Classification of Depressed and Non-depressed Subjects and Predicting Their MADRS Score Using Machine Learning

Thesis Report

FOR THE DEGREE OF

MASTER OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted by:

Shivam Kasat (MIT2019024)

UNDER THE SUPERVISION OF

Dr. Sonali Agarwal

(Associate Professor)

INDIAN INSTITUTE OF INFORMATION TECHNOLGY,

ALLAHABAD (U.P.)

10 July, 2020

Index

Abstract 03

Objective 04

Motivation 05

Thesis Outline 07

Literature Review 08

- Mental Health 08
- Machine Learning 09
- Related Work 10
- Dataset and performance matrices 12

Methods and Algorithms Used 14

- Decision Trees 14
- Random Forest 15
- AdaBoost Ensemble Model 16
- XGBoost Ensemble Model 17
- Gradient Boosting Machines 19
- CatBoost Ensemble Model 20

Associated Challenges & Research Gap 21

Work Plan 22

Work Done 24

Results 29

Conclusion 31

References 32

Abstract

Wearable sensors measuring different parts of people's activity are a common technology nowadays. Data created using these devices holds a lot of potential besides measuring the quantity of daily steps or calories burned, since continuous recordings of heart rate and activity levels usually are collected. Furthermore, there is an increasing awareness in the field of psychiatry on how these activity data relates to various mental health issue such as changes in mood, personality, inability to cope with daily problems or stress and withdrawal from friends and activities. In this thesis work we present the analysis of unique dataset containing sensor data collected from patients suffering from unipolar and bipolar depression. The dataset contains the motor activity data of 23 unipolar and bipolar depressed. We will apply machine learning methods on this imbalance dataset to classify patients as depressed and non-depressed, further we will try to improve the performance of model using some oversampling methods to make the dataset balanced. We will also try to make a multi-class classification model using the same dataset to classify patients as non-depressed, mild-depressed and severe-depressed patients. At last we will create a prediction model for predicting the MADRS score based on the motor activity of patients which further can be used for classification and predicting his status in terms

of depression.

Objective

In this thesis work we will focus our mind and power in these three objectives stated below:

1. The first objective is to classify the subjects are depressed and non-depressed based on the motor-activity data.
2. The second objective is to classify the subjects as suffering from no-depression, mild-depression or severe depression based on the motor activity data.
3. The third objective is to create regression model which can predict the MADRS score of a patient using its motor activity data which later can be used into classification using the model created as the solution to the first two stated problems.

The dataset used in this thesis work is The Depresjon Dataset which consist of data collected from conditions and controls by using actigraph watch which they wore in their right hand throughout the process of data collection. The actigraph watch collects the data of their motor-activity. The sampling frequency is 32Hz and movements over 0.05g are recorded. A corresponding voltage is produced and is stored as an activity count in the memory unit of actigraph watch. The number of counts is proportional to the intensity of movement. Total activity counts were continuously recorded in one-minute intervals.

Motivation

In this time of pandemic everyone is suffering from some sort of mental imbalance due to loss of their jobs, loss of quality of education, change is their lifestyle, staying home for long, not being able to go outside etc. all these changes in normal life of a human being has affected their brain in some way and in most of them the effect of this change is not good or we can say is adverse making them suffer from some level of depression varying from person to person based on change in their lifestyle.

So, this changing human behavior caught my interest and bought light upon this problem of identifying the level of depression in person based on their motor-activity and predicting their MADRS score which tells us the level of depression a person is suffering from.

Nowadays it has become possible to measure different human activities using wearable devices. Besides measuring the number of steps and calories burned, these datasets have much more potential since different activity levels are also collected. Such data would be helpful in the field of psychology because it can relate to the various mental health issues such as changes in mood and stress.

In this thesis work, I will try to present machine learning approach to detect depression using a dataset with motor-activity recordings of one group of people with depression and one group without, i.e., the condition group includes 23 unipolar and bipolar persons and the control group includes 32 persons without depression.

The concept of Artificial Intelligence came into picture when Alan Turing gave his first public lecture in London 1947 to mention computer intelligence saying “What we want is a machine that can learn from experience” and this will only be possible when a machine will be capable of altering its own decision based on its state. Now we came very far from the base given by Alan Turing and we have Robots and Machines taking decisions and performing certain tasks even better than humans like tagging images to a particular category. Even AI has won against world champions in board games.

Machine learning and Deep learning are subsets of Artificial Intelligence, Machine learning learns patterns in data via different algorithms like Logistic Regression, Random forest, SVM etc. by iterating over data several times and learning the parameters which later are used on unseen data. Deep-learning is another subset of AI which brings Neural Networks or Deep Neural Networks into picture. The base algorithm that is used to learn parameters in deep learning is back propagation.

AI and Medical Science can seem to work together to ease the task on identifying and diagnosing different types of diseases. The Area that AI touches in Medical Science is very vast right from Identifying if a person is having a broken bone to predicting the chances of death due to heart attack. We here in this thesis work will try to touch one such sector of medical Industry which is Mental Health.

According to WHO (World Health Organization) more than 264 million people of all ages are suffering from depression, is a leading cause of disability worldwide and is a major contributor to the overall burden of disease. In India as on October 9, 2020 more than 7.5% of the population is suffering from some mental disorder and predictions are such that the stats will rise to 20% by the end of the year. In numbers 56 million of Indians suffer from depression and another 38 million suffer from anxiety disorder.

WHO also states that the mental health workforce is also not up to the mark, The desired numbers are 3 psychiatrists and psychologists per 100,000 population and we have 0.3 psychiatrists, 0.12 nurses, 0.17 psychologists and 0.7 social workers per 100,000 population. Which is nowhere close to the desired numbers.

This motivates us to work on this sector of medical Science and use machine learning to help improve the condition by detecting depression in subjects using the motor activity data collected from actigraph watches which subjects are made to wear throughout the experiment whose data we are going to use to train our machine learning model. We aim to develop three different machine learning models which help us classify the subjects who are depressed on Non-Depressed or will help us determine the level of depression the subject is suffering from namely normal, mild or severe. We also aim to calculate the MADRS score of patients which is a measure of depression.

Thesis Outline

The main Idea behind this thesis work is detection of depression in subject and classifying the subjects into different categories based on the level of depression they are suffering from. Furthermore, we will create a regression model which will predict the MADRS score of the subject from motor activity data which further can be used to classify that subject.

To achieve the required target, we will create three different machine learning models as described below:

- **Binary Classification Model:**
  The first model will be the base model which will classify the subjects into two categories namely Depressed and Non-depressed. The input to the model will be motor activity data and output will the class of the subject as Depressed or Non-Depressed.
- **Multi-Class Classification Model**
  The second model will classify the subjects into three categories based on MADRS score, which are Normal (No-Depression), Mild Depression or Severe Depression. The input to the model will be motor activity data and output will be one of the three class Normal, mild, Severe.
- **MADRS Prediction Model**
  The third and last model will output the MADRS score of the subject from Motor activity data. This predicted MADRS score can further be used to classify the patient using the earlier two model described above. The input to this model will be motor activity data and output will be MADRS score.

