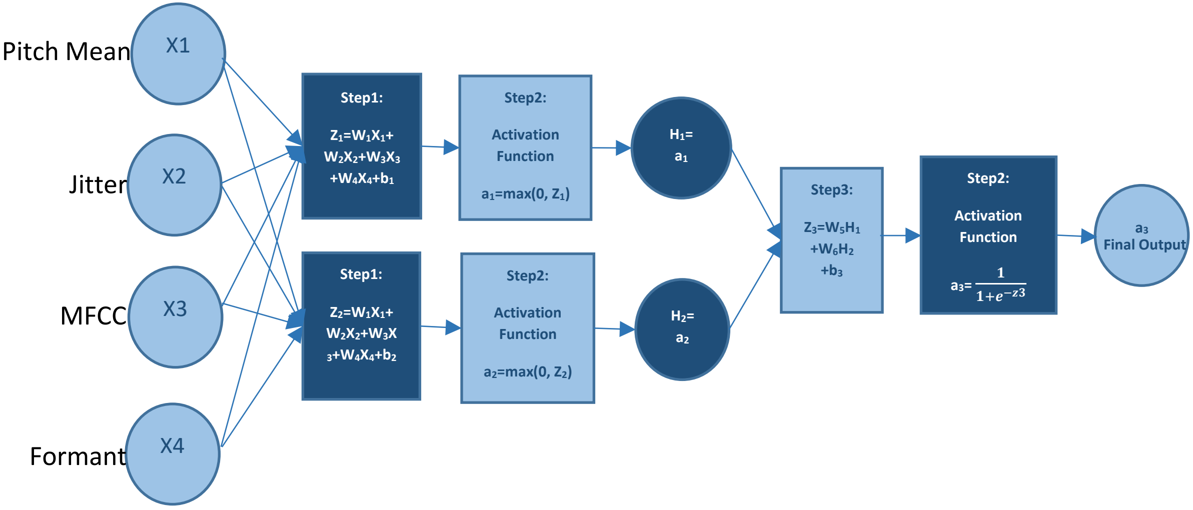


Figure 3:16 ANN on Extracted Features

Activation function performs non linear transformation on input data in order to make model learn and execute complicated job. Model is build using two activation functions; ReLu(Rectified Linear) activation function for hidden layer and sigmoid activation function for output layer.
ReLu has simple calculation, it returns one as long as Z is positive and return zero for all negative values. Sigmoid Activation function produces output between zero and one, it is used in output layer for binary classification.

After training model for 400 epochs we got accuracy of 86% along with 87.60% precision and 85.70% recall value on test data.

3.4.4 H2O Deep Neural Network

H2O is open source used for building machine learning models on big data. It supports multiple data sources such as local file system, remote file system HDFS, S3 and some relational data bases. H2O has various data manipulation is build function commonly used during data preparation some of them are Importing Multiple Files, Combining Columns from Two Datasets, Combining Rows from Two Datasets, Fill NAs, Group By, Imputing Data, Merging Two Datasets, Pivoting Tables, Replacing Values in a Frame, Slicing Columns, Splitting Datasets into Training/Testing/Validating. It supports supervised as well as unsupervised learning. (H2O.ai, 2019)

Supervised learning algorithms supported by H2O

- Naïve Base
- Distributed Random Forest
- Gradient Boost Machine
- Generalized Linear
- Deep Neural Network

Deep Neural Network (DNN)

H20 deep neural network is also called as ANN or MPL which are only types of network supported by H2O. It is based on concept of feed forward network. It can consist of substantial number of neurons making multiple hidden layers with various activation function; tanh, rectifier, maxout. Newly developed features in H2O DNN are l1, l2 regularization, dropout, grid search, checkpointing (H2O.ai, 2019).

3.4.4.1 Input Parameters

There are various parameters which should be known before building H2O Deep Neural Network.

- Model Id: It is ID for model, used as reference. By default, H2O generates model id if not specified.
- N Folds: Used to specify number of cross validation folds
- Keep cross validation prediction: By enabling this option we can preserve cross validation prediction.
- Y: It is dependant variables, can be numeric or categorical.
- X: Independent variables
- Ignore const cols: Determines if constant training columns are to be overlooked, as no details can be gained from them. By default, this option is available.
- Score each iteration: Determines whether to grade a training model at each iteration.
- Balance classes: Determine if the minority classes should be oversampled to match the distribution of the population. It can used only for classification problems; it is not enabled by default.
- Class factor ratio: It is applicable only to classification as when balance classes is enabled. It determines under sample or over sample ratio of all columns, if not specified it automatically calcite this ratio.
- Overwrite with best model: Determine if to replace the current model to best model determined while training, this operation is based on stopping matrix. By default, this option is enabled.
- Activation: Several types of activations are supported by H2O for hidden layers (Tanh, Tanh with dropout, Rectifier, Rectifier with dropout, Maxout, Maxout with dropout).
- Hidden: It gives size of hidden layers and number of hidden layers. Numbers specified in hidden layers should not be negative
- Epochs: Specifies number of times dataset should be iterated.
- Seed: It makes components of algorithm to be dependent on random number generator seed for randomization.
- l1: Specify the L1 regularization to add stability and improve generalization; sets the value of many weights to 0.
- l2: Specify the L2 regularization to add stability and improve generalization; sets the value of many weights to smaller values.
- Loss: H2O support various loss functions such as Automatic, CrossEntropy, Quadratic, Huber, or Absolute. If not specified explicitly it takes automatic loss function.
  Loss functions for classification and regression
  - Classification: CrossEntropy, Quadratic, Huber, Absolute
  - Regression: Quadratic, Absolute, Huber
- Distribution: H2O has various distributions; huber, gaussian, bernoulli, multinomial, poisson, laplace, AUTO, quantile, tweedie. Bernoulli and multinomial are used for categorical response column, whereas remaining all are used for numeric response column (H2O.ai, 2019).

3.4.4.2 Model Building

H2o.init() is used to initialize h2o instance.
Model:

```
model= H2ODeepLearningEstimator(
                                hidden=[50,50,50],
                                activation = "rectifier",
                                epochs = 10,
                                distribution="bernoulli",
                                loss='CrossEntropy',
                                nfolds=10,
                                l2=1e-3,
                                l1=1e-5,
                                mini_batch_size=1)
model.train(X, y, training_frame = train, validation_frame = valid)
```

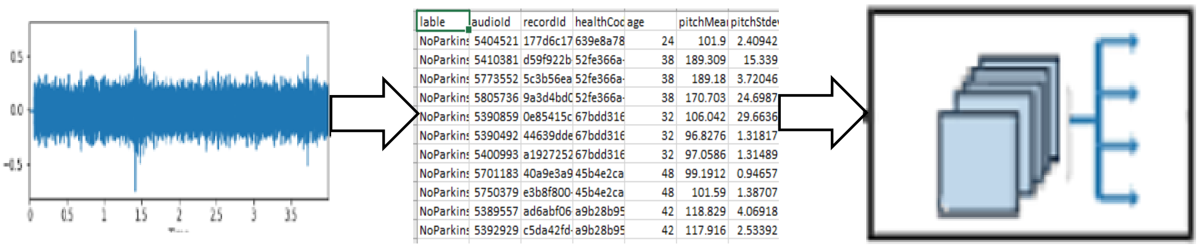
Build model consist of three hidden layers with 50 neurons each. Rectifier is activation function which acts same as ReLu activation function that is transforming input in range of -1 to 1. We are using nfolds parameter to specify number of folds for cross validation. Cross Entropy is loss function used to reduce prediction error. L1 and L2 are used to regularize the output in simple words it is used to reduce overfitting.

recall value on test data.

3.4.5 Convolution Neural Network

Convolution neural network is one of the primary machine learning algorithms used for object detection, image recognition, image classification as well currently used for sequential data. In Parkinson's disease prediction audio features are fed as input to CNN algorithm and then processed to classify into Parkinson or no Parkinson groups. Features of audio are perceived as two-dimensional matrix of features; h × w where, h is height, w is width. Each feature is transferred through sequence convolution layer with filters, pooling and fully connected network and finally is passed through activation function such as sigmoid to achieve probabilistic value between zero and one.

Process below explains how CNN work on audio features.



lable	audioId	recordId	healthCoc	age	pitchMea	pitchStde
NoParkins	5404521	177d6c17	639e8a78	24	101.9	2.40942
NoParkins	5410381	d59f922b	52fe366a-	38	189.309	15.339
NoParkins	5773552	5c3b56ea	52fe366a-	38	189.18	3.72046
NoParkins	5805736	9a3d4bd0	52fe366a-	38	170.703	24.6987
NoParkins	5390859	0e85415c	67bdd316	32	106.042	29.6636
NoParkins	5390492	44639dde	67bdd316	32	96.8276	1.31817
NoParkins	5400993	a1927252	67bdd316	32	97.0586	1.31489
NoParkins	5701183	40a9e3a9	45b4e2ca	48	99.1912	0.94657
NoParkins	5750379	e3b8f800-	45b4e2ca	48	101.59	1.38707
NoParkins	5389557	ad6abf06-	a9b28b95	42	118.829	4.06918
NoParkins	5392929	c5da42fd-	a9b28b95	42	117.916	2.53392

Audio Signal **Extracted Features** **CNN Model**

3.4.5.1 Architecure of CNN

Convolution neural network consist of three main layers Convolution layer, Pooling layer, Fully connected layer. Convolution layer has weighted filter which is convoluted through input data and dot product of weighted matrix and feature matrix is passed to pooling layer. Average pooling layer will perform down sampling by choosing average value from dot product matrix, this output is passed to fully connected network to transform it to give desirable results.

