Identifying diagnostic features of Parkinson's disease using classical machine learning

One Semester Individual Project
Final report – 40 credits

Author: Neofytos Neokleous
Supervisor: Dr Matthias Treder
Module Number: CM3203
May 2020

Abstract

In this research project clinical population's structural preprocessed tabular data extracted form MRI brain images partitioned into ROIs (Regions of Interest) are analyzed using supervised classical machine learning algorithms. Clinical population includes patients and controls from 3 different neuropsychiatric disorders including Parkinson's disease (PD), Post-traumatic stress disorder (PTSD) and schizophrenia (SC). Eight different machine learning algorithms were tested in the study. The primary aim was to achieve a comprehensive comparison between them and hence identify which algorithms can be used in distinguishing the different classes that are present in the given datasets. The secondary aim of this study was to compare 4 different feature selection methods in order to identify which are the diagnostic features of PD. Two main classification approaches were followed in the project. The first approach involved classifying the patients into the 3 different diseases. The second approach involved classifying the patients from the controls. The collective results from those two approaches concluded that multilayer perceptron, logistic regression and random forest classifiers were the ones that had the best overall performance. The brain areas (diagnostic features) that appear to be correlated with the occurrence of PD were found from the common features identified by the two classification approaches. The features that were identified as the diagnostic ones for PD are the: 3rd ventricle, left putamen, left amygdala, right ventral DC, pontine crossing tract, body and genu of corpus callosum, right anterior corona radiata, superior middle and inferior cerebellar peduncle.

Acknowledgments

Firstly, I would like to thank my supervisor Dr Matthias Treder who guided me through this educational journey of completing my firs data science project. Our weekly meetings through the semester and lectures prior to the start of the project gave me a solid background in machine learning and hence enabled me to complete this research project to the best of my abilities.

Finally, I would like to thank Dr Maria Neokleous, whose help was valuable regarding the scientific understanding of the brain structures and allowed me to further establish the relationships between the diagnostic features of PD.

Table of Contents

Table of Figures

Table of Abbreviations

Abbreviation	Explanation
MRI	Magnetic Resonance Imaging
DTI	Diffusion Tensor Imaging
PD	Parkinson's Disease
PTSD	Post-Traumatic Stress Disorder
ROIs	Regions Of Interest
CVD	Cardiovascular Disease
LDA	Linear Discriminant Analysis
KNN	K-Nearest Neighbors
PCA	Principal Component Analysis
BNB	Bernoulli Naïve Bayes Classifier
SVM	Support Vector Machine
SGD	Stochastic Gradient Decent
MLP	Multilayer Perceptron
ANN	Artificial Neural Networks
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
CV	Cross Validation
ROC	Receiver Operating Characteristics
AUC	Area Under the Curve
RFE	Recursive Feature Elimination
LASSO	Least Absolute Shrinkage and Selection Operator
MAE	Mean Absolute Error
LG	Logistic regression
T-SNE	t-Distributed Stochastic Neighbour Embedding

Introduction

Living in a world where people's suffering increases from diseases that are related to the dysfunction of the brain, encourages scientists to seek for the causes and possible cures. Despite years of research, the scientific society is still unable to explain with accuracy how the human brain functions and therefore allow the early detection and cure of various diseases. The 86 billion neurons [1] working simultaneously in the brain contribute in making it almost impossible to decrypt the secrets of the brain using traditional approaches.

My research aims to identify the brain areas that are mostly correlated with the occurrence of Parkinson's disease (PD). Unfortunately, research shows that the number of Parkinson's patients has increased over the recent years with the estimated number of patients reaching 6.1 million compared to the 2.5 million patients back in 1990 [2]. PD was firstly described in 1817 by the British apothecary James Parkinson [3] as he separated the diseases from other tremor symptoms and multiple sclerosis. Despite the official declaration of the disease in 1817 there are accounts of tremor symptoms in both, the Bible and ancient Egyptian papyrus, suggesting that the disease has been around since the 12th century BC [4]. There are a number of different symptoms that describe PD including tremor, slowed movement and rigid muscles. Researches have shown that the causes could be both environmental and genetic as scientists have discovered specific gene mutations that could cause the disease. Additionally, the exposure to certain toxins or other environmental factors could increase the risk of being affected by the disease. [5]. It has been proven that patients suffering PD have been identified with many changes to their brains [6]. My research will be focused on identifying those changes using a specific set of machine learning algorithms. Studies suggest that neurological disorders (including PD) are related with patients developing cardiovascular diseases (CVDs) [7], meaning that the analysis completed on neurological disorders, can also potentially contribute to research of CVDs. As the pathophysiology relating the features of the brain causing people to suffer from PD is poorly understood, the project is aiming to provide further insight.