To achieve the objectives described above we will experiment with different machine learning algorithms like SVM, logistic regression, Random forests, KNN, decision Trees, Ensemble models etc. We will also experiment with different Deep learning models created using neural networks, Convolutional neural networks, Long Short Term Memory Networks (LSTM), Bi-directional LSTM etc. Finally, we will use the model/Algorithm which gives the best results.

Literature Review

In this thesis, we will be focusing majorly on two fields which are Mental Health and Machine learning/ Deep learning. In this section, we will give a brief knowledge about the two fields. Firstly, we will dig a bit into Mental health, what mental health is and what are different disorders related to mental health then we will shed some light on different machine learning algorithms and deep learning models.

Mental Health

In India and in many other countries around the globe mental health is not taken seriously compared to physical health. Mental health refers to cognitive, behavioral, and emotional well-being. It is all about how people think, feel, and behave. People sometimes use the term "mental health" to mean the absence of a mental disorder. "Without mental health, there can be no true physical health", Dr. Brock Chisholm (first Director-General of the World Health Organization). The most common types of mental illnesses are anxiety disorder, schizophrenia disorders, mood disorders. we will try to understand each of these in brief in the below sections. This improper awareness of mental health should be taken into consideration by the government and WHO.

Anxiety Disorder

Anxiety is the body's natural response to stress. It is a feeling of fear or apprehension that one feels about the situation going on. A normal level of anxiety in daily life is not a problem but it becomes a disorder when anxiety lasts longer than usual like for weeks or months. This disorder is called Anxiety disorder. People having such disorder frequently have intense, excessive, and persistent worry/fear about everyday life situations. Symptoms of anxiety disorder include:

- Feeling nervous and tense
- Having sense of danger and fear
- Sweating
- Less sleep
- Feeling weak and tired

Schizophrenia disorder

Schizophrenia is distortions in thinking, perception, emotions, decision making, behavior, etc. This disorder affects a person's ability to think, feel or behave in certain situations. The exact cause of schizophrenia is unknown but is considered to be a combination of genetics, environment, and altered brain chemistry. Common symptoms of Schizophrenia include:

- Psychosis such as delusions and hallucinations
- Lack of facial expression
- Lack of motivation and low self-esteem
- Lack of emotional expression
- Lack of concentration

Mood disorder

Mood disorder is a frequent distortion of mood that means a person may feel extremely sad, empty, depressed or periodic mood of extreme happiness and sadness (Mania). The most common types of mood disorders include Major Depression, Dysthymia, Bipolar disorder, Substance-induced mood disorder. Some common symptoms of Mood disorder include:

- Ongoing sad, anxious or empty mood
- Excessive guilt
- Having low self esteem and feeling in-adequate or worthless
- Relationship problems like breakup or divorce
- Very sensitive to failure or Rejection

Machine Learning

Machine learning is a branch of artificial intelligence, where the machine learns patterns from the data and tries to make predictions from the experience gained. Here we did not write instruction which computer need to follow exactly but we make use of some algorithms and data. We feed the data to the algorithm and let the machine pass the data through the algorithm several times. By passing the data through an algorithm our machine is trying to learn the different patterns that data posses. This part of the learning pattern from data is called Training the Model. After our model is trained we use the trained model to make predictions on unseen data. In order to see how the model is performing on the unseen data we split the data we have into two/three parts generally and name them as train, dev, and test sets. Generating results on unseen data to check the efficiency of the model is called testing phase. So this is the general flow of Machine learning consisting of training and testing phase. If our testing results are good we deploy the model to our production environment.

Machine learning seems easier just making the model learn through the data using algorithm but it isn't the whole scenario as one must know which algorithm to choose to achieve the best results. Now you might be thinking why not try a bunch of algorithms and choose the one which gives the best result. So simple! Isn't it, But it isn't. One must know a small subset of algorithms to try out ample algorithms available. In case when you have a large dataset (consisting of millions of data sample) trying a large set of algorithms is a foolish choice as it requires a large amount of computation power for algorithms to execute and we pass the data through the algorithm thousands of times (sometimes) to make our model learn better. Thus high computational power is one of the requirements of Machine Learning. Earlier this task of training models was difficult as GPUs were expensive and CPUs consumed a lot of time but now with the growth of technology and competition in the market. GPUs are available at a lower cost, Also some of the companies provide cloud-based GPUs for cheaper rates. This lead to more and more people showing interest in machine learning.

Most of the machine learning algorithms are based on probabilistic and statistics theorems and some of them are result of experience in the field which we are trying to prove with mathematics.

Earlier these machine learning models are coded explicitly using some programming language from scratch and then trained but now we use different frameworks/libraries available in the market to train the model. These frameworks make our task easier, clean and we can now focus on the main part which is training the model instead of coding the algorithm. Examples of such frameworks are sklearn, pytorch, Tensorflow, etc. with these algorithm training a model is just a few lines of code having some function calls and testing is another few lines of code. This framework has made the life of data scientists much easier compared to earlier times.

Deep learning is that part of machine learning which explicitly deals with Neural networks. Generally, we say a neural network fall under deep learning when it has more than one hidden layer. Deeper the network better the learning capacity of the network but it does not mean that you can make the neural network very deep like hundreds of hidden layer and expect it to learn really well because after a certain threshold exceeding the layers in deep learning in idempotent that is they'll not contribute to the learning of model. Frameworks generally used for deep learning include Keras, Tensorflow, pyTorch etc.

One thing that these framework are not capable of is structuring the data and for this task we use different libraries like Pandas and Numpy. They make the data suitable for algorithms as machine learning data only expects numerical data values and do not support strings or any other data type as data values. So we use these libraries to convert string data to numbers using one-hot encoding technique available as functions in frameworks. Numpy also provide computational advantage over python list as mathematical operations on numpy arrays are faster.

Related Work

In this section, we will try to shed some light on work done earlier in the field of Mental health and Monitoring Systems by other researchers.

Mental Health and Monitoring System

Here we will discuss earlier research related to mental health and how machine learning worked in this field of Medical science.

E. Garcia-Ceja et al [1]. did some research and survey study on recent work in machine learning and MHMS by giving different work labels: Study duration (short or long term), sensor types (software/external/wearable/social media) or study type (association/detection/forecasting).

Association studies help us to understand the relationship between different variables and includes methods like linear regression, analysis of variance, t-tests and correlation analysis. Detection studies deals with recognising different state of Mental health using methods like classification models or clustering algorithms. Forecast studies deals with predicting something like MADRS score, epileptic seizures. Different sensors used by researchers include wearable one like smart-watches, smart-phones, external sensors like cameras and microphones continuously monitoring patients activities like sleeping habits, eating habits, hours of sleep, tone of talks, talking hours etc. Some researches used software/social media as sensors like twitter, Instagram to collect data as it has been observed that now a days people share most of their feeling on social media platforms.