Output volume is controlled by three hyperparameters

- Number of filters(F)
- Strides(S)
- Zero Padding(P)

Simple formula to calculate output volume is

$$Output\ Volume = \frac{W - F + 2P}{S} + 1$$

Equation 3:7 CNN Output Volume

In this research we have used conv1d convolution layers with tanh activation function. In total CNN model consist of five layers 2 convolution layers, one average pooling layer and two dense layers constituting fully connected network.

3.5 Evaluation

Evaluation of machine learning model is essential task as it assists in truly judging a model. Classification accuracy is universally used parameter for measuring performance of the model, but it is not enough to evaluate model truly.

3.5.1 Confusion Matrix

Confusion matrix is matrix which is frequently used to portray performance of machine learning classification models (Mishra, 2018).

	Actually Parkinson	Actually NoParkinson
Predicted Parkinson	True Positive	False Positive
Predicted NoParkinson	False Negative	True Negative

Table 3:2 Confusion Matrix

True Positive: People are predicated to have Parkinson and they do suffer from Parkinson.

False Negative: People are predicted to not have Parkinson and they do not have Parkinson.

True Negative: People are predicated to have Parkinson, but they do not have Parkinson.

False Positive: People are predicated to not have Parkinson, but they do have Parkinson.

3.5.2 Classification Accuracy

Classification accuracy is ratio of total number of correct predictions to total number of data points used as input (Mishra, 2018)..

$$Accuracy = \frac{Total\ Number\ of\ Correct\ Prediction}{Total\ Number\ of\ \ Input\ Data\ Points}$$

Equation 3:8 Accuracy Calculation

3.5.3 Area Under Curve (AUC)

AUC is wildly used in classification problem; it is scale of separability of two classes. Higher is AUC high is correct prediction of ones as ones and zeros as zeros. Higher value of model indicates better performance of model in prediction (Mishra, 2018).

AUC is plot of false positive vs true positive rate which lies between 0 to 1.

True Positive Rate (Sensitivity): Sensitivity is defined as positive data points which are predicted as positive divided by total positive data points (Mishra, 2018)..

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Equation 3:9 Sensitivity Calculation

Specificity: Specificity is defined negative data points which are predicted as negative data points divided by total negative data points.

$$Specificity = \frac{True\ Negative}{True\ Negative + False\ Positive}$$

Equation 3:10 Specificity Calculation

False Positive Rate: It is defined as negative data points which are predicted as positive data points divided by total sum of true negative, false positive.

$$False\ Pasitive\ Rate = \frac{False\ Positive}{True\ Negative + False\ Positive}$$

Equation 3:11 False Positive Rate Calculation

It can also be written as,

$$False\ Positive\ Rate = 1 - Specificity$$

Equation 3:12 False Positive Rate in relation with Specificity

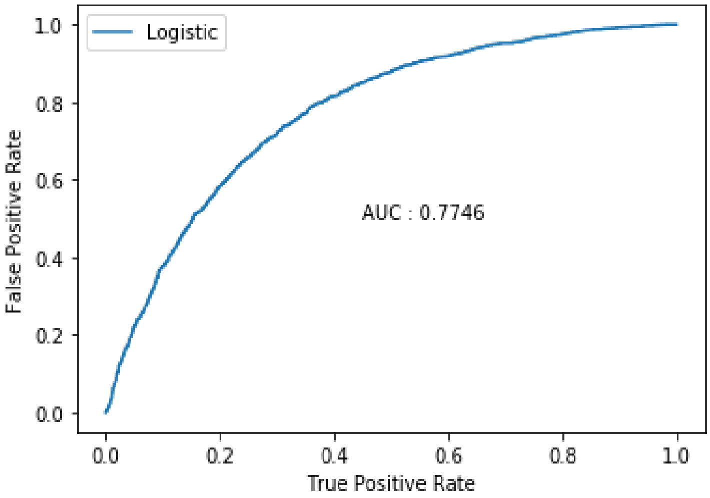


Figure 3:17 Area Under Cover for Logistic Regression

3.5.4 F1 Score

F1 is harmonic mean of recall and precision, it lies between zero and one. F1 score describes how accurately model classify data. Higher the value of F1 score better is the performance model (Mishra, 2018)..

$$F1\ Score = 2 \times \frac{1}{\frac{1}{Precision} + \frac{1}{recall}}$$

Equation 3:13 F1 Score Calculation

Precision: It can be defined as ratio of total number of positive correct prediction to positive results of prediction.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Equation 3:14 Precision Calculation

Recall: It can be defined as ratio of total number of positive data points predicted correctly to total positive data points

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Equation 3:15 Recall Calculation

3.5.5 Models used for evaluation

Machine Algorithm Name	Narrative	Category
Logistic Regression	Sklearn Logistic Model	Supervised (Classification)
GLM	Rapid Miner Core	Supervised (Regression)
Random Forest	Sklearn Ensemble Model	Supervised (Regression/Classification)
K Nearest Neighbour	Sklearn KNN Classifier Model	Supervised (Classification)
Deep Neural Network	H20	Deep Neural Network
Naïve Base	Rapid Miner Studio Core	Supervised (Classification)
Decision Tree	Sklearn Decision Tree Model	Supervised (Regression/Classification)
Artificial Neural Network	Keras	Supervised (Regression/Classification)
Convolution Neural Network	Keras	Supervised (Regression/Classification)

Table 3:3 Models Description

All above mentioned models are compared based on different parameters discussed in next section; which are used to evaluate machine learning models.

Chapter 4: Results - Model Comparison

This chapter consist of comparison of various traditional and deep learning models which are intensively used in fields of data science

4.1Comparison of Machine Learning Models

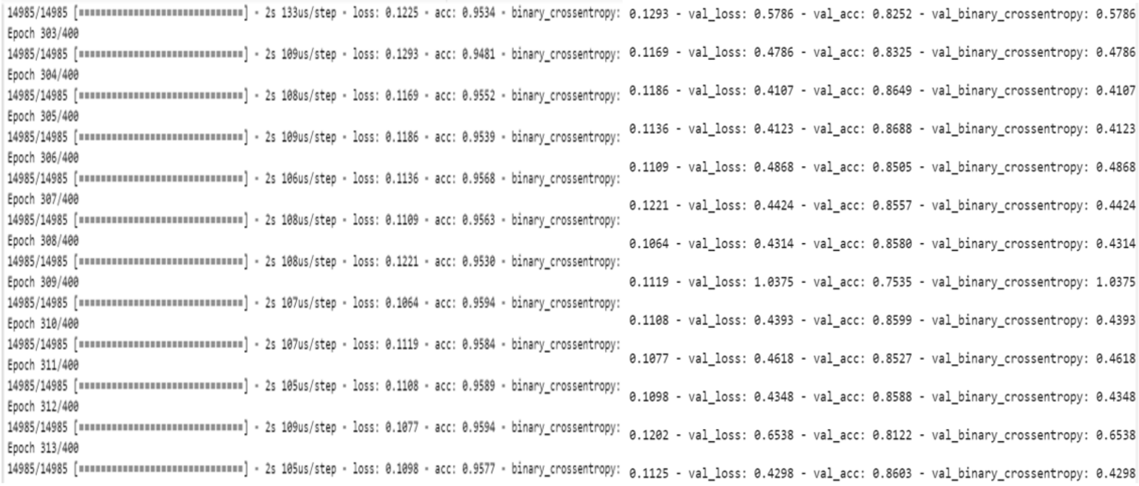
Model Name	Accuracy	AUC	Precision	Recall	F1 Score	Sensitivity	Specificity	Classification Error
Naïve Bayes	65.195%	0.72	59.69%	88.94%	69.40%	88.94	42.41	34.81
Logistic Regression	70.91%	0.77	72.31	71.29	71.49	71.29	70.51	29.09
Decision Tree	74.39	0.76	74.28	75.38	74.49	75.38	73.43	25.61
KNN	72.07	0.78	74.32	75.33	74.64	75.33	73.43	27.93
Random Forest	82.67	0.90	88.37	80.60	83.80	80.60	86.43	17.33
GLM	50.01%	0.79	49.38	99.62	65.55	99.62	2.77	49.99
Artificial Neural Network	86%	0.93	87.60%	85.70%	86.63%	85.70	86.89	14
Deep Neural Network (H2O)	86.12%	0.93	87.66	87.75	87.70	87.75	83.39	13.88
Convolution Neural Network	85.11%	0.92	87.81	71.89	79.05	71.89	79.16	22

Table 4:1 Comparison of Machine Learning Models

It is evident from above table that Deep Learning models are well performing and well balanced when compared to other algorithms.