Machine learning can contribute to the identification of diagnostic features based on extracted data from the human brain through structural MRI images. The diagnostic features which this project seeks to identify are the brain areas that can be found in an MRI and can collectively define with accuracy whether an individual is a patient of a particular disease. Based on that, the primary aim of this thesis is to achieve a comprehensive comparison between different classification algorithms. The secondary aim is to compare feature selection methods in order to identify the consistency of different methods in identifying the diagnostic features. The results of the project will enable scientists to understand which are the brain areas that are related to PD as well as identifying which classification algorithm works better for the given datasets. Assuming that machine learning approach will produce models that can identify whether an individual suffers from PD, they will give scientists further clarification on the areas of the brain affected in PD. The basic concept that makes machine learning applicable to disease diagnosis it that machine learning algorithms can learn out of experience to interpret MRI brain images. The more data a machine learning algorithm is provided for training the more accurately will be able to classify the clinical population between the different diseases or between patients and controls.

Agile methodology was applied in this research; agile refers to the “ability to create and respond to change”. [4] The process was broken down into the 6 main stages shown in Figure 1.

Figure 1 The six stages of the data science project [4]

Despite Figure 1 illustrating a linear movement across the 6 stages, the project involved revisiting and refining some of the previous stages. My approach followed the fundamental idea of agile which was to develop the software in iterations that each one contained mini improvements on the code [8] minimizing the risk of bugs. One of the most important parts of the development was the literature review and the data analysis. In order to get the most out of each classification algorithm used I had to understand how it works and if it is suitable for use on the particular dataset. The algorithm development part involved the use of classification algorithms that were imported from the Sci-Kit learn library. The algorithm developed involved the collection of appropriate metrics that would enable the result analysis to be successful. The final three steps illustrate the process where the results were analyzed, reviewed and the development of the solution was adjusted as appropriate.

Background

The problem

Despite years of research our understanding of the links between brain areas and the occurrence of PD is yet poorly understood. This research will aim to identify the reasons behind the development of PD and therefore allow scientists to better understand the disease's pathophysiology. Classical machine learning classification models have become increasingly popular over the last years due to the improvement of computational performance which is accessible to most people nowadays. This study uses classical machine learning algorithms to answer the following question; Which are the diagnostic features of PD and how can they be spotted in pre-processed tabular data extracted from MRI brain images?

In order to answer the research questions of the project all the code was developed using Python as a programming language, mainly due to the variety of libraries that it offers which are related to machine learning such as Sci-Kit Learn. The platform used to develop the solution was Google Colab, a cloud-based service that allows code execution on the internet.

The dataset

My research is based on the Shared Roots dataset which was obtained from the University of Stellenbosch and it has undergone approval by the University's ethics committee. The Shared Roots' clinical population consists of 617 participants and they are separated into patients and controls. To add on that, three different diseases can be found in the dataset, Parkinson's Disease (PD), Post-Traumatic Stress Disorder (PTSD) and schizophrenia (SC). The dataset represents extracted preprocessed tabular data from MRI brain images that have been partitioned into ROIs, which allowed the further investigation of brain areas that were suspected to have a correlation with the occurrence of one of the diseases. Additionally, the DTI (Diffusion Tensor Imaging) dataset was provided which had the same characteristics as the Shared Roots dataset but had a smaller number of clinical population (290 entries) and less variables available for each participant. DTI dataset can allow the better analysis of location and orientation of brain's white matter tracts. [9] White matter's main function is to enable the communication between the different brain areas where it can also affect learning and multiple brain functions. [10]

For each patient, in both datasets, demographic information was provided, such as age and sex, as well as the data on the different areas of the brain and a diagnosis was given. Furthermore, in Shared Roots dataset there are labeled columns that indicate whether a particular feature is part of the hippocampus (see Appendix A). In total, Shared Roots dataset consists of 187 columns (features) and 129 of them were used in order to classify the given participants into the different classes. The demographic data on each participant was excluded during the classification process in order to allow the identification of the diagnostic features that may define the occurrence of PD. DTI dataset contained 48 different brain features that were used for classification.

Aims

The primary aims of this study are:

1. The comprehensive comparison between different classification algorithms.
2. The comparison of feature selection methods in order to identify the diagnostic features of PD (brain areas that are highly correlated with PD occurrence).

Research Questions

In order to achieve the comparison between the different classifiers a research will be contacted in order to identify the most suitable algorithms to be used for the particular problem. An appropriate platform has to be selected where the code will be developed and results will be analysed. In the pre-processing phase of the study the data will be cleared and prepared to be used by the classifiers. Then, the model development and evaluation will take place and subsequently by using those models, feature selection methods are going to be implemented. Classification methods are going to be repeated based on the selected features. Finally, the whole process is going to be repeated several times and results will be stored for analysis and visualization.

Classical Machine Learning

Classical machine learning has become extremely popular over the last years as it can be applied in many different fields such as medical diagnosis, traffic prediction, email spam filtering. It is considered a subset of Artificial Intelligence and as by definition, it is "the study of computer algorithms that improve automatically through experience" [11]. There are many different machine learning algorithms which are used to make predictions upon a desired field. The process involves the splitting of the available data into train and test sets. Train set is used to allow the model to learn how to classify an object where the test set is used to evaluate the performance of the model. Then, the model developed can be used to predict the class for an object that has the same features as the ones in the train and test set. Machine learning is associated with statistics but due to the volume of computation needed to construct a model it can only be done through a computer. The basic idea of machine learning is to create algorithms that can learn how to predict the classification of different objects based on their experience. For example, if an algorithm is presented with the different characteristics from different plants, then it should be able to classify them with an accuracy higher than selecting their class randomly. Machine learning is separated into three main categories: supervised, unsupervised and reinforcement learning.