Studies related to depression and bipolar disorder. One such study by O'Brien, J.T. et al. [2] was an association study about bipolar disorder and depression. twenty-nine controls (healthy subjects) and thirty conditions (subjects with depression) were under study as subjects of the experiment and the goal was to find the relation between depression and physical activity referred to as late-life depression (LLD) in the paper and they concluded that physical activity of subjects suffering from LLD was lower compared to healthy subjects.

Another study of type detection was done by Grunerbl, A. Et al. [3] where the experiment subjects included ten bipolar patients between the age of 18 and 65 years old. They used phone calls and microphone data of patients and achieved an accuracy of 76% with precision and recall over 97% for recognition of bipolar state detection. Along with microphone data they also used GPS and accelerometer data and achieved recognition accuracy of 70% with accelerometer and 80% with GPS.

The same data was used by Maxhuni, A. [4] Et al but along with microphone and GPS data, they also used data obtained through questionnaire on the participants. They applied various machine learning algorithms and their best average accuracy was 85.57%.

Faurhalt-Jepson, M. Et al. [5] did an association study on 29 bipolar participants regarding actions on their smartphones like daily usage, number of received and sent text messages, number of incoming calls, etc. They found a strong correlation between the recorded information and the mental health of the patients.

Andrew G. Et al. [6] used Instagram photos to study depression and used machine learning on the data. A total of 43,950 photos of 166 participants was used as a dataset. Statistical features extracted from photos using color analysis, metadata, and face detection are used as input to machine learning algorithms and achieved an accuracy of 70\%, using social media platform for depression detection was an interesting study as nowadays social media is used by most of the people to share their feelings.

Mowery, D. Et al [7] did a similar study but used Twitter posts instead of Instagram posts. Their aim was to classify if the Twitter post contains some evidence of depression. Features extracted from tweets included syntax of tweets, usage of emoticons, and sentiment in text. A good accuracy has been achieved in identifying tweets with no evidence of depression but the results were not satisfactory for other cases.

Garcia-ceja et al. [8] also published a paper where he used machine learning algorithms like Random forests, Decision trees, and Dense neural networks to classify subjects as depressed or non-depressed. The dataset of experiments included twenty-three unipolar and bipolar patients and thirty-two healthy controls and achieved an f1-score of 0.73 using random forests and 0.7 using Dense neural networks.

we will be using the same dataset used by Garcia-ceja et al named as depression dataset for our study and will try to balance the dataset to achieve better performance using different machine learning algorithms including decision trees, random forest, support vector machines, CNN, LSTM, etc. For oversampling, we will be using SMOTE oversampling technique.

Dataset and performance metrices

Dataset

The dataset was collected to study motor activity in patients suffering from schizophrenia and major depression. The dataset consists of motor activity data collected using an actigraph watch which patients were made to wear in their right hand. The actigraph watch measures the daily activity level of patients at a frequency of 32Hz and movement over 0.05g is recorded. Whenever the watch senses a moment, the corresponding voltage level is produced and is stored in the memory unit of the actigraph watch. The counts stored in the watch are proportional to the intensity of the movement. Activity count was continuously recorded at an interval of one minute.

The dataset is named as depression dataset and is freely and openly available to everyone. The dataset consists of two folders: one consisting of motor-activity data of each patient suffering from some sort of unipolar or bipolar disorder and the other consisting of motor-activity data of healthy subjects not suffering from any depression disorder. The first folder named "condition" has 23 unipolar and bipolar patients and the other folder named "control" contains the data of 32 healthy controls. For each subject, a different CSV file is provided containing the actigraph data collected over time. In addition, a separate file is provided named as "scores.csv" which contains the MADRS score of each subject along with the following columns:

- number : A unique ID for each subject
- days : number of days of data collection
- gender : 1 = female, 2 = male
- age : age of the participant
- afftype : affliction type, 1-Bipolar II, 2-Unipolar depressive, 3-Bipolar I
- inpatient : whether patient is inpatient (1) or outpatient(2)
- edu : years of education completed
- marriage : 1-married/cohabiting, 2-single
- work : 1-work/study, 2-unemployed/sick leave/pension
- madrs1 : MADRS score before activity measurement started
- madrs2 : MADRS score after activity measurement ended

```
scores_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 55 entries, 0 to 54
Data columns (total 12 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   number     55 non-null     object
 1   days       55 non-null     int64
 2   gender     55 non-null     int64
 3   age        55 non-null     object
 4   afftype    23 non-null     float64
 5   melanch    20 non-null     float64
 6   inpatient  23 non-null     float64
 7   edu        53 non-null     object
 8   marriage   23 non-null     float64
 9   work       23 non-null     float64
 10  madrs1     23 non-null     float64
 11  madrs2     23 non-null     float64
dtypes: float64(7), int64(2), object(3)
memory usage: 5.3+ KB
```

Fig 1: Figure showing the information of scores.csv explaining differnt parameters and number of non-null values.

	number	days	gender	age	...	marriage	work	madrs1	madrs2
0	condition_1	11	2	35-39	...	1.0	2.0	19.0	19.0
1	condition_2	18	2	40-44	...	2.0	2.0	24.0	11.0
2	condition_3	13	1	45-49	...	2.0	2.0	24.0	25.0
3	condition_4	13	2	25-29	...	1.0	1.0	20.0	16.0
4	condition_5	13	2	50-54	...	2.0	2.0	26.0	26.0

[5 rows x 12 columns]

	number	days	gender	age	...	marriage	work	madrs1	madrs2
50	control_28	16	2	45-49	...	NaN	NaN	NaN	NaN
51	control_29	13	2	50-54	...	NaN	NaN	NaN	NaN
52	control_30	9	2	35-39	...	NaN	NaN	NaN	NaN
53	control_31	13	1	20-24	...	NaN	NaN	NaN	NaN
54	control_32	14	2	25-29	...	NaN	NaN	NaN	NaN

[5 rows x 12 columns]

Fig 2: Figure showing the first five and last five rows of scores.csv. The first five rows shows data of condition group and last five rows shows the data of control group.

Methods and Algorithms Used

Decision Tree

Decision Tree is one of the widely used tools used for both classification and regression tasks in supervised machine learning. In general decision trees are constructed via different algorithmic approaches based on different ways to split the data on different attributes. A decision tree basically consists of the root node, internal nodes, leaf nodes, and edges. Root Node and internal nodes represent the decision made based on different attributes chosen at a time to split the tree further, while the leaf node represents the target label and edges representing the answer to the question asked by the parent node to make the decision.

At each internal node the choice of attribute is done based on loss or gain, we recursively try different attributes and calculate the loss or gain score. For that node we choose the attribute to make the decision which has the least loss or highest gain, this way we make a single split at one node. Further we make the split at child node using the same approach if the loss made at the child node split is higher then the loss at parent node then we don't make the split further and make that node a leaf node as there is no point in further splitting the node.

Let us take the example of famous titanic dataset to explain the decision tree. From the titanic dataset we will choose three features namely sex, age, and sibsp (number of spouses or children along). Using these features we will try to predict if the passenger survives or not.