4.2 Results of Artificial Neural Network

4.2.1 Execution Window



```
14985/14985 [==============================] - 2s 133us/step - loss: 0.1225 - acc: 0.9534 - binary_crossentropy: 0.1293 - val_loss: 0.5786 - val_acc: 0.8252 - val_binary_crossentropy: 0.5786
Epoch 303/400
14985/14985 [==============================] - 2s 109us/step - loss: 0.1293 - acc: 0.9481 - binary_crossentropy: 0.1169 - val_loss: 0.4786 - val_acc: 0.8325 - val_binary_crossentropy: 0.4786
Epoch 304/400
14985/14985 [==============================] - 2s 108us/step - loss: 0.1169 - acc: 0.9552 - binary_crossentropy: 0.1186 - val_loss: 0.4107 - val_acc: 0.8649 - val_binary_crossentropy: 0.4107
Epoch 305/400
14985/14985 [==============================] - 2s 109us/step - loss: 0.1186 - acc: 0.9539 - binary_crossentropy: 0.1136 - val_loss: 0.4123 - val_acc: 0.8688 - val_binary_crossentropy: 0.4123
Epoch 306/400
14985/14985 [==============================] - 2s 106us/step - loss: 0.1136 - acc: 0.9568 - binary_crossentropy: 0.1109 - val_loss: 0.4868 - val_acc: 0.8505 - val_binary_crossentropy: 0.4868
Epoch 307/400
14985/14985 [==============================] - 2s 108us/step - loss: 0.1109 - acc: 0.9563 - binary_crossentropy: 0.1221 - val_loss: 0.4424 - val_acc: 0.8557 - val_binary_crossentropy: 0.4424
Epoch 308/400
14985/14985 [==============================] - 2s 108us/step - loss: 0.1221 - acc: 0.9530 - binary_crossentropy: 0.1064 - val_loss: 0.4314 - val_acc: 0.8580 - val_binary_crossentropy: 0.4314
Epoch 309/400
14985/14985 [==============================] - 2s 107us/step - loss: 0.1064 - acc: 0.9594 - binary_crossentropy: 0.1119 - val_loss: 1.0375 - val_acc: 0.7535 - val_binary_crossentropy: 1.0375
Epoch 310/400
14985/14985 [==============================] - 2s 107us/step - loss: 0.1119 - acc: 0.9584 - binary_crossentropy: 0.1108 - val_loss: 0.4393 - val_acc: 0.8599 - val_binary_crossentropy: 0.4393
Epoch 311/400
14985/14985 [==============================] - 2s 105us/step - loss: 0.1108 - acc: 0.9589 - binary_crossentropy: 0.1077 - val_loss: 0.4618 - val_acc: 0.8527 - val_binary_crossentropy: 0.4618
Epoch 312/400
14985/14985 [==============================] - 2s 109us/step - loss: 0.1077 - acc: 0.9594 - binary_crossentropy: 0.1098 - val_loss: 0.4348 - val_acc: 0.8588 - val_binary_crossentropy: 0.4348
Epoch 313/400
14985/14985 [==============================] - 2s 105us/step - loss: 0.1098 - acc: 0.9577 - binary_crossentropy: 0.1202 - val_loss: 0.6538 - val_acc: 0.8122 - val_binary_crossentropy: 0.6538
                                                                                                            0.1125 - val_loss: 0.4298 - val_acc: 0.8603 - val_binary_crossentropy: 0.4298
```

Figure 4:1 Execution Screen

Figure 4:1 is execution screen illustrate that we are at step 303 out of total 400 epochs, we can see step number at epoch keyword to get idea how long it would take to finish execution. Binary cross entropy is a loss function which learns to decrease prediction error. It is difference between probability distribution of training data to correct label, and its value should decrease with process of execution. As we can see there are two accuracies and binary cross entropy values as data is split into training, validation, and testing subset to avoid risk of the network to memorise inputs. Having validation subset, it makes sure that during training phase model has unseen data to avoid memorising problem.

4.2.2 Correlation between accuracy and binary cross entropy

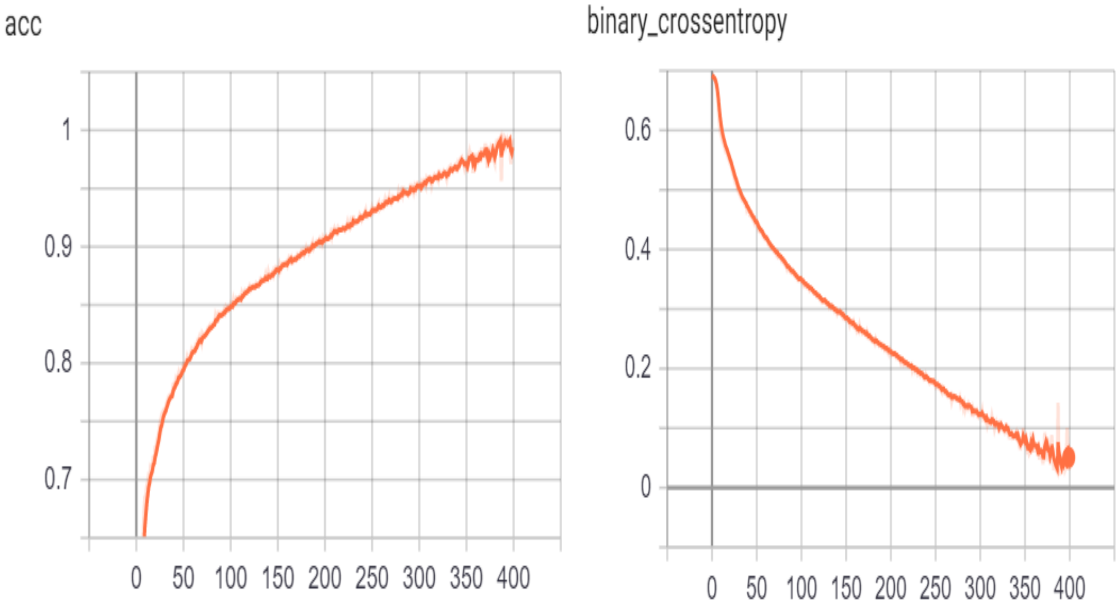


Figure 4:2 Correlation between binary cross entropy and Accuracy

Figure 7:6 depict graph of binary cross entropy and accuracy with number of epochs. It shows influence of binary cross entropy on accuracy with epochs. With decrease in binary cross entropy accuracy increases in accordance with iterations of epoch.

4.2.3 ROC of ANN

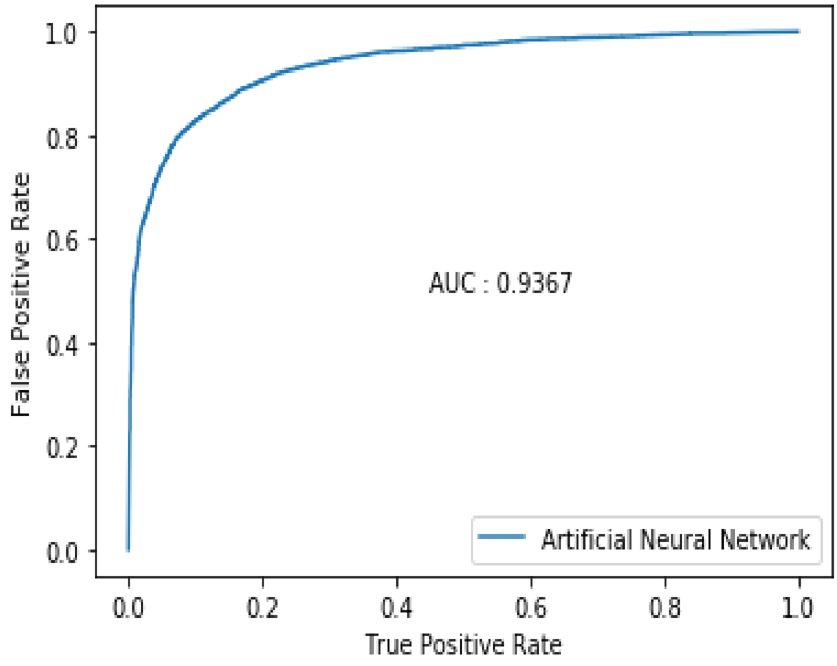


Figure 4:3 ROC of ANN

4.2.4 Resultant Confusion Matrix

	Actually Parkinson	Actually NoParkinson	Class Precision
Predicted Parkinson	4084	578	87.60%
Predicated NoParkinson	681	3833	84.91%
Class Recall	85.70%	86.68%	

Table 4:2 Confusion Matrix of ANN

4.3 Results of Deep Neural Network H2O

4.3.1 Execution Window

```
deeplearning Model Build progress: |██████████████████████████████████████████████████████| 100%
```

4.3.2 ROC of H2O DNN Model

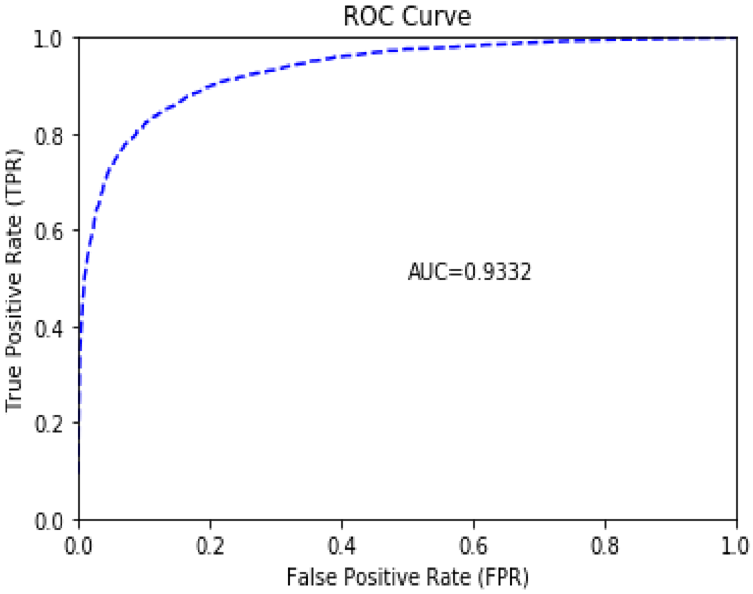


Figure 4:4 ROC of H2O DNN

4.3.3 Resultant Confusion Matrix

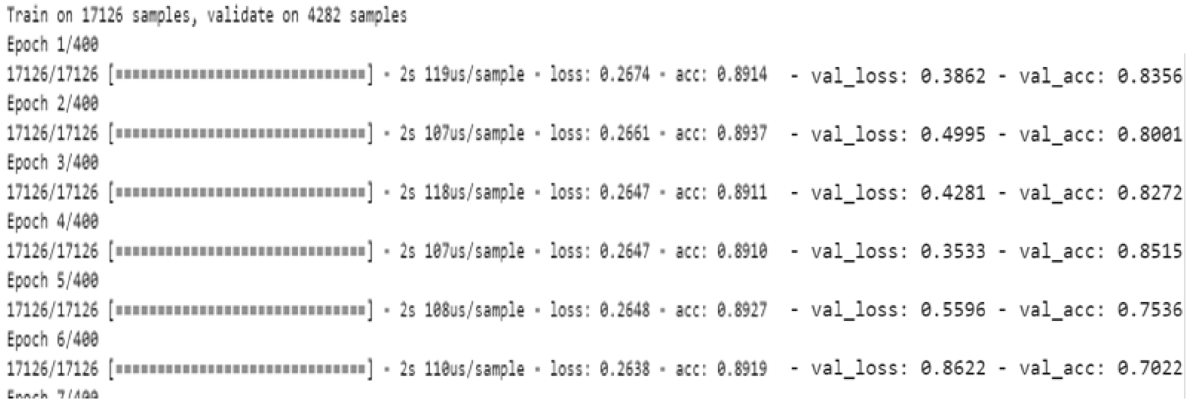
	Actually Parkinson	Actually NoParkinson	Class Precision
Predicted Parkinson	**2566**	**361**	**87.66%**
Predicated NoParkinson	**511**	**2586**	**83.35%**
Class Recall	**87.75%**	**83.39%**	