Supervised Learning

As by definition, supervised learning is "the machine learning task of learning a function that maps an input to an output based on example input-output pairs." [9] Supervised learning is used in this research project and it applies when the class that each object belongs to is known to the algorithm; meaning that the model can be evaluated based on the true predictions made for each record. In other words, each individual record in the data contains different scores for some specific features and the label of the class that the particular record belongs to. Then the algorithm learns to map the different records based on their variable values to the classes given. After the learning process, the supervised learning algorithm will try to map the test set records to the classes that are already known from the train set. [12]In this paper, 8 different supervised machine learning algorithms were used.

Model Fitting

Each model in a machine learning algorithm has to be fitted using the train set in order to enable classification of the given records with the highest accuracy possible. The accuracy is given as a number from 0 to 1. A well fitted model means that it can classify with high accuracy other similar datasets (with the same given variables) but that is not always the case. During the fitting process, each algorithm will attempt to best map the different variable values to the given class labels. Then those results are compared to the actual values of the classes and the accuracy value can be found. The accuracy score is defined from the equation shown in Figure 2.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Figure 2 Formula to calculate the accuracy of a model

There are 3 different types of fit (Figure 3) that could occur after the training of a model with the best outcome being the 'good fit' or 'robust', meaning that the model has achieved high accuracy but it is not adapted specifically for the particular data. In many occasions a model is 'overfitted', meaning that it learns to classify the train set so much that its performance on other datasets is poor. That means that the line (function) that tries to separate the different classes is too closely fit to a limited set of data points. [13] Due to the fact that the algorithm may be trained too much, it has as a result the function to take into consideration noise that could exist in the dataset as well as accounting in extensive detail the fluctuation of the different data points. The resulting graph will look like the "Overfitted" in Figure 3. [2]

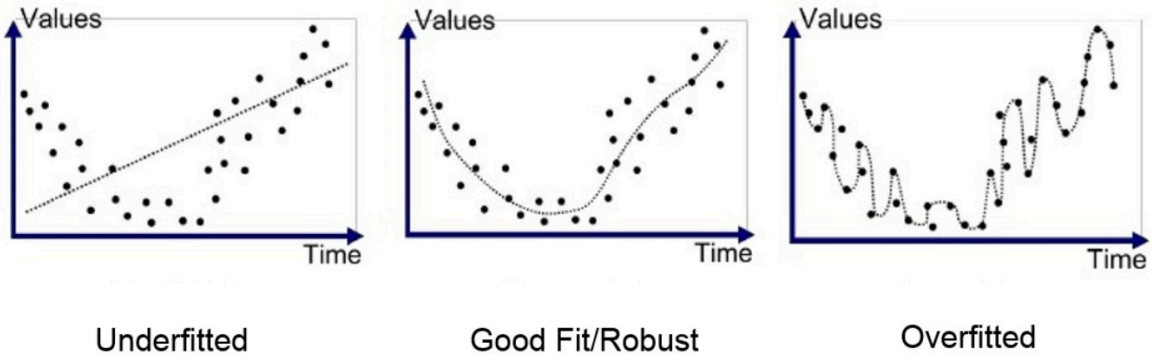

Figure 3 Different Fitting Types [2]

The exact opposite of overfitting is underfitting which can also be seen in Figure 3. It occurs when the model during the training process was unable to determine the structure of the data and thus results into poor classification results on both test and train datasets. [14] Underfitting can occur when the dataset is too small meaning that there was not enough data for the classifier to train on or when a linear model is fitted onto non-linear data.

Data exploration methods

Data exploration will allow an 'overview' of the given dataset allowing some helpful observations to be conducted prior to the classification process. A satisfactory data

exploration can lead to better decisions when choosing the classifiers to use and thus achieving better performance models. The data exploration methods applied in the particular paper are Principal Component Analysis, t-Distributed Stochastic Neighbour Embedding, t-test, and correlation matrices.