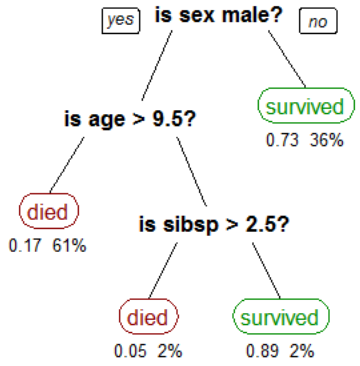

Fig 3: Figure explaining decision tree

One of the Loss function that we use to choose the attribute at node is Gini Impurity.

$$I_G = 1 - \sum_{j=1}^{c} P_j^2$$

P_j is the portion of the samples that belong to the class c for a particular node.

Random Forest

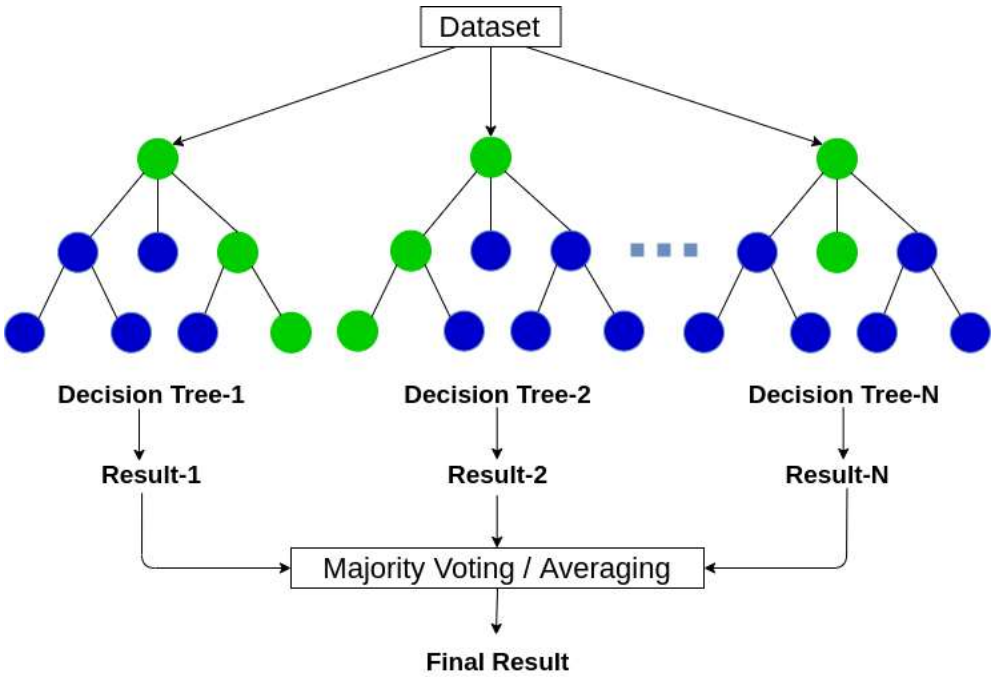

Fig 4: Figure showing decision trees constructing random forest

Random forest is a bagging Tree model i.e, it bootstraps the data and the result is the aggregation of all the models we made using bootstrapped data. Random forests are created using the following steps:

- We create a bootstrap sample from the dataset, The bootstrap sample may contain duplicate samples.
- From the bootstrapped sample we use random attributes (maybe all) to create the decision tree.
- We repeat the step 1 and 2 several numbers of times (maybe hundreds or thousands or larger).
- The target labels for unseen data are obtained by aggregation of the result from all the decision trees created during the process.
- We measure the accuracy of this random forest by out of the bag samples (samples that never became part of bootstrap data).
- We try a different number of random attributes for creating different random forest and choose the one with the highest accuracy.

Random forest is considered better over Decision tree because different decision trees which are part of the random forest are uncorrelated and a larger number of such uncorrelated models working as a single model in association tend to outperform one single individual constituent which is a part of random forest.

AdaBoost Ensemble Model

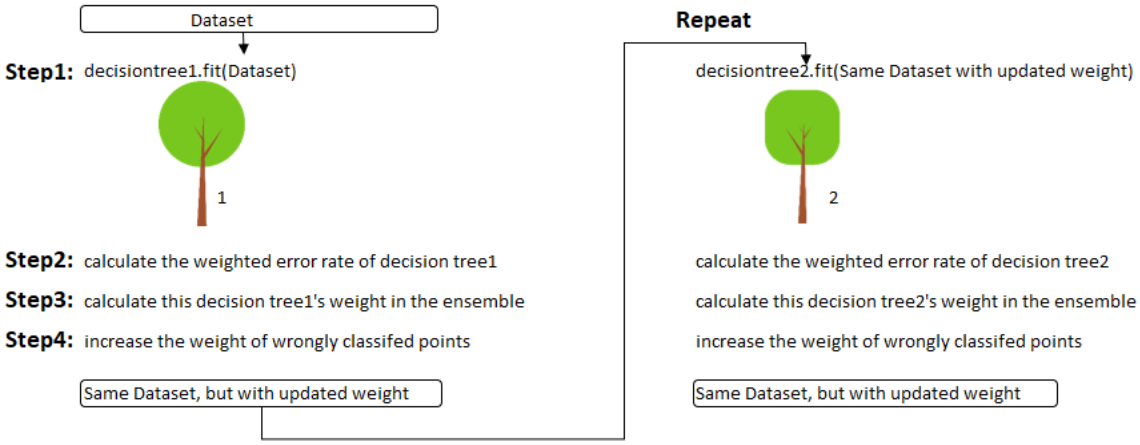

Fig 5: Figure showing decision making in AdaBoost Ensemble Model

AdaBoost is a boosting model, The main idea behind boosting model is we create a new model by improving over the mistakes of the previous model.

The Three main ideas behind boosting model are:

1. AdaBoost combines a lot of "weak learners" to make classifications. The weak learners are always almost stumps. Stumps are small size decision three that is just one node having two children leaf nodes.
2. Some stumps get more say in the classification than others.
3. Each stump is made by taking mistakes of the previous stump into account.

The steps involved in creating forest of stumps are:

1. We give each sample a sample weight that indicates how important it is to be correctly classified. Initially each sample is given equal sample weight.
2. Now we will try to choose the correct root for the stump by finding which feature classified the data with least error.
3. After finding the stump with least error we will find out the importance of that stump as a whole. We calculate the importance of a stump based on its accuracy to classify the samples.

$$importance\_of\_stump = \frac{1}{2} * log\ log\ \frac{1 - total - error}{total\_error}$$

4. Now we will increase the weight of samples of incorrectly classified samples and decrease the weights of correctly classified samples.

5. increase the weight of incorrectly classified samples to

$$new\_weight = old\_weight * exp^{importance\_of\_stump}$$

6. decrease the weight of correctly classified samples to

$$new\_weight = old\_weight * exp^{importance\_of\_stump}$$

7. Normalize the new sample weights
8. Now we will use the new normalized sample weights to generate next stump and the sample set in such a way that we can give more emphasis to the incorrectly classified samples.
9. Repeat the steps from 1 to 8 to get stumps that can classify better.
10. We will use these to classify new samples by combining the results of all stumps and adding the importance of stumps. The new sample will be classified into the class for which the total importance is highest.