Table 4:3 Confusion Matrix for H2O DNN Model

After training model for 10 epochs we got accuracy of 86.12% along with 87.66% precision and 87.75%

4.4 Results of Convolution Neural Network

4.4.1 Execution Window



```
Train on 17126 samples, validate on 4282 samples
Epoch 1/400
17126/17126 [==============================] - 2s 119us/sample - loss: 0.2674 - acc: 0.8914  - val_loss: 0.3862 - val_acc: 0.8356
Epoch 2/400
17126/17126 [==============================] - 2s 107us/sample - loss: 0.2661 - acc: 0.8937  - val_loss: 0.4995 - val_acc: 0.8001
Epoch 3/400
17126/17126 [==============================] - 2s 118us/sample - loss: 0.2647 - acc: 0.8911  - val_loss: 0.4281 - val_acc: 0.8272
Epoch 4/400
17126/17126 [==============================] - 2s 107us/sample - loss: 0.2647 - acc: 0.8910  - val_loss: 0.3533 - val_acc: 0.8515
Epoch 5/400
17126/17126 [==============================] - 2s 108us/sample - loss: 0.2648 - acc: 0.8927  - val_loss: 0.5596 - val_acc: 0.7536
Epoch 6/400
17126/17126 [==============================] - 2s 110us/sample - loss: 0.2638 - acc: 0.8919  - val_loss: 0.8622 - val_acc: 0.7022
Epoch 7/400
```

Figure 4:5 Execution Screen of CNN Model

Above figure depicts that model is trained on 17126 sample and 4282 samples are used for validation. It is showing training and validation accuracy and loss values for given input samples for 400 epochs.

4.4.2 ROC of CNN Model

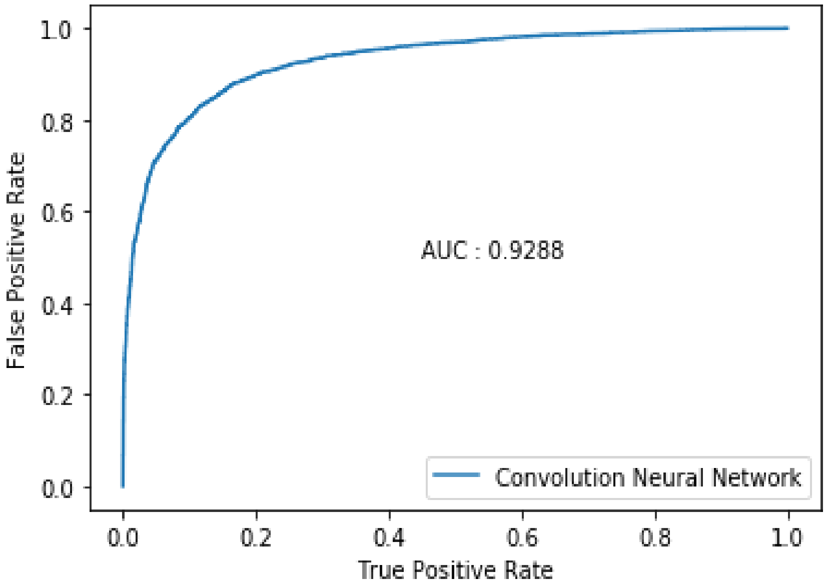


Figure 4:6 ROC of CNN

4.4.3 Resultant Confusion Matrix

	Actually Parkinson	Actually NoParkinson	Class Precision
Predicted Parkinson	2471	400	87.71%
Predicated NoParkinson	966	3539	78.55%
Class Recall	71.89%	89.84%	

Table 4:4 Confusion Matrix for CNN Model

After training model for 100 epochs we got accuracy of 78% along with 80% precision and 76.92% recall value on test data.

4.5 Resulting Confusion Matrices

4.5.1 Naïve Bayes

	Actually Parkinson	Actually NoParkinson	Class Precision
Predicted Parkinson	3804	2569	59.69%
Predicated NoParkinson	473	1892	80.00%
Class Recall	88.94%	42.41%	

Table 4:5 Confusion Matrix for Naïve Bayes

4.5.2 GLM

	Actually Parkinson	Actually NoParkinson	Class Precision
Predicted Parkinson	4246	4352	49.38%
Predicated NoParkinson	16	124	88.57%
Class Recall	99.62%	2.77%	

Table 4:6 Confusion Matrix for GLM

4.5.3 Logistic Regression

	Actually Parkinson	Actually NoParkinson	Class Precision
Predicted Parkinson	2265	867	72.31%
Predicated NoParkinson	912	2073	69.44%
Class Recall	71.29%	70.51%	

Table 4:7 Confusion Matrix for Logistic regression

4.5.4 Decision Tree

	Actually Parkinson	Actually NoParkinson	Class Precision
Predicted Parkinson	2327	805	73.24%
Predicated NoParkinson	760	2225	74.53%
Class Recall	75.38%	73.43%	

Table 4:8 Confusion Matrix for Decision Tree

4.5.5 K Nearest Neighbour

	Actually Parkinson	Actually NoParkinson	Class Precision
Predicted Parkinson	2328	804	74.32%
Predicated NoParkinson	762	2223	74.47%
Class Recall	75.33%	73.43%	

Table 4:9 Confusion Matrix for KNN

4.5.6 Random Forest

	Actually Parkinson	Actually NoParkinson	Class Precision
Predicted Parkinson	2768	364	88.37%
Predicated NoParkinson	666	2319	77.68%
Class Recall	80.60%	86.43%	

Table 4:10 Confusion Matrix for Random Forest

4.6Visualization

4.6.1 The Receiver Operating Characteristic (ROC)

ROC is a graphical representation of performance of model with False Positive Rate on X-axis and True Positive Rate on Y-axis. Higher the value of AUC high is the performance of model.