Principal Component Analysis (PCA)

When working with objects consisting of large number of variables (more than 2 or 3); a dimensionality reduction process may be necessary. It can be achieved using a technique called feature extraction. In feature extraction, a new feature is created for every existing feature in the dataset where each one of the new features created is a combination of all the other features in the dataset. Every new feature created has data from all of the features in the dataset hence even when dropping some of them, information about all of the features still exists; this represents dimensionality reduction. Thus, 2-dimentional representation of a multiple dimension dataset is possible. The new feature vectors are created after normalizing the dataset and subsequently calculating the eigenvectors and the corresponding eigenvalues of the normalized results. An eigenvector is a non-zero vector that is being changed by a scalar factor when a linear transformation is applied to it. The eigenvalue is the factor by which the eigenvector is multiplied. [15]

t-Distributed Stochastic Neighbour Embedding (T-SNE)

T-SNE is primarily used for reducing the dimensions of a multidimensional dataset in order to be presented in a low-dimensional space. The algorithm creates a probability distribution between pairs of neighboring data-points using Gaussian distribution. It then randomly plots the data points in a low-dimensional space and does the same thing for probability distribution using student t-distribution as it has heavier tails. Finally, it moves the points in the low-dimensional space such that they reflect the probabilities calculated in the high dimensional space. [16]

T-test

T-test is used to examine whether there is a significant difference between 2 groups of data points; it tests the mean values of the two groups and can determine if they come from the same population. T-test can determine whether the 2 groups have a significance difference when the p-value is smaller than a threshold number (in this study $p=<0.05$ was the threshold).

Correlation Matrix

A correlation matrix is a table with dimensions n*n were n is the number of features in the data. Each row and column represent a different feature meaning that each cell in the table shows the relationship between those two particular features. It is mainly used to show a summary of the dataset as well as present any possible patterns in the data.

Classifiers used in the paper

Linear Discriminant Analysis (LDA)

The first classifier used was Linear Discriminant Analysis which is considered a linear classifier. That means that it can classify a given object based on the result obtained from the linear combination of the features. LDA is used for dimensionality reduction in order to allow a 2-dimentional representation of the data. In the dataset used there were more than 100 different metrics for each object that translates to more than 100 dimensions needed to plot the results. The main goal is to achieve the dimensionality reduction and the best possible class separation on the 2-d plane. There are 5 main steps that the LDA algorithm uses in order to classify the given data. The first one is to compute the d-dimensional mean vectors; meaning that for every class in the dataset it will create a 1-dimentional vector containing the mean value of each variable that maps to the particular class. The second step is to compute the scatter (4*4) matrices which include both the within-class and between-class scatter matrix (see Appendix B for how they are calculated). Then the generalized eigenvalue problem is solved for the inverse of the within-class scatter matrix and the between-class scatter matrix. The fourth step is to select linear discriminants for the new feature space. In order to achieve that, the vectors obtained from the third step are sorted by decreasing eigenvalues and the k eigenvectors with the largest eigenvalues are selected to achieve the dimensionality reduction. After that, k* d-dimensional eigenvector matrix W is calculated and so the dimensions are reduced from four to two-dimensional feature space. . The final step is the transformation of the samples into the subspace by using the equation shown in Figure 4 (using the matrix W from step 4) [17]

$$\boldsymbol{Y} = \boldsymbol{X} \times \boldsymbol{W}.$$

(where $\boldsymbol{X}$ is a $n \times d$-dimensional matrix representing the n samples, and $\boldsymbol{Y}$ are the transformed $n \times k$-dimensional samples in the new subspace).

Figure 4 Equation to transform the samples to the new subspace [17]

Random Forest

Random forest classifier is an improved version of the decision tree classifier. The word "forest" implies that it consists of multiple decision trees (Figure 5). 'Random' means that the algorithm selects random samples from the data to create a bootstrap dataset. Variables are then randomly selected to construct the decision tree. Once a number of decision trees is created, a new object can be classified based on its attributes. The classification that each decision tree makes counts as a vote to the particular class and so the class with the most votes is selected for the respectable object. Finally, the records not included in the bootstrap dataset are used to validate the performance. [18]

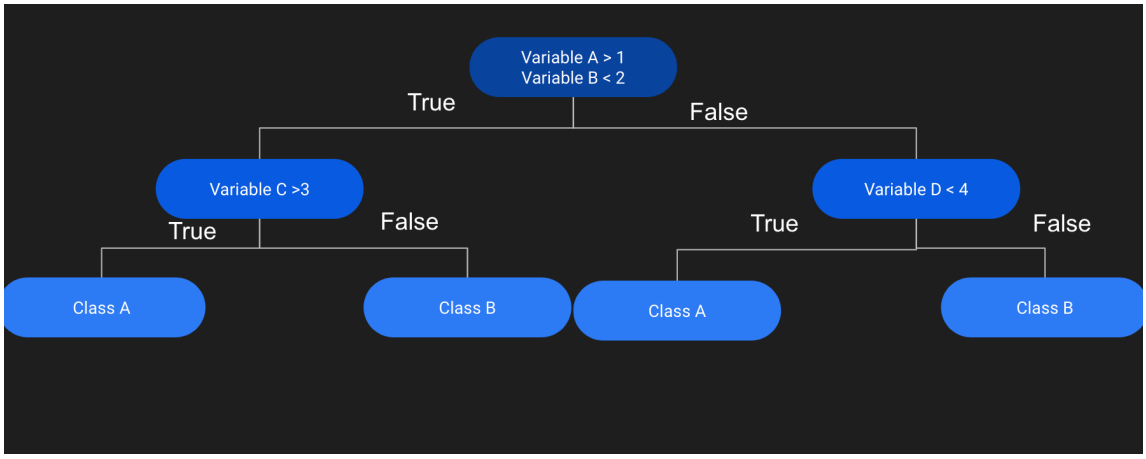

Figure 5 An example of a decision tree

K-Nearest Neighbours (KNN)