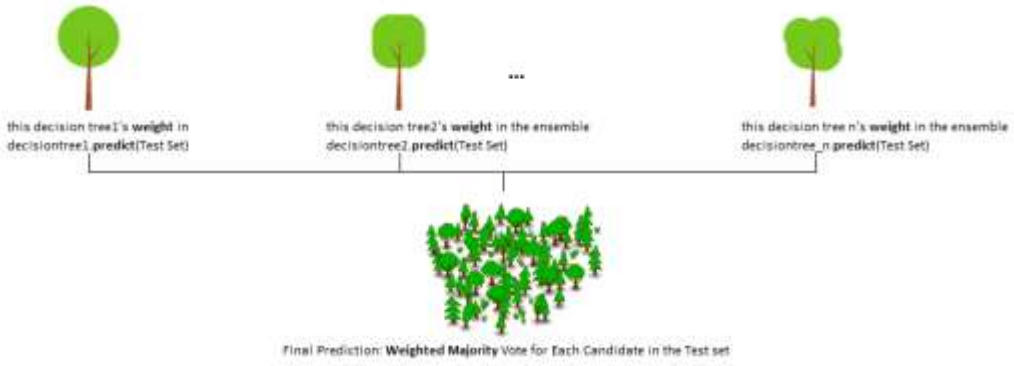

Fig 6: Figure showing forest of stumps of AdaBoost

XGBoost Ensemble Model

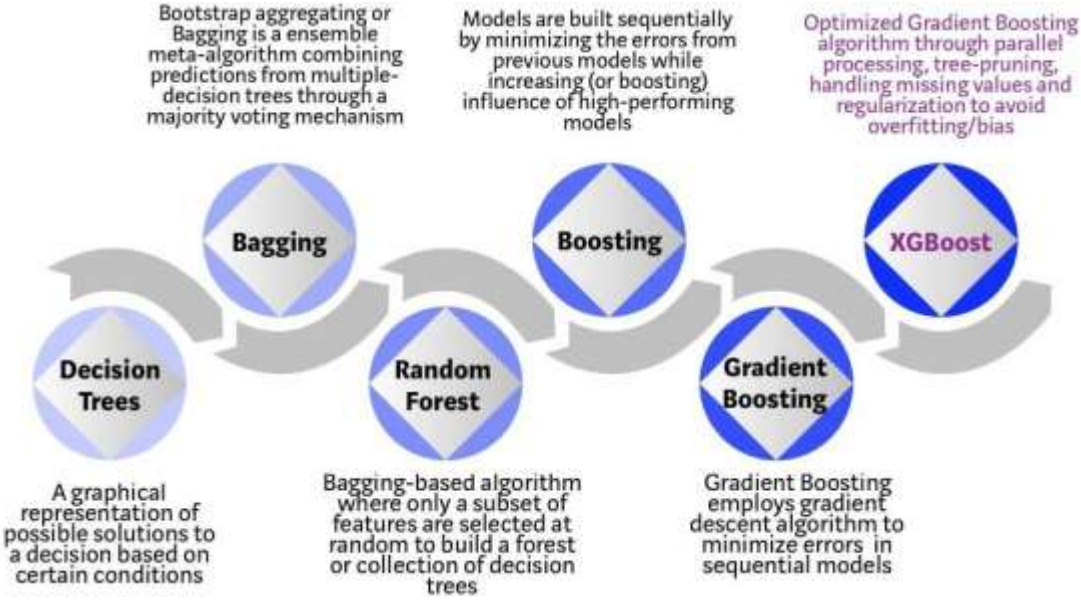

Fig 7: Figure showing evolution of decision tree to XGBoost ensemble model

XGBoost stands for eXtreme Gradient Boosting. A boosting algorithm where the errors of the previous model are reduced using gradient boosting Algorithm. XGBoost improves over GBM by system optimization and algorithmic enhancements. XGBoost was developed by Tianqi Chen and Carlos Guestrin as a research project at the university of Washington.

Reasons why XGBoost Performs Well

- **Parallelization:** The sequential tree building approach in XGBoost is supported by parallelization. This was made possible as the loops responsible for creating base learners can be interchanged. Earlier parallelization was not possible due to this inability to interchange the loops that is, without completing the inner loop responsible for leaf nodes in tree we cannot start the outer loop. In XGBoost to improve the performance and achieve parallelization these loops are made interchangeable through global scan of all instances and sorting using parallel threads.
- **Tree Pruning:** In Grading Boosting Model the stopping criteria for tree splitting was greedy in nature. XGBoost uses max depth criteria for tree pruning starting from leaf nodes and based on pruning factor gamma and gains calculated using similarity score.
- **Hardware optimization:** Hardware optimization is achieved using caching techniques by allocating internal buffers to each thread to store gradient statistics. For disk space optimisation it uses out of core computing while dealing with big dataframe that do not fit in memory.
- **Regularization:** It uses both l1 (lasso) and l2 (Ridge) regularisation to prevent over-fitting.
- **Sparsity Awareness:** XGBoost handles different type of sparsity patterns in the data more efficiently by automatically learning best missing values depending on training loss.
- **Weighted Quantile Sketch:** XGBoost uses weighted quantile sketch algorithm to find the most optimal splitting points among weighted datasets.
- **Cross-validation:** There is no need of perform cross validation externally, It comes with inbuilt cross validation techniques thus making it easy to perform cross validation if required.

Fig 8: Figure explaining reasons why XGBoost performs well.

Gradient Boosting Machines

The main idea behind GBM is to take weak learners to combine them to form strong learners and the basis behind choosing weak learners is achieved through minimising the loss function. The loss function help us choose which tree should be included in GBM.

Boosting is a process of combining weak learners to form strong learners. Weak learners can be different machine learning model. In case of GBM weak learners are decision Trees.The decision trees that are used in GBM are called stumps. Stumps are decision tree of size upto 2 level. The main idea is how do we improve the performance of these weak learners. Different ensemble models have different methods to improve the performance of the model.

GBM makes use of Loss function to optimise the model, the choice of the loss function depends on the problem we are trying to solve. In case of regression problem, the loss function can be something like squared loss or Root mean squared loss and in case of classification problem it can be logistic loss. The optimisation take place by minimising these loss function using gradient descent algorithm which is the heart of machine learning. Every iteration we try to parameterise the weak learners such that overall cost/loss can be reduced. Thus leading toward a better model.

In GBM trees are constructed in a greedy approach. The choice of splits is done using gini scores. We try to add a tree to the model such that we improve the overall performance of the model. This approach is greedy because we try to make all possible choice to come to the best possible outcome.