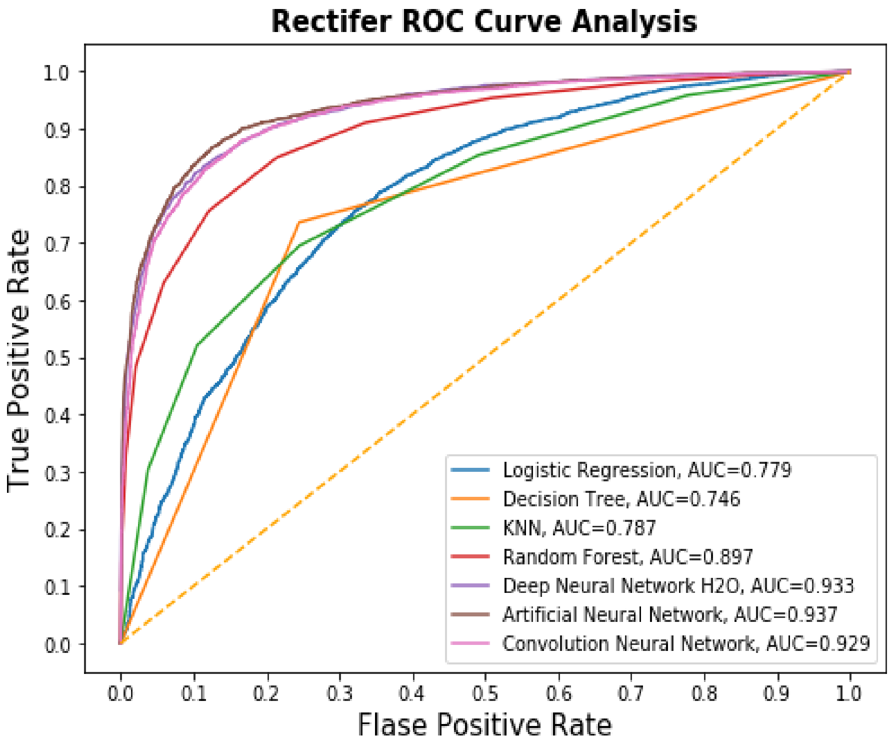


Figure 4:7 ROC of Models

4.6.2 Sensitivity vs Specificity

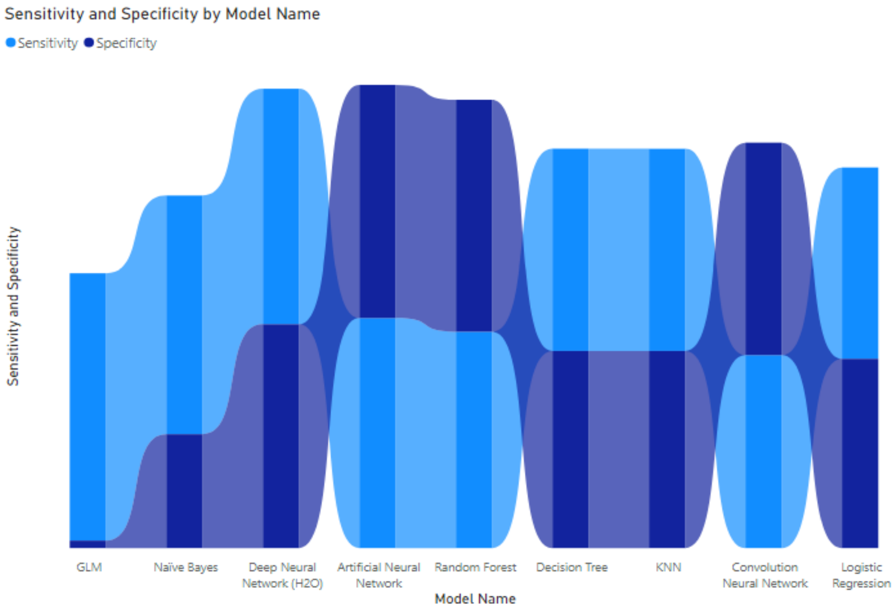


Figure 4:8 Sensitivity Vs Specificity

4.6.3 Recall and Precision

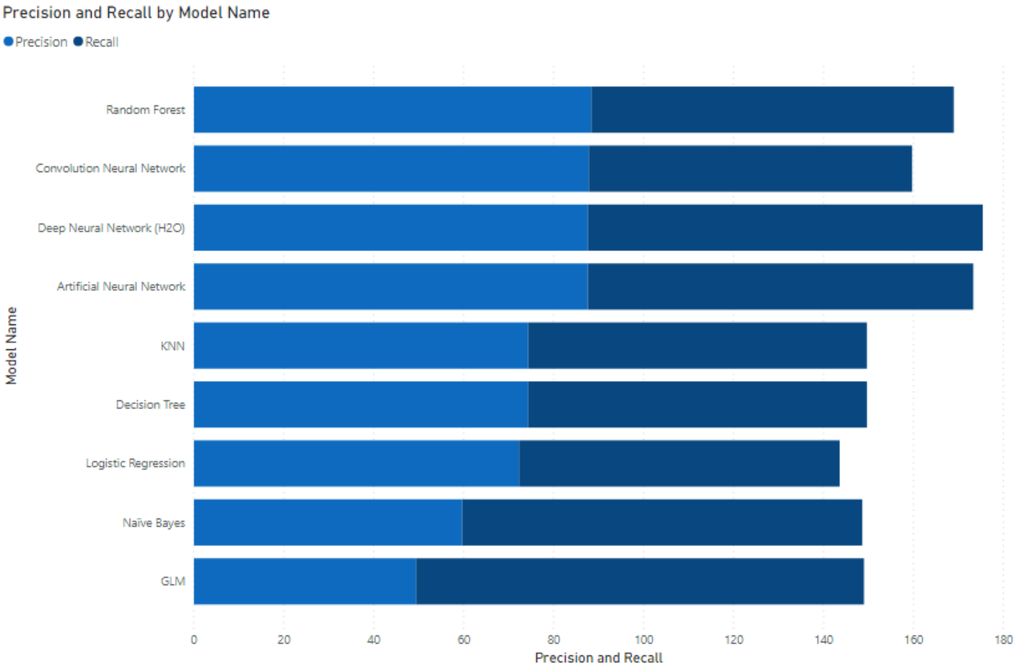


Figure 4:9 Recall and Precision

4.6.4 Accuracy comparison of Model in bar plot

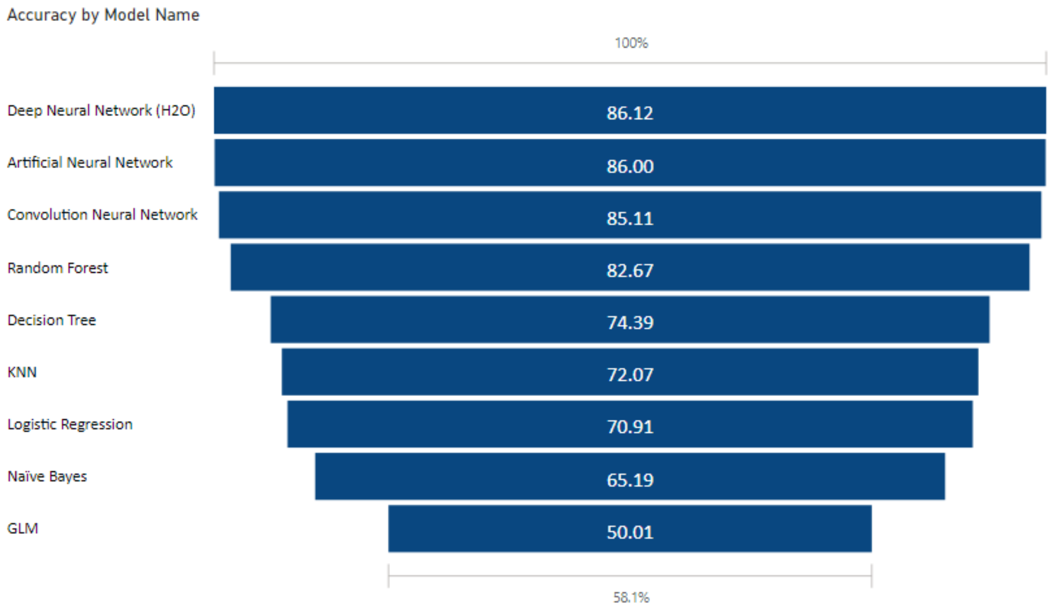


Figure 4:10 Compare Accuracies of Model

4.7Observation

From table 4:1 it is observed that Deep learning using H2O is outperforming over all other algorithm for audio processing with highest accuracy 86.12% and lowest classification error 13.88% and with AUC 0.93.

Chapter 5: Discussion and Conclusion

In this chapter will discuss and answer all the research questions using specific chapters, tables, sections, and figure references

5.1 Discussion

Vocal feature change is an earliest characteristic of Parkinson's disease thus audio recordings can be used to analyse disease, but currently PET-CT imaging test is used to determine reduction in dopaminergic neurons. This testing process is expensive and thus reduces rate of detecting disease. Thus, in order to increase ease and make diagnoses inexpensive this thesis focuses on building machine learning model for diagnosis of Parkinson's disease.

In order to achieve objective of research ,(T. J. Wroge, 2018) paper is considered as base paper of research, they have compared models trained on AVEC features 1200 unique features and GeMaps 62 features. According analysis AVEC feature trained model outperformed compared to GeMaps features trained model as there is more information encoded within the feature vectors for AVEC compared to GeMaps.

In this research main goal was to use features that are most affected by Parkinson disease and acquire higher accuracy using deep learning. With 42 features extracted from audio and developed deep learning model we were able to acquire accuracy of 86.12% and area under curve 0.93

Developed deep learning models along with random forest outperforms on six wisely used other machine learning algorithms in case of accuracy, precision, specificity, sensitivity as depicted in Table 4:1 Comparison of Machine Learning Models.

5.2 Research Question Answers

Question 1: How audio is affected in people suffering from Parkinson disease?

In Chapter 1:, I have explained what is Parkinson Disease and in section 1.1.2 using reference to paper (M. Wodzinski, 2019) have explain which features of audio are affected and how are they affected due to Parkinson disease.

Question 2: How to extract features of audio file in image format and in numeric(.csv) format?

In Chapter 3:Research Methodology, to extract Jitter, Shimmer, Pitch, MFCC features from audio, multiple python functions are defined. These functions are using various inbuild python libraries like Librosa and PRAAT which are described in section 3.3.5 'Extracted Features and Description' and 3.3.6 Section describes functioning of all self-defined function used to extract features. Description of all these features is given in section 3.3.4.

Question 3: Which algorithms will work better in classification; Traditional or Deep Learning algorithms?

On comparing results Table 4:1 Comparison of Machine Learning Models it is evident that deep learning algorithms Artificial Neural Network, Deep Neural Network H2O, Convolution Neural Network along with random forest outperform when compared to traditional algorithms like Naïve Base, GLM, Logistic Regression.

Question 4: Is there scope of growth in audio processing for Parkinson classification?

Yes, from Table 4:1 Comparison of Machine Learning Models and figures it is evident that self-developed deep learning model; Deep Neural Network H2O is well performing with accuracy 86.12%. Thus, there is scope of improvement in field audio processing for disease recognition using deep learning.

Question 5: How can we apply this work in real-time: health care institutes?

Model is build using Google colab which is working well, and model is saved which can be used in flask web application in hospitals. This web application can be used to take patients voice as input and classify as person having Parkinson or no.

Chapter 6: Future Work

Various neurological disorders have impact on vocal cords and thus affects speech. Comparing traditional models and deep learning models on audio data for classification it is evident that deep learning models outperform. Thus, future work could be as follows

- Use deep learning for classification multiple neurological disorders which affects speech.
- To increase accuracy of currently working model we can use other parameters such stability, handwriting
- Could use convolution neural network on spectrogram images of audio data to check how it works in prediction.

References

A. Bourouhou, A. J. C. N. a. A. H., 2016. Comparison of Classification Methods to Detect the. *2nd International Conference on Electrical and Information Technologies ICEIT,* pp. 421-424.

Bolat, S. Ç. a. B., 2016. Diagnosis of Parkinson's disease by using ANN. *International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC),* pp. 119-121.

Can, M., 2013. Neural Networks to Diagnose the Parkinson's Disease. *SOUTHEAST EUROPE JOURNAL OF SOFT COMPUTING,* pp. vol. 2, no. 1.

Das, R., 2010. A comparison of multiple classification methods for diagnosis of Parkinson disease. *Expert Systems with Applications,* 37(2), pp. 1568-1572.