In KNN, an object is assigned a class defined by the popularity vote of its neighbors as shown in Figure 6. 'k' indicates the number of neighbors that are 'voting' for the class to be assigned. For example, if k=5 then the algorithm will look at the 5 closest points from the object to be classified and decide on the class to be assigned to it. A KNN model is trained is by storing the feature vectors of each of the different variables and class labels of the train set. After the training phase, the distance for a newly inserted object to its neighbors is usually calculated using Euclidean distance. [19] Prior to the training phase, dimension reduction should occur as the data consists of more than 2 dimensions which can be achieved using PCA or LDA.

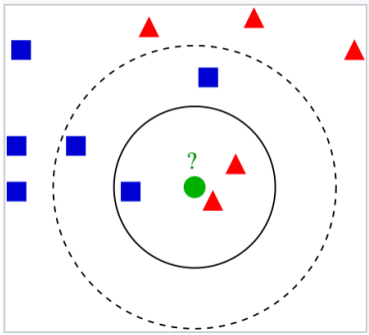

Figure 6 K-NN classifier looks at the classes of the K-Nearest Neighbours and accordingly decides on the classification of a new object [19]

Naïve Bayes Classifiers

Naïve Bayes defines a family of different classifiers; Bernoulli Naïve Bayes (BNB) classifier was used in this research project. All the classifiers in this family follow the Bayes theorem shown in Figure 7 that describes the probability of event A happening given that event B has occurred. During classification, B can be swapped with the variables of x_1, x_2 … x_n where n is the number of variables. In BNB the predictors are Boolean variables so that all the variables used to define an object can take Boolean values (True or False). That makes the BNB extremely fast to execute and at the same time makes it a good-performing classification algorithm. [20]

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Figure 7 Bayes Theorem [20]

Support Vector Machines (SVM)

Similar to Naïve Bayes Classifiers, Support Vector Machines define a family of classifiers; SVC (Support Vector Clustering) classifier was used. In this type of classification each data point is plotted into an n-dimensional space where n is the number of variables in the data. Consequently, classification is performed by finding a hyperplane (or set of hyperplanes) that can efficiently separate the classes defined (Figure 8). The best hyperplane to be used is the one that can maximise the distance between the two classes. Support vectors are the data points which are closer to the hyperplane and therefore they are the ones which will define its position. [21]

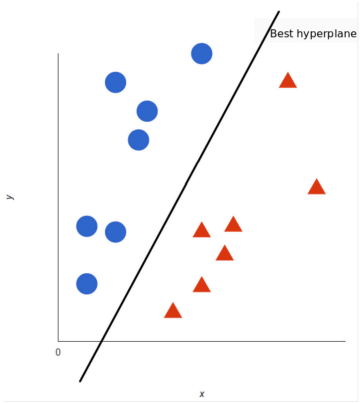

Figure 8 The best hyperplane that can separate the two classes [17]

Logistic Regression

Logistic Regression model uses the probability of a certain class occurring in the data. That means that this classifier can predict whether something is true or false and this can be achieved by adding an 'S' shape logistic function to the data (Figure 9) in order to separate the classes in a 2-d plane. The difference with linear regression is that the line is fitted in the data based on the 'least squares', whereas in logistic regression 'Maximum likelihood' is applied. The logistic function is able to map any variable's value to 0 or 1 but in order to set the function at the best possible position in the graph, the decision boundary is used which defines a threshold value of whether an object is classified as true or false. Then a cost function is used to represent the optimization objective and therefore attempting to minimize the cost function. The cost value can be decreased using the gradient descent which can 'feel' the change of cost value and therefore try to find the global minimal point where the cost function is the lowest possible. Gradient Descent is an optimization strategy that changes the parameters of the Sigmoid function in order to reduce the cost function. The algorithm takes steps proportional to the gradient of the function as shown in Figure 10. [22]

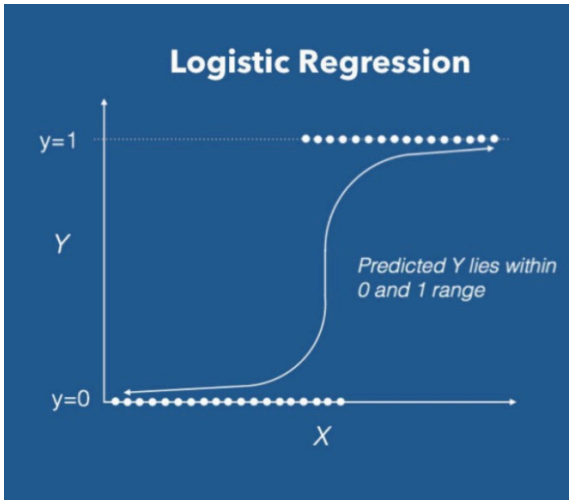

Figure 9 The logistic function or Sigmoid Function [23]