We improvise over basic boosting machine in the following ways:

1. **Imposing Tree constraints** - We need to make sure that individual sub-model or weak learners do not take much time to make the decision i.e, they should be simple but effective. To achieve this we impose constraints over weak learners (stumps) by fixing the size (levels) in decision tree, Number of nodes or leaves, Number of stumps in model etc.
2. **Stochastic Gradient Descent -** Instead of just using the naive gradient descent we can use stochastic gradient descent or mini-batch gradient descent, to improve the performance of the model.
3. **Penelized Gradient Boosting** - In case we see our model to be overfitting we can also use penalising through regularisation techniques, Both regularisation can be used here l1 (Lasso) regularisation and L2 (Ridge) regularization.

Catboost Ensemble Model

Catboost is another gradient boosting algorithm introduced in the field by yandex, It came into light when it outperformed older in the segment gradient boosting algorithms like xgboost, adaboost etc.

Features of Catboost are as follows:

- **Better performance on categorical dataset:** From the benchmark scores one can see that catboost tends perform better than all the earlier gradient boosting algorithm but significant

difference is observed when dataset has categorical data. For example on Amazon dataset available openly catboost out-performed LightGBM by 18.79%.

- **Faster Predictions:** Catboost is light weight and faster by 15 times compared to other gradient boosted decision trees, that gives catboost a heads up compared to other algorithms.
- **Pre-tuned hyperparameters:** The default hyper parameter setting in catboost is better compared to other Tree based boosting models, which makes it easy to use for beginners and to achieve baseline results with no hyper tuning.
- **Categorical Feature handling:** The main feature which brought most of the difference in benchmarks was due to this internal categorical feature handling done by catboost algorithm.

  Catboost internally uses target statics which encodes each categorical feature with the estimate of expected target y conditioned by the category.

  To fight prediction shift catboost uses more reliable ordering principle in which algorithm gets training examples sequentially in time. In base settings the value of target shift for each example depends only on the observed history. To achieve better results catboost uses different permutations of samples for different steps of gradient boosting.
- **Ordered boosting/Tree construction:** There are two ways by which catboost choose the tree structure.
  1. Plain Mode: Plain mode uses a combination of different decision tree boosting algorithm techniques to construct its tree through ordered target statistics.
  2. Ordered Mode: In Ordered mode it make different permutations of training examples and maintain different supporting model with each different permutation such that M_i model will be constructed with permutation of first i training samples.

Associated Challenges & Research gap

- **Imbalanced Dataset**
  The dataset is imbalanced i.e., number of samples of depressed patients are 23 which contains both unipolar and bipolar depressed patients while the number of non-depressed controls are 32.
- **Dataset Size**
  Overall size of the dataset is very small to perform deep neural network methods as in total we have 55 data samples having 23 samples of depressed class and 32 samples of non-depressed class.
- **Improvisation in results/performance of model**
  As we have seen research in this field is already conducted and some results are already generated by different classification and prediction models, it will be a challenging task to find a way to further improve the performance of the model.
- **Multi-class Classification**
  Dataset by default contains two classes, one labeled as depressed and other as non-depressed. We on the other hand trying to perform multi-class classification of the dataset based on their MADRS score and improve the performance of classification model.

Work Plan

While going through most of the literature related to this thesis work, I came across various methods that can be used for classification and various ways with which I can balance the datasets but non-of the research paper I went through tried all the combinations possible, The only combinations which I came across were random forest with SMOTE oversampling and deep neural network with random sampling. Still there are different possibilities available that might increase the performance of the model.

Classifications Methods I can use are Linear SVM, RBF SVM, Ensemble model, Logistic Regression, One dimensional CNN, Markov model etc.

Oversampling methods which would be suitable for this task are Random oversampling, SMOTE oversampling, GAN etc.

So, we are planning to create three different models corresponding to each task:

1. For binary classification i.e., Classification of depressed and non-depressed patients.
2. For multi-class classification i.e., Classification of patients based on their MADRS score in three classes. No – depression (0 – 10), Mild depression (11 – 19), Severe depression (> 20)
3. For prediction of MADRS score based on the motor activity of the subject which further can be used for classification task.

1. Binary Classification.

a. Over-sampling:

Since the dataset is imbalanced, we first need to balance it using some oversampling methods which best suits the classification algorithm used. There are around 84 different oversampling methods available but we will choose and try only those which are most suitable for the task of classification some of them includes SMOTE, GANs, Random oversampling etc.

b. Classification:

Now after performing one of the oversampling we have data in its balanced form, now we can perform different classification methods and compare their performance along with the oversampling method used. We have various choices available for creating our classification model such as random forest, ensemble model, linear SVM, RBF-SVM, logistic regression, one dimensional CNN etc. At the end we will choose the one which gives us best results.

2. Multi-class classification

a. Over-sampling :

Again, as the data is imbalanced and after going through different literature, I observed that no depression and severe depression classes approximately have comparable data points but the class with mild depression has comparatively lesser number of samples. So first we will use oversampling to make the data samples comparable for better results and model performance.

b. Classification

For multiclass classification the classification task will be based on MADRS score of the conditions and controls, if the MADRS score lies between 0 -10 then we will classify him as no-depression and if the score lies between 11 – 19 we will classify him as mild-depression and if the score is greater than or equal to 20 we will label him as severe-depressed. For this classification we can use linear SVM, RBF-SVM, random forest, ensemble model, neural network, one dimensional CNN

3. MADRS score prediction

a. Oversampling

Balancing the data will give us better prediction of MADRS score, so we will first make the data balanced using oversampling method that we used in earlier models or some other one which best suits our requirement.

b. Prediction

After the balancing of data, we will use different methods to predict MADRS score of the subject which later can be used to classify him in one of the classes described earlier in multi-class classification. Methods we will try include linear regression, Linear SVM, Random forest, Neural Networks, One dimensional CNN etc.

Work Done

Two out of three model are complete i.e, we are almost done with binary classification and multiclass classification models.

Instead of trying the same model that were used earlier on this model I tried the machine learning models / Algorithms that were not used on this dataset expecting to get better results.

1. Binary Classification

a. Dataset preparation

The dataset available has motor activity data measured using an actigraph watch and so is a time series data. We can feed the timeseries data directly to any machine learning algorithm. So, in order to make it suitable for any machine learning algorithm I divided the time series data into segments each of length 1440 mins (1 Day) and attached a corresponding label with it i.e., 1 if the patient is suffering from depression and 0 if is not suffering from depression.

After performing this data-preparation step we got 1029 data samples each corresponding to a motor activity pattern either of a depressed subject or a non-depressed subject.

b. Modeling the data

Data modeling is done step by step with different model. So that we can enhance the results of the previous step.

Initially We proceed with baseline model without any oversampling method and used the following ensemble models on the dataset

- AdaBoost
- XGBoost
- CatBoost
- GBM

Among this baseline model we got the best accuracy from CatBoost which is of 77.18% and best f1 score of 62.99 from Catboost ensemble model

After creating this baseline model, we tried to SMOTE oversampling as the dataset was imbalanced due to less number of condition samples and more number of control samples. After performing oversampling on training data, we got 1100 data samples for training. Prior we had just 823 data samples for training the model. Again, we used the four-ensemble model on oversampled data. The models were

- AdaBoost
- XGBoost
- CatBoost
- GBM

Out of these four models we got the best accuracy of 71.35 from catboost and f1 score of 61.43% from catboost.