Downward, E., 2017. *ParkinsonsDisease.net.* [Online]
Available at: https://parkinsonsdisease.net/symptoms/speech-difficulties-changes/
[Accessed March 2017].

E. J. Alqahtani, F. H. A. H. F. S. a. S. O. O., 2018. Classification of Parkinson's Disease Using NNge Classification Algorithm. *1st Saudi Computer Society National Computer Conference (NCC), Riyadh,* pp. pp. 1-7.

Frid, A. &. H. H. &. H. D. &. M. L. &. R. L. &. S. S., 2014. Computational Diagnosis of Parkinson's Disease Directly from Natural Speech Using Machine Learning Techniques. Proceedings. *2014 IEEE International Conference on Software Science, Technology and Engineering,.*

Frid, A. &. K. A. &. S. D. &. M. L., 2016. Diagnosis of Parkinson's Disease from Continuous Speech using Deep Convolutional Networks without Manual Selection of Features. *IEEE International Conference on the Science of Electrical Engineering (ICSEE), Eilat,* pp. pp 1-4.

Guimarães, I., 2007. A Ciência e a Arte da Voz Humana. *Escola Superior de Saúde de Alcoitão,* p. .

Gupta, P., 2017. *Towards Data Science.* [Online]
Available at: https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f

H. Hazan, D. H. L. M. L. O. R. a. S. S., 2012. Early diagnosis of Parkinson's disease via machine learning on speech data. *IEEE 27th Convention of Electrical and Electronics Engineers in Israel, Eilat,* pp. pp. 1-4.

H2O.ai, 2019. *H2O.ai.* [Online]
Available at: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/starting-h2o.html
[Accessed 16 Dec 2019].

Ho, A. &. I. R. &. M. C. &. B. J. &. G. S., 1998. "Speech Impairment in a Large Sample of Patients with Parkinson's Disease". *Behavioural neurology,* pp. pp 131-137.

I.C, Z. &. F. R. &. R. T. &. S. D., 2016. Digital signal processing in the differential diagnosis of benign larynx diseases. *SCIENTIA MEDICA,* Volume 16, p. pp 109.

Jadou, Y., 2019. *praat-parselmouth 0.3.3.* [Online]
Available at: https://pypi.org/project/praat-parselmouth/

João Paulo Teixeira*, C. O. C. L., 2013. Vocal Acoustic Analysis - Jitter, Shimmer and HNR Parameters. *Elsevier Ltd,* p. 1112 – 1122 .

Krishnan, A. R. a. S., 2017. "Feature analysis of dysphonia speech for monitoring Parkinson's disease". *39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Seogwipo,* pp. pp. 2308-2311.

M. Wodzinski, A. S. D. H. J. R. O.-A. a. E. N., 2019. Deep Learning Approach to Parkinson’s Disease Detection Using Voice Recordings and Convolutional Neural Network Dedicated to Image Classification,. *41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC),* pp. pp. 717-720..

Mishra, A., 2018. *Towards Data Science.* [Online]
Available at: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

Ozkan, H., 2016. A Comparison of Classification Methods for Telediagnosis of Parkinson's Disease. *Entropy,* March, Volume 18, p. 115.

Reiman, T., 2019. *Body Language University.* [Online]
Available at: http://www.bodylanguageuniversity.com/public/203.cfm

S. Aich, H. K. K. y. K. L. H. A. A. A.-A. a. M. S., 2019. A Supervised Machine Learning Approach using Different Feature Selection Techniques on Voice Datasets for Prediction of Parkinson’s Disease. *2019 21st International Conference on Advanced Communication Technology (ICACT), PyeongChang Kwangwoon_Do, Korea (South),* pp. pp. 1116-1121.

S. Aich, K. Y. K. L. H. A. A. A.-A. a. M. S., 2018. A nonlinear decision tree based classification approach to predict the Parkinson's disease using different feature sets of voice data. *20th International Conference on Advanced Communication Technology (ICACT), Chuncheon-si Gangwon-do, Korea (South),* pp. pp. 638-642.

scikit-learn, 2007 - 2019. *scikit-learn developers.* [Online]
Available at: https://scikit-learn.org/stable/modules/cross_validation.html

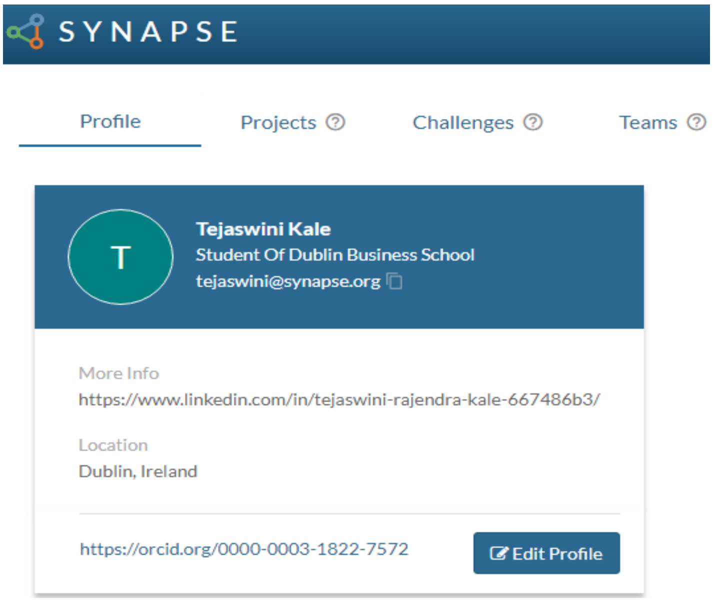
T. J. Wroge, Y. Ö. C. D. D. S. D. C. A. a. R. H. G., 2018. "Parkinson’s Disease Diagnosis Using Machine Learning and Voice". *IEEE Signal Processing in Medicine and Biology Symposium (SPMB), Philadelphia, PA,* pp. pp. 1-7.

Zhijing Xu, J. W. Y. Z. X. H., 2018. Voiceprint recognition of Parkinson patients based on deep learning. *ArXiv,* Volume 1812.06613.

Appendix

Appendix A Documents for Data Access

Account Creation:

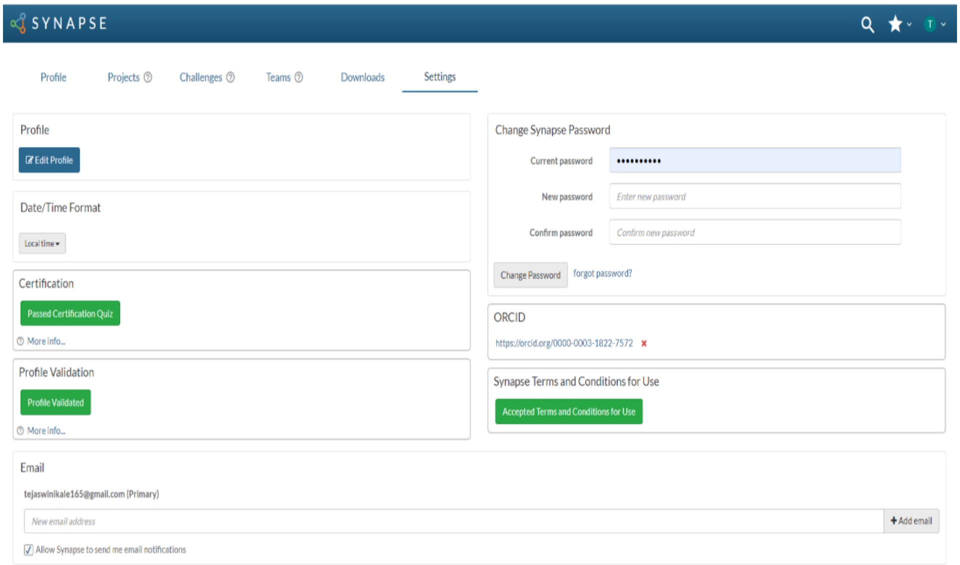


Tejaswini Kale
Student Of Dublin Business School
tejaswini@synapse.org

More Info
https://www.linkedin.com/in/tejaswini-rajendra-kale-667486b3/

Location
Dublin, Ireland

https://orcid.org/0000-0003-1822-7572



Profile

Date/Time Format
Local time ▾

Certification
Passed Certification Quiz

Profile Validation
Profile Validated

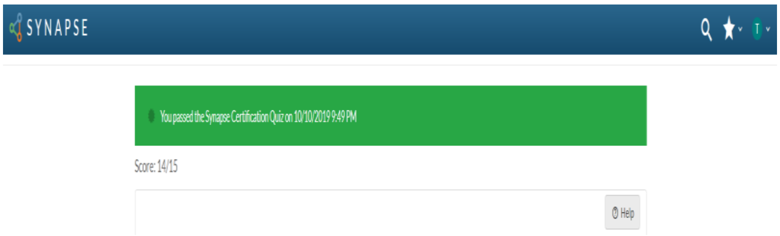
Change Synapse Password
Current password ••••••••••
New password Enter new password
Confirm password Confirm new password

ORCID
https://orcid.org/0000-0003-1822-7572

Synapse Terms and Conditions for Use
Accepted Terms and Conditions for Use

Email
tejaswinikale165@gmail.com (Primary)
New email address
☑ Allow Synapse to send me email notifications