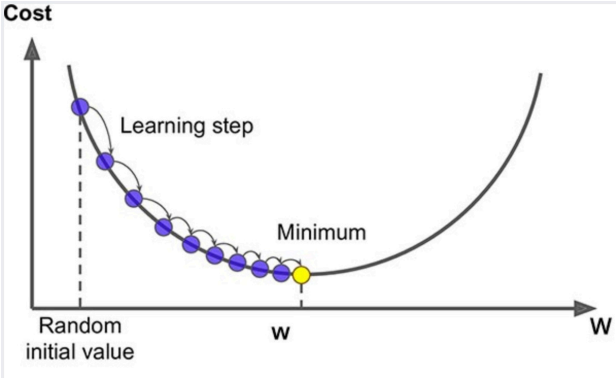

Figure 10 Gradient Descent [22]

Stochastic Gradient Descent Classifier (SGD)

SGD is an optimisation method but the SGD classifier is a linear classifier that uses SGD for optimisation. SGD classifier uses the gradient descent used in the logistic regression classifier as its optimisation function but weights are changed after considering one point rather than the whole training set. The word stochastic means random as the point selected to adjust the weights is selected randomly. This improves the performance of the algorithm compared to the logistic regression. After the algorithm selects a random point then it updates the gradient function using the coordinates of the newly selected point. It then calculates the step size by multiplying the gradient with the learning rate (a parameter set to influence the magnitude at which step size changes in each iteration). Finally, the new parameters are calculated by subtracting the step size from the old parameters and the iterations continue until the loss function is at its minimum. [24]

Multilayer Perceptron (MLP)

MLP classifier can be considered one of the easiest artificial neural networks (ANN) to be implemented for classification (Figure 11). It usually consists of at least three fully connected layers (input, hidden and output) and each node in the input layer uses a non-linear activation function such as rectified linear or logistic. The nodes of the input layer represent one column (feature) of the given dataset. Nodes in the output layer represent the different classes in the

dataset. Activation functions are used to map the weighted inputs to the output of each neuron. Learning is achieved by changing the weights that connect the neurons using backpropagation; this can be associated with the least mean square algorithm in linear algebra. MLP can also use the gradient descent algorithm in order to change its weight so it can achieve higher accuracy in the model. [25] Overall artificial neuron networks work similarly to a biological one but the artificial neural network takes probabilistic inputs and converts them to output classes.

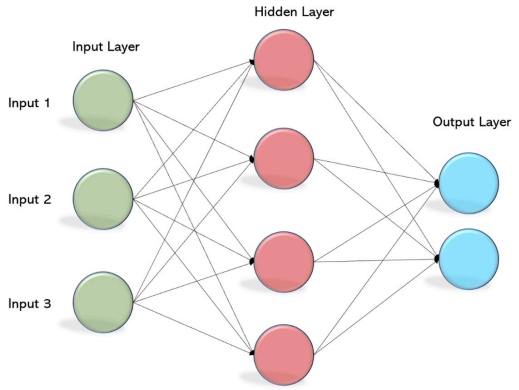

Figure 11 A simple ANN with 3 layers [25]

Hyperparameter Tuning

All the classifiers described have a different set of hyperparameters that can be altered in order to affect the performance of the model. A base model is the model that classifies the given data using the default parameters set while a tuned model is the one that tests a number of different combinations of those parameters and chooses to use the one that makes the model perform better. The parameters that are going to be tested are defined in a grid with the different values that each hyperparameter can take.

Evaluation of performance

There are many different ways in which models' performance can be evaluated. The ways described below are the ones that have been chosen in order to validate the models developed in this study. In order to find the most suitable algorithm that can identify the diagnostic features of PD, a large number of metrics were taken to achieve a comprehensive comparison.

Train and Test accuracy

Train and test accuracy will represent the level at which the model is able to predict correctly whether an object belongs to the expected class. Train and test accuracy are calculated in the same way but they are applied on different datasets. Train accuracy shows the correct predictions made on the training set where test accuracy shows the correct predictions made on the test set. The way that they are calculated is by adding the correct predictions (TP and TN) and then divide that number by the total predictions (TP+TN+FP+FN). The result will be a number from 0 to 1 and will show the percentage of accuracy.

Nested k-Fold Cross-validation

Nested Cross-validation (CV) uses the idea of train-test split but in a more sophisticated way. CV separates the data into *k* partitions (folds) and uses the one partition for testing and the rest are used for training as shown in Figure 12. *k* test accuracy scores are subsequently obtained, implying that the result will be more representative in comparison to using a single test set. Nested CV uses an inner and outer loop of CV. The outer loop breaks up the whole data into k folds and applies the iterations of the CV described and so for each split the model is trained, tested and the best hyperparameters are selected. The inner loop takes the train set for each of the splits occurred in the outer loop and for each one of them, it trains, validates and determines the optimal hyperparameters set by the average of the validation errors. In summary, a nested k-fold CV will help in validating the performance of a model by giving more metrics; thus, allowing better conclusions to be made. The value is obtained using the "cross_val_score" from the Sci-Kit learn library.