Now instead of doing oversampling just on the training data we thought of performing oversampling on whole data because earlier might a case can be that during train-test split and uneven distribution of control and condition group can occur. So to resolve this issue we performed oversampling on whole data. Prior to oversampling we had 1029 total samples, after performing oversampling we got 1340 samples in total balancing the condition and control group. Now we performed these ensembles model on the data

- AdaBoost
- XGBoost
- CatBoost
- GBM

Out of these four models we got the best accuracy from catboost which is of 83.95% and f1 score of 82.86%.

Since we have a small dataset, we thought of performing k fold cross-validation with k = 5 after performing SMOTE oversampling on whole dataset, expecting to get better and stable results. Doing so we go the best accuracy of 82.91% and f1 score of 82.91% with catboost model.

Using 1D convolutional Neural Network with architecture as follows:

- 1d Convolution Layer with filters = 100, kernel size =10, activation = relu
- 1d Convolution Layer with filters = 100, kernel size = 10, activation = relu
- Max pooling layer with pool size = 2
- 1d Convolution Layer with filters = 160, kernel size = 10, activation = relu
- 1d Convolution Layer with filters = 160, kernel size = 10, activation = relu
- Global Average Pooling layer
- Dropout layer rate = 0.5
- Dense layer
- Optimizer = Adam
- Loss = binary cross entropy

We got a mean accuracy of 63.25% and f1 score of 60.20%.

2. Multiclass Classification

a. Dataset preparation

The dataset consists of motor-activity data measured using actigraph watch and thus dataset for each individual is a times series data and along with time series-based motor activity data we also have a scores.csv file containing MADRS scores of each conditioned subject there are two such columns of MADRS scores named as madrs1 and madrs2. We know that time series data cannot be fed to machine learning or deep learning models directly. So, we preprocessed that data and created segments of data each of length 1440 mins (1 day) and labelled each segment as 0 (no-depression), 1 (mild depression) or 2 (severe depression) based on mean MADRS value obtained using both madrs1 and madrs2

- Mean madrs >= 0.0 and mean madrs <= 6.0 => 0 (no-depression)
- Mean madrs > 6.0 and mean madrs <= 19.0 => 1 (mild-depression)
- Mean madrs > 19.0 and mean madres <= 34.0 => 2 (severe depression)

We removed some features which didn't seemed to contribute to model's performance. After performing this task, we got 1029 samples in total out of which each label has following samples

- Label 0 (no-depression) – 670 samples
- Label 1 (mild-depression) – 168 samples
- Label 2 (severe-depression) – 191 samples

Now we are clear that dataset in imbalanced and using this imbalanced data as it is will not give good results. So, prior to feeding this data to any model or algorithm we thought of balancing the dataset using oversampling techniques. So, in order to perform the oversampling task, we choose SMOTE oversampling and after oversampling the data set we got samples of each label as follows:

- Label 0 (no-depression) – 670 samples
- Label 1 (mild-depression) – 670 samples
- Label 2 (severe-depression) – 670 samples

Now our data is ready to fed to machine learning models.

b. Modeling data

To model the data, we tried different deep learning architectures constructed using CNN layers, dense layer networks, LSTM, Bidirectional LSTM, etc.

First, we used data without oversampling to know how good the model performs on raw data having 1029 samples is described below:

- Conv1D(filters = 64, kernel_size = 3, activation = relu)
- Conv1D(filters = 64, kernel_size = 3, activation = relu)
- MaxPooling1D(pool_size=2)
- Flatten()
- Dense(units = 128, activation = relu)
- Dense (units = 3, activation = softmax)

Loss used was categorial crossentropy loss and optimizer used was ADAM and 10 fold cross-validation is used. The results achieved were as follows:

- Accuracy = 83.58%
- Recall = 83.99%
- Precision = 84.56%
- F1_score = 84.27%

We tried the same model with data oversampled using SMOTE we got 2010 samples in total and got better results but we further decided to twitch the model a little bit so as to improve the results and the model that gave the best results was explained below:

- Conv1D(filters = 64, kernel_size = 3, activation = relu)
- Conv1D(filters = 64, kernel_size = 3, activation = relu)
- MaxPooling1D(pool_size = 2)
- LSTM(units = 64)
- LSTM(units = 64)
- Flatten
- Dense(units = 128, activation = relu)
- Dense(units = 128, activation = relu)
- Dense(units = 3, activation = sigmoid)

Loss used was categorial crossentropy loss with ADAM optimizer and 10 fold cross validation. The results obtained were as follows:

- Accuracy = 97.71
- Recall = 97.72
- Precision = 97.72
- F1score = 97.72

We also used the ensemble model we used in binary classification like CatBoost, AdaBoost, XGBoost and GBM and got the best performance using CatBoost and the results were as follows:

- Accuracy = 96.76
- Recall = 96.76
- Precision = 96.76
- F1 score = 96.76

3. MADRS Prediction Model

a. Dataset Preparation

The dataset consists of motor-activity data measured using actigraph watch and thus dataset for each individual is a times series data and along with time series-based motor activity data we also have a scores.csv file containing MADRS scores of each conditioned subject there are two such columns of MADRS scores named as madrs1 and madrs2. We know that time series data cannot be fed to machine learning or deep learning models directly. So, we preprocessed that data and created segments of data each of length 1440 mins (1 day) and labelled each data segment with average madrs score value of madrs1 and madrs2. After creating such segment, we got in total 1029 data samples with 1440 features each corresponding to a motor activity value at each point in time. These data points are now ready to be fed to our models.

b. Modeling data

Since this task of MADRS prediction is a regression type task so we tried two different approaches. In first approach we tried to model the data using deep learning model. The deep learning model was constructed using CNN layers, dense layer networks, LSTM, Bidirectional LSTM, etc. The architecture of our model is described below:

- LSTM (filters = 64, activation = relu)
- LSTM (filters = 64, activation = relu)

- MaxPooling1D(pool_size = 2)
- Flatten
- Dense(units = 128, activation = relu)
- Dense(units = 1, activation = linear)

The optimizer used was ADAM optimizer with loss as mean squared loss. The results that we got using this deep learning model gave a mean squared error of 9.56.

Since we know that the dataset size is not satisfactory for deep learning model. So we moved on to second approach which is using decision tree based ensemble models. To create the required regression model we work with the following ensembles:

- AdaBoost Regressor
- XGBoost Regressor

And as expected XGBoost gave slightly better results compared to the earlier deep learning model and AdaBoost Regressor which is mean squared error of 7.01. while the AdaBoost Regressor gave mean squared error of 8.80. The best mean absolute error we got was in case of XGBoost Regressor which is 1.31 with standard deviation of 0.236.