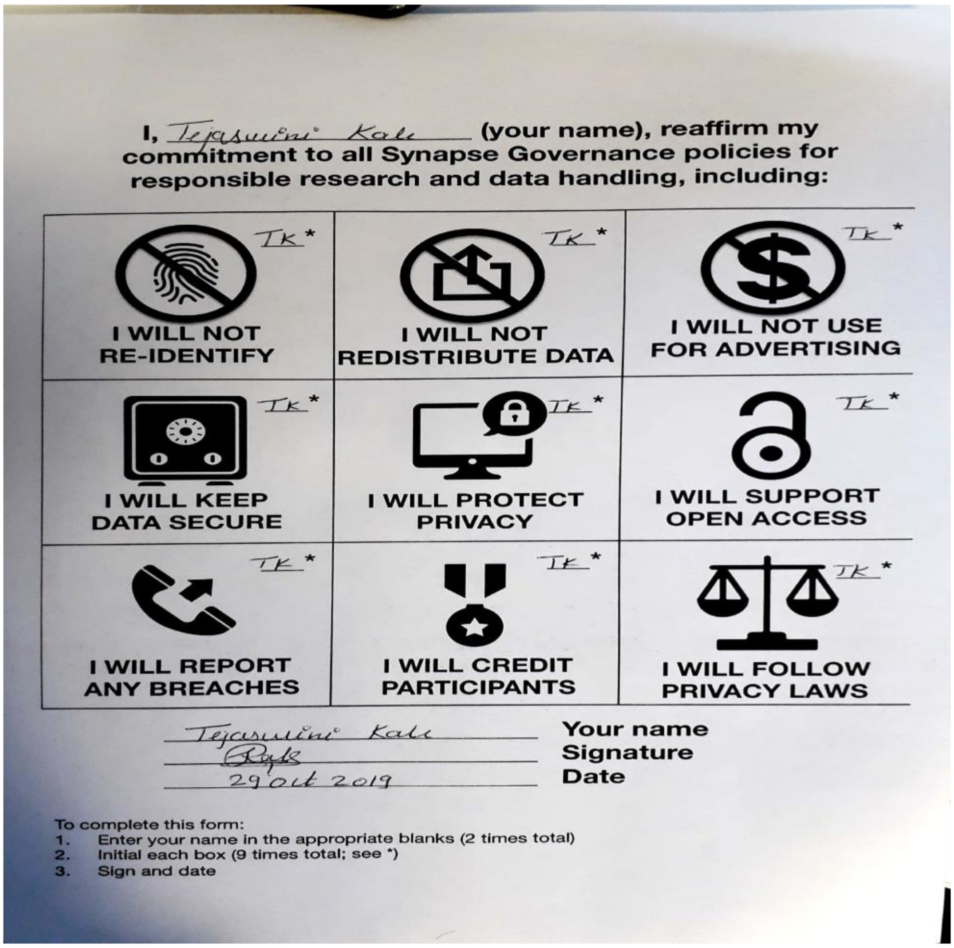
Become Certified User



You passed the Synapse Certification Quiz on 10/10/2019 9:49 PM

Score: 14/15

Oath Document



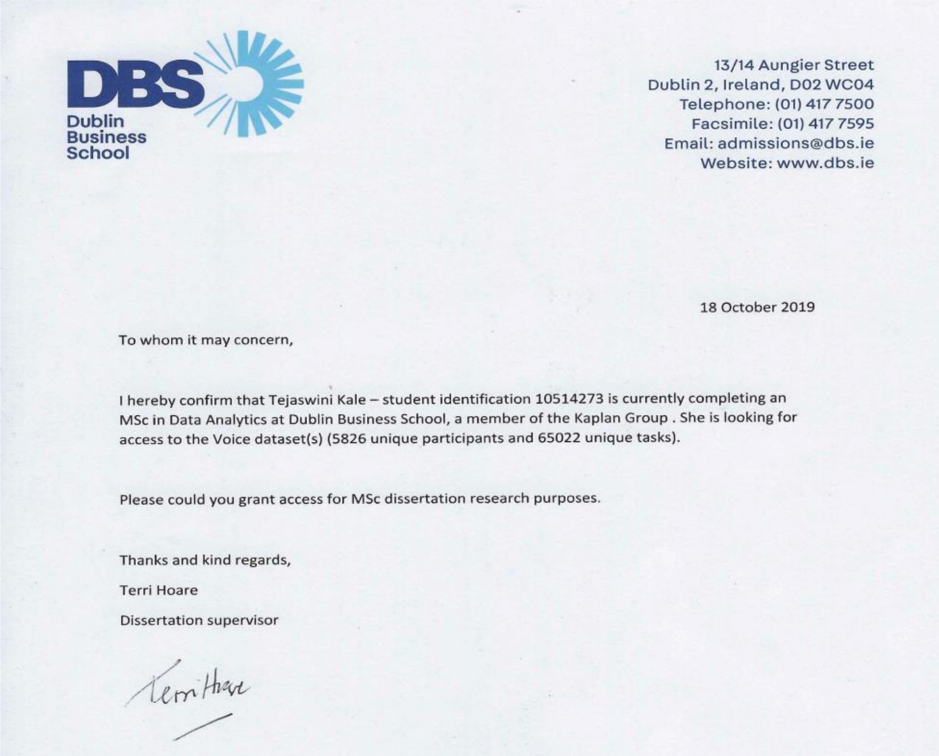
I, Tejaswini Kale (your name), reaffirm my commitment to all Synapse Governance policies for responsible research and data handling, including:

TK * I WILL NOT RE-IDENTIFY	TK * I WILL NOT REDISTRIBUTE DATA	TK * I WILL NOT USE FOR ADVERTISING
TK * I WILL KEEP DATA SECURE	TK * I WILL PROTECT PRIVACY	TK * I WILL SUPPORT OPEN ACCESS
TK * I WILL REPORT ANY BREACHES	TK * I WILL CREDIT PARTICIPANTS	TK * I WILL FOLLOW PRIVACY LAWS

Tejaswini Kale Your name
Kale Signature
29 oct 2019 Date

To complete this form:
1. Enter your name in the appropriate blanks (2 times total)
2. Initial each box (9 times total; see *)
3. Sign and date

Letter From College



DBS
Dublin
Business
School

13/14 Aungier Street
Dublin 2, Ireland, D02 WC04
Telephone: (01) 417 7500
Facsimile: (01) 417 7595
Email: admissions@dbs.ie
Website: www.dbs.ie

18 October 2019

To whom it may concern,

I hereby confirm that Tejaswini Kale – student identification 10514273 is currently completing an MSc in Data Analytics at Dublin Business School, a member of the Kaplan Group . She is looking for access to the Voice dataset(s) (5826 unique participants and 65022 unique tasks).

Please could you grant access for MSc dissertation research purposes.