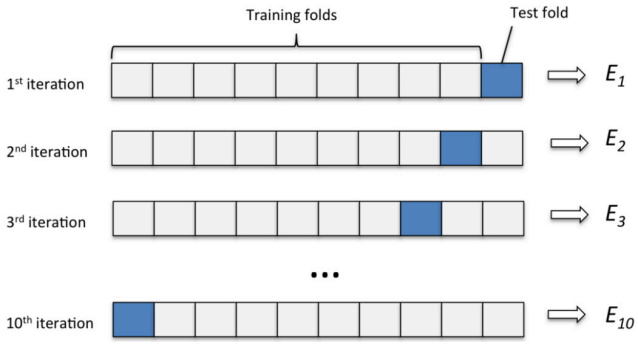

Figure 12 10-Fold Cross Validation [26]

Confusion Matrix

A confusion matrix is used in order to describe the results of classification models as shown in Figure 13. The numbers of correct and false predictions are summarised in the four boxes of the table. Each term in the table has a different meaning; TP (true positives) means the predictions that predicted correctly to belong to a particular class, TN (True Negatives) means the predictions made were correctly predicted not to belong to a particular class, FN (False Negatives) means the object actually belongs to the class but the model predicted the opposite and FP (False Positives) means that the object didn't belong to the class and the model predicted the opposite.

	Class 1 Predicted	*Class 2 Predicted*
Class 1 Actual	**TP**	**FN**
Class 2 Actual	**FP**	**TN**

Figure 13 Confusion Matrix format [27]

Precision – Recall

Precision and recall are scores that are calculated from the numbers obtained by the confusion matrix described above. They can be calculated as an average for all the classes

that the model deals with or they can be used to represent the precision and recall of a particular class in the diagram. Recall is the ratio of correctly predicted positive objects (TP) divided by the total number of positive objects (FN and TP) as shown in Figure 14. The higher the recall, the more accurately the model can classify the given class as there will be a small number of FN.

$$Recall = \frac{TP}{TP + FN}$$

Figure 14 Recall Formula

The next score which is used to measure the performance of a model is the precision. It represents the ratio of TP compared to the total number of predicted positive objects as shown in Figure 15. In the model developed, the aim was to maximise both of these scores. [28]

$$Precision = \frac{TP}{TP + FP}$$

Figure 15 Precision Formula

Area Under the Receiver Operating Characteristics (ROC) curve

The Receiver Operating Characteristics (ROC) curve defines the performance of the model by plotting the TP rate against the FP rate at different classification thresholds. Specifically, ROC defines the ability of the classification model to distinguish between the classes. The TP rate is defined by dividing the TP with the total of TP and FN. The FP rate is calculated by dividing the FP with the sum of FP and TN. AUC (area under the curve) measures the area under the ROC curve and therefore the larger the AUC, the better the model performs. The AUC score varies from 0 to 1; where 1 represents that the classification model is perfect as shown in Figure 16. Usually, a "strong" classifier is considered to have an AUC score greater than 80%. [29]

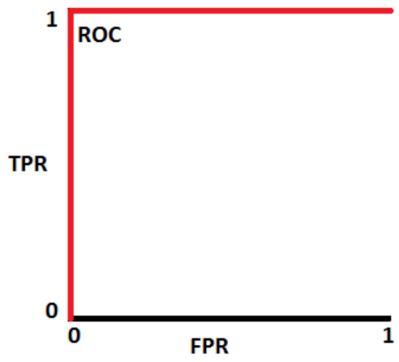

Figure 16 A ROC curve with AUC score of 1 [29]

Feature selection methods

The feature selection methods will help to distinguish the features that can influence the decisions of the classifiers the most. Therefore, by creating models that are using those selected features the classification performance increases. In this paper, 4 different feature selection methods were used.

Feature Importance in Random Forest

The Feature Importance method for the random forest classifier uses the sum of reduction in Gini Impurity from all the features in the model in order to decide which once are influencing the decisions made by the classifier the most. The Gini Impurity of a feature is defined by the probability that a randomly chosen sample in a node would be wrongly predicted based on the observed training data. In the formula shown in Figure 17, p(*i*) represents the probability of selecting a data point which belongs to class *i*, where C indicate the total classes in the data. [30] [31] In fact, Gini measures the inequality of object distribution between classes, so a lower Gini coefficient would represent an even distribution of classes. A zero Gini impurity would imply a perfect split. At the end of the procedure, the mean decrease of Gini impurity that each feature has achieved on the different decision trees is calculated so the most important features can be spotted. [32]

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

Figure 17 Formula to calculate the Gini Impurity of a given node [32]