Results

1. Without Oversampling (binary classification)

Ensemble Model	Accuracy (%)	F1 Score (%)	Precision Score (%)	Recall Score (%)
AdaBoost	71.84	56.06	56.06	56.06
XGBoost	74.45	58.73	61.66	56.06
CatBoost	77.18	62.99	65.57	60.60
GBM	71.84	52.45	57.14	48.48

2. With Oversampling on Train set (binary classification)

Ensemble Model	Accuracy (%)	F1 Score (%)	Precision Score (%)	Recall Score (%)
AdaBoost	70.59	70.33	70.46	71.18
XGBoost	72.96	72.57	72.15	73.00
CatBoost	78.35	78.13	78.14	78.65
GBM	76.35	76.43	67.14	76.65

3. With oversampling on whole dataset (binary classification)

Ensemble Model	Accuracy (%)	F1 Score (%)	Precision Score (%)	Recall Score (%)
AdaBoost	76.11	73.77	75.63	72.00
XGBoost	82.46	81.42	80.46	82.39
CatBoost	83.95	82.86	82.53	83.20
GBM	80.97	79.18	80.83	77.60

4. **With oversampling and cross-validation with 5 folds (binary classification)**

Ensemble Model	Accuracy (%)	F1 Score (%)	Precision Score (%)	Recall Score (%)
AdaBoost	76.71	76.44	76.34	76.86
XGBoost	78.13	78.08	77.58	78.95
CatBoost	82.91	82.91	81.83	84.32
GBM	78.88	78.99	78.03	80.29

5. **1D convolutional neural network (binary classification)**

Accuracy (%)	F1 Score (%)	Precision Score (%)	Recall Score (%)
63.25	65.00	66.00	60.00

6. **Without oversampling (Multiclass classification)**

Model	Accuracy (%)	F1 Score (%)	Precision Score (%)	Recall Score (%)
AdaBoost	82.34	82.54	82.64	82.34
XGBoost	86.26	86.34	86.89	86.14
CatBoost	85.15	85.24	85.32	85.36
XGBoost	84.32	84.12	85.24	84.26
Deep Learning Model	83.58	84.27	84.56	83.99

7. **With oversampling on Whole data(Multiclass classification)**

Model	Accuracy (%)	F1 Score (%)	Precision Score (%)	Recall Score (%)
CatBoost	96.76	96.76	97.76	96.76
Deep Learning Model	83.58	84.27	84.56	83.99

8. **Without oversampling (MADRS Prediction)**

Model	MAE	MSE
AdaBoost	1.68	8.60
XGBoost	1.32	7.01
Deep Learning Model	2.31	9.56

9. **With oversampling on Train Set (Multiclass Classification)**

Ensemble Model	Accuracy (%)	F1 Score (%)	Precision Score (%)	Recall Score (%)
AdaBoost	89.59	89.33	89.46	89.18
XGBoost	87.96	87.57	87.15	87.00
CatBoost	89.35	89.13	89.14	89.65
GBM	88.35	88.43	88.14	88.65
Deep Learning Model	83.42	84.54	83.46	84.65

Conclusion

Depression is a severe mental disorder with characteristic symptoms like sadness, the feeling of emptiness, anxiety as well as general loss of interest in initiative and activities.

Actigraph recordings of motor activity can be considered as an objective method for observing depression. We can use the data collected from actigraph watch of subjects to classify them as depressed and non-depressed using machine learning classification models.

MADRS score can be used to make this classification more lucid by dividing the subject under three labels which are No-depression, Mild-depression, severe-depression. So that we can predict the level of depression in subject more precisely.

Given the motor activity data of the subjects we can use this data to predict MADRS score which further can be used in our previously stated classification models.

References

1. Enrique G Ceja, Michael Riegler, Tine Nordgreen, Petter Jakobsen, Ketil J Oede-gaard, and Jim Tørresen. 2018. Mental health monitoring with multimodal sensingand machine learning: A survey.Pervasive and Mobile Computing(2018), 1–26.
2. Maria F Jepsen, Maj Vinberg, Mads Frost, Sune Debel, Ellen M Christensen,Jakob E Bardram, and Lars Vedel Kessing. 2016. Behavioral activities collected through smartphones and the association with illness activity in bipolar disorder. International journal of methods in psychiatric research(2016), 309–323.
3. Enrique G Ceja, Michael Riegler, Petter Jakobsen, Jim Torresen, Tine Nordgreen,Ketil J Oedegaard, and Ole Bernt Fasmer. 2018. Motor activity based classifi-cation of depression in unipolar and bipolar patients. InProceedings of the 31stInternational Symposium on Computer-Based Medical Systems. IEEE, 316–321
4. Enrique G Ceja, Michael Riegler, Tine Nordgreen, Petter Jakobsen, Ketil J Oede-gaard, and Jim Tørresen. 2018. Mental health monitoring with multimodal sensingand machine learning: A survey.Pervasive and Mobile Computing(2018), 1–26.
5. Joakim Ihle Frogner, Farzan Majeed Noori, Pal Halvoren, Steven Alexander Hicks, Enrique Garcia-Ceja, Jim Torresen, Michel Alexander Riegler. 2019. One-dimensional convolutional neural networks on motor activity measurements in detection of depression (2019): 4th workshop on multimedia for personal health and health care.
6. RM Hirschfeld. 2014. Differential diagnosis of bipolar disorder and major depressive disorder. Journal of affective disorders169 (2014), S12–S16.
7. Nathalie Japkowicz and Shaju Stephen. 2002. The class imbalance problem: Asystematic study.Intelligent data analysis6, 5 (2002), 429–449
8. N Keshan, PV Parimi, and Isabelle Bichindaritz. 2015. Machine learning for stress detection from ECG signals in automobile drivers. InBig Data (Big Data), 2015IEEE International Conference on. IEEE, 2661–2669.
9. Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas, and others. 2006.Handling imbalanced datasets: A review.GESTS International Transactions on Computer Science and Engineering 30, 1 (2006), 25–36.
10. Eva M Marco, Elena Velarde, Ricardo Llorente, and Giovanni Laviola. 2016.Disrupted circadian rhythm as a common player in developmental models of neuropsychiatric disorders. In Neurotox in Modeling of Brain Disorders — Life-long Outcomes in Behavioral Teratology. Springer, 155–181.
11. O'Brien, J., Gallagher, P., Stow, D., Hammerla, N., Ploetz, T., Firbank, M., Ladha,C., Ladha, K., Jackson, D., McNaney, R., et al.: A study of wrist-worn activitymeasurement as a potential real-world biomarker for late-life depression. Psycho-logical medicine47(1), 93–102 (2017)
12. Penedo, F., Dahn, J.R.: Exercise and well-being: a review of mental and physicalhealth benefits associated with physical activity. Current Opinion in Psychiatry18, 189–193 (2005)
13. Reece, A.G., Danforth, C.M.: Instagram photos reveal predictive markers of de-pression. EPJ Data Science6, 1–12 (2017)
14. Twenge, J.M.: Time period and birth cohort differences in depressive symptomsin the us, 1982–2013. Social Indicators Research121(2), 437–454 (2015)