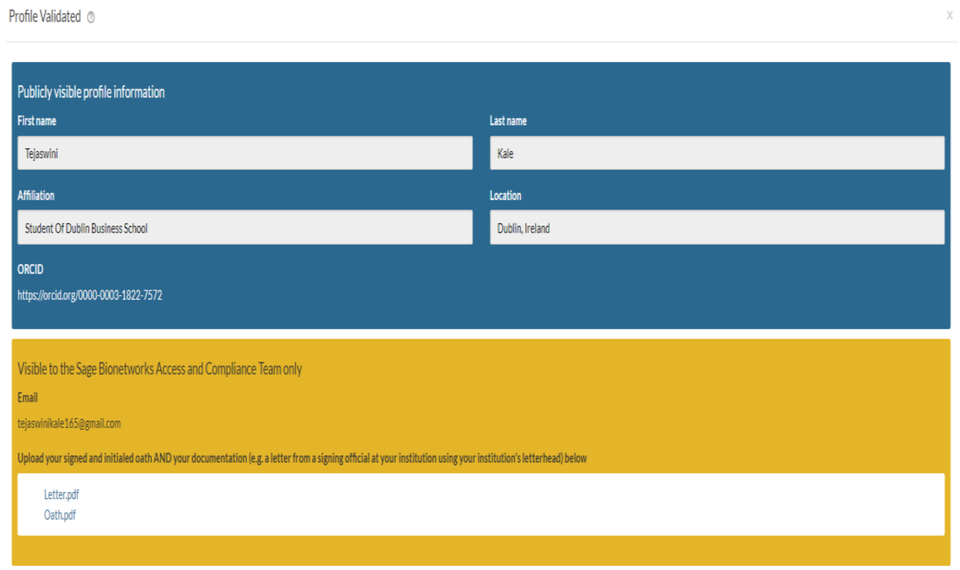
Thanks and kind regards,

Terri Hoare

Dissertation supervisor

Terri Hoare

Document Submited



Profile Validated

Publicly visible profile information

First name: Tejaswini

Last name: Kale

Affiliation: Student Of Dublin Business School

Location: Dublin, Ireland

ORCID

https://orcid.org/0000-0003-1822-7572

Visible to the Sage Bionetworks Access and Compliance Team only

Email

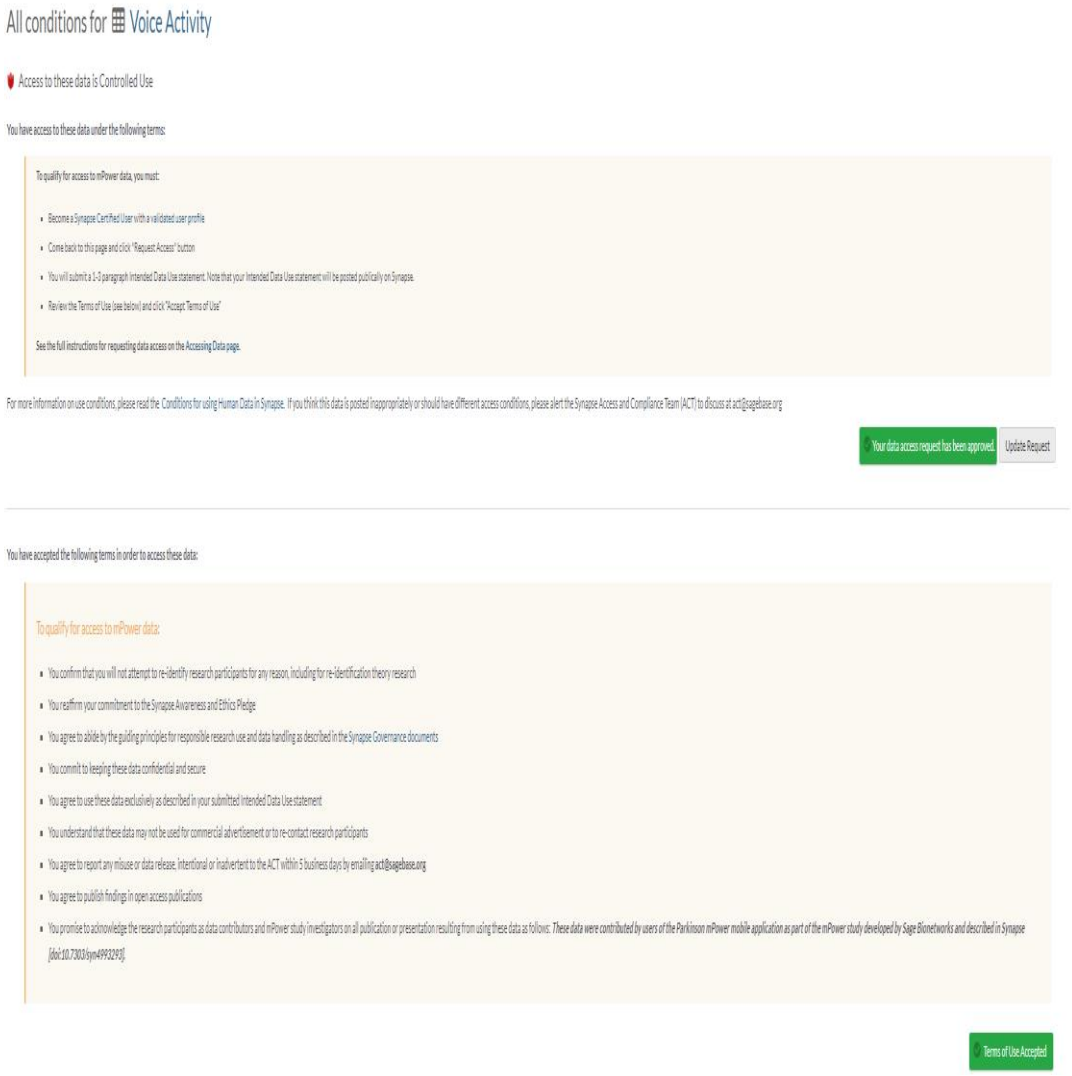
tejaswinikale165@gmail.com

Upload your signed and initialed oath AND your documentation (e.g. a letter from a signing official at your institution using your institution's letterhead) below

Letter.pdf

Oath.pdf

Accept Terms and Condition for data access



All conditions for Voice Activity

Access to these data is Controlled Use

You have access to these data under the following terms:

To qualify for access to mPower data, you must:

- Become a Synapse Certified User with a validated user profile
- Come back to this page and click "Request Access" button
- You will submit a 1-3 paragraph Intended Data Use statement. Note that your Intended Data Use statement will be posted publically on Synapse.
- Review the Terms of Use (see below) and click "Accept Terms of Use"

See the full instructions for requesting data access on the Accessing Data page.

For more information on use conditions, please read the Conditions for using Human Data in Synapse. If you think this data is posted inappropriately or should have different access conditions, please alert the Synapse Access and Compliance Team (ACT) to discuss at act@sagebase.org

You have accepted the following terms in order to access these data:

To qualify for access to mPower data:

- You confirm that you will not attempt to re-identify research participants for any reason, including for re-identification theory research
- You reaffirm your commitment to the Synapse Awareness and Ethics Pledge
- You agree to abide by the guiding principles for responsible research use and data handling as described in the Synapse Governance documents
- You commit to keeping these data confidential and secure
- You agree to use these data exclusively as described in your submitted Intended Data Use statement
- You understand that these data may not be used for commercial advertisement or to re-contact research participants
- You agree to report any misuse or data release, intentional or inadvertent to the ACT within 5 business days by emailing **act@sagebase.org**
- You agree to publish findings in open access publications
- You promise to acknowledge the research participants as data contributors and mPower study investigators on all publication or presentation resulting from using these data as follows: ***These data were contributed by users of the Parkinson mPower mobile application as part of the mPower study developed by Sage Bionetworks and described in Synapse*** *[doi:10.7303/syn4993293].*

Appendix B Artefact

Model Results

Artificial Neural Network

ANNModel.h5: ANN model is build using training data. This model is saved to be used in future using python inbuild functions.

Deep Neural Network (H2O)

DNNH2OModel: DNN model is build using h2o and it is trained using training dataset. This model is saved to use in future.

Convolution Neural Network (H2O)

CNNModel: CNN model is build using data extracted from audio files. This model is saved to use in future.

Python file

Parkison_Prediction.ipynb: It is primary python code used for extraction of features and for building models and testing of new data.

Power BI Visualization

PowerBI_Visualization: It is Power BI visualisation file which contain three sheets which are used in main report to represent accuracies of models, stacked bar for precision and recall, ribbon chart for specificity and sensitivity.

Referred Papers

All referred papers are present in research_papers folder.

Readme

It explains contents of artefact and gives explanation of complete python code

Appendix C Self-Developed Python Script

```
def downloadAll():
     results = syn.tableQuery("SELECT  * FROM syn5511444 ")
    demographic = syn.tableQuery('SELECT* FROM syn5511429')
    df=results.asDataFrame(rowIdAndVersionInIndex=True)
    df.set_index(['recordId'], inplace = False)
    demographicdf=demographic.asDataFrame(rowIdAndVersionInIndex=True)
    colList=['healthCode','professional-diagnosis','age','gender']
    demographicdf=demographicdf[colList]
    mergedDf = pd.merge(df, demographicdf, how='inner',on=['healthCode'])
    mergedDf = mergedDf.set_index("audio_audio.m4a", drop = False)
    return mergedDf

def librosaMeasures(y,sr):
    chroma_stft = librosa.feature.chroma_stft(y=y, sr=sr)
    rmse = librosa.feature.rmse(y=y)
    spec_cent = librosa.feature.spectral_centroid(y=y, sr=sr)
    spec_bw = librosa.feature.spectral_bandwidth(y=y, sr=sr)
    rolloff = librosa.feature.spectral_rolloff(y=y, sr=sr)
    zcr = librosa.feature.zero_crossing_rate(y)
    mfcc = librosa.feature.mfcc(y=y, sr=sr)
    return chroma_stft, rmse, spec_cent, spec_bw, rolloff, zcr, mfcc

def pitchHrnMeasures(f0min,f0max,unit):
    pitch = call(sound, "To Pitch", 0.0, f0min, f0max) #create a praat pitch object
    pitchMean = call(pitch, "Get mean", 0, 0, unit) # get mean pitch
    PitchStdev = call(pitch, "Get standard deviation", 0 ,0, unit) # get standard deviation
    harmonicity = call(sound, "To Harmonicity (cc)", 0.01, f0min, 0.1, 1.0)
    hnr = call(harmonicity, "Get mean", 0, 0)
    return pitchMean, PitchStdev, hnr
```

```
df=downloadAll()
for index,row in df.iterrows():
 if (row['audio_audio.m4a'] not in dataCheck.values):
    if row["professional-diagnosis"]:
           p="Parkinson"
    else:
           p="NoParkinson"
    recordId=row['recordId']
    location=dowloadByRecordId(recordId)
    try:
          wave_file = covertToWav(location)
          y, sr = librosa.load(wave_file)

          time=librosa.get_duration(y=y, sr=sr)

          (chroma_stft, rmse, spec_cent ,spec_bw ,rolloff ,zcr,mfcc)=librosaMeasures(y,sr)
          sound = parselmouth.Sound(wave_file)
          print("Processing {}...".format(wave_file))

          pointProcess = call(sound, "To PointProcess (periodic, cc)", f0min, f0max)
          #ff0min, f0max=75,600 default

          (pitchMean, PitchStdev, hnr)=pitchHrnMeasures(f0min,f0max,unit)

          (localJitter ,localabsoluteJitter ,rapJitter, ppq5Jitter, ddpJitter, localShimmer, localdbShimmer,
          apq3Shimmer, aqpq5Shimmer, apq11Shimmer, ddaShimmer)=
          jitterShimmerMeasures(sound,pointProcess)

          (f1_mean,f2_mean,f3_mean,f4_mean)=formantMeasures(sound,pointProcess)

          to_append = f'{p} {row["audio_audio.m4a"]} {row["recordId"]} {row["healthCode"]} {row["age"]}
                         {row["gender"]} {time} {pitchMean} {PitchStdev} {hnr} {np.mean(chroma_stft)}
                         {np.mean(rmse)} {np.mean(spec_cent)} {np.mean(spec_bw)} {np.mean(rolloff)}
                         {np.round(localJitter,6)} {np.round(localabsoluteJitter,6)} {np.round(rapJitter,6)}
                         {np.round(ppq5Jitter,6)}  {np.round(ddpJitter,6)} {np.round(localShimmer,6)}
                         {np.round(localdbShimmer,6)} {np.round(aqpq5Shimmer,6)}
                         {np.round(apq11Shimmer,6)}{np.round(ddaShimmer,6)}
                         {f1_mean} {f2_mean} {f3_mean} {f4_mean}'

          for e in mfcc:
                   to_append += f' {np.mean(e)}'

          file = open('/content/drive/My Drive/dataCheck.csv', 'a', newline='')
          with file:
                    writer = csv.writer(file)
                    writer.writerow(to_append.split())

          os.remove(location)
          os.remove(wave_file)

    except:
          os.remove(location)
          continue
```