Chi-Squared

Chi-Squared also called 'goodness-fit' is used to test the likelihood that the observed distribution fits with the expected one. Specifically, it is used to test the relationships between the features in the data. Through this test, the degree of deviation between two different features is calculated (formula shown in Figure 18). 'c' indicates the degree of freedom, "O" indicates the observed values and "E" indicates the expected values. Each variable is tested against the rest of the variables; the features that are depended on the most variables are the once selected. To conclude, a high Chi-Squared value means that the feature is more dependent on the response and therefore it's more significant. [33]

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Figure 18 Chi-Squared formula [33]

Recursive Feature Elimination (RFE)

RFE tries to select the most "valuable" feature by recursively calculating the importance of a list of features. It does that by continuously training models and based on their performance, finds the most important features. The importance of all the features in the data is calculated with ".feature_importances_" or with ".coef_". The algorithm continues by eliminating

features from the pruned list on each iteration until it reaches the desired number of features. [33]

LASSO (Least Absolute Shrinkage and Selection Operator) – SelectFromModel

LASSO does two main things; regularization and feature selection. This method puts a constraint to the sum of the absolute values of the model parameters and that sum has to be less than a pre-set value (upper bound). This is achieved by applying the shrinkage method where the classification variables are "penalized"; subsequently some them are shrunk to 0. The penalty factor is decided after cross – validation and therefore the penalization occurs in a fair manner. The variables that still have values greater than 0 are selected for the feature selection. [34]

Python Libraries

Python Language

Python is a high-level, general purpose language and was selected to be used for the particular project as it's one of the most suitable programming languages used for data science. Importantly, it has many data science-related libraries that help in the development of the various models. In addition, Python has a large community of developers so support is easily accessible.

Sci-Kit Learn

Sci-Kit Learn is one of the most useful libraries for machine-learning projects as it offers a variety of build-in functions ready to use with minimal additional coding required. Also, its documentation is supportive with code examples that enable the understanding of the features imported. To conclude, many developers choose to use Sci-Kit Learn for machine learning and therefore there is lots of support online.

NumPy

NumPy is a Python library for scientific computing that offers "a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more" [35] Most of the features referred are extremely useful for a machine learning project and therefore it is a beneficial library to include in the project.

Pandas

Pandas is a Python library that works well with tabular data which is the form of data used in the project. It is suitable for data pre-processing and analysis where the most frequently used feature included in the project is the DataFrame which is used to store the tabular data.

Seaborn

Seaborn is a Python library that is used primarily for data visualization and it's based on matplotlib library. It offers an easy high-level interface and informative statistical graphics. Seaborn will be used for plotting the results obtained from the classifiers.

Google Colab

Google Colab is a cloud-based service that can be used to execute code without the need to locally install any libraries. One of the biggest advantages of using Google Colab is that it has free access to CPU, GPU and TPU. The only disadvantage is that it has a maximum runtime of 8-12 hours and it may cause problems as some machine learning tasks require long runtimes.

Approach

Problem Statement and requirements

The problem, as stated previously, is to identify the features that are correlated with the occurrence of PD in the clinical population. To achieve that, a comprehensive comparison of the different classification algorithms would have to take place to allow the identification of the diagnostic features. In total, 8 different classification methods are going to be tested on the given datasets. The solution should be able to identify the most important features that can boost the performance of the classifiers. To achieve that, 4 different feature selection methods will be tested. In order to evaluate the classification and feature selection methods, a variety of metrics were used to monitor their performance and thus making the comparison more complex.

The first requirement is to perform an analysis which will clarify the classifier that outperforms the rest algorithms tested on the given datasets. In addition to that, the system, through comparing different feature selection methods, should be able to identify the most important diagnostic features influencing the decision of the classifiers

The code will be developed using multiple notebooks with Google Colab, which can execute commands on the cloud and therefore no installations would be needed. Furthermore, Google Colab can provide a GPU which can speed up the process as the training of the different models would require a lot of processing power.

Data flows of the solution

Based on the requirements of the system a data flow diagram will be used to visualise how the data will flow through the system and eventually enable the statistical analysis and representation of the results. In the real system, multiple train - test sets were used for model training and testing but for simplicity reasons and to avoid repetition on the diagram it was assumed in the diagram that the system deals with one Train/Test dataset. All different Train/Test sets come from the clinical population but each one of them classifies different things and they are listed below;

- Disease classification using Shared Roots dataset
- Disease classification using DTI dataset
- Patient – Control classification using Shared Roots dataset
- Patient – Control classification using DTI dataset

All of the above classification approaches produce more than one train-test split as different subsets of the data are used. For example, in all of the four approaches classification after feature selection will require a new train – test split.

As shown in Figure 19, the data flow diagram of the system begins by taking as an input the Shared Roots or DTI datasets. After the pre-processing phase the data is split into train and test sets and by using the different classification algorithms the models are derived. Following that, model evaluation occurs and the results are stored. Then tuning occurs by testing the different hyperparameters and deriving the best performing model which undergoes evaluation. Next phase is the feature selection where the number of features that are used for train and test are reduced to the selected ones and the model is again tuned and validated

with the same process as before. The system's output are the performance results of all the different models form all the different classification algorithms as well as the best selected features from the dataset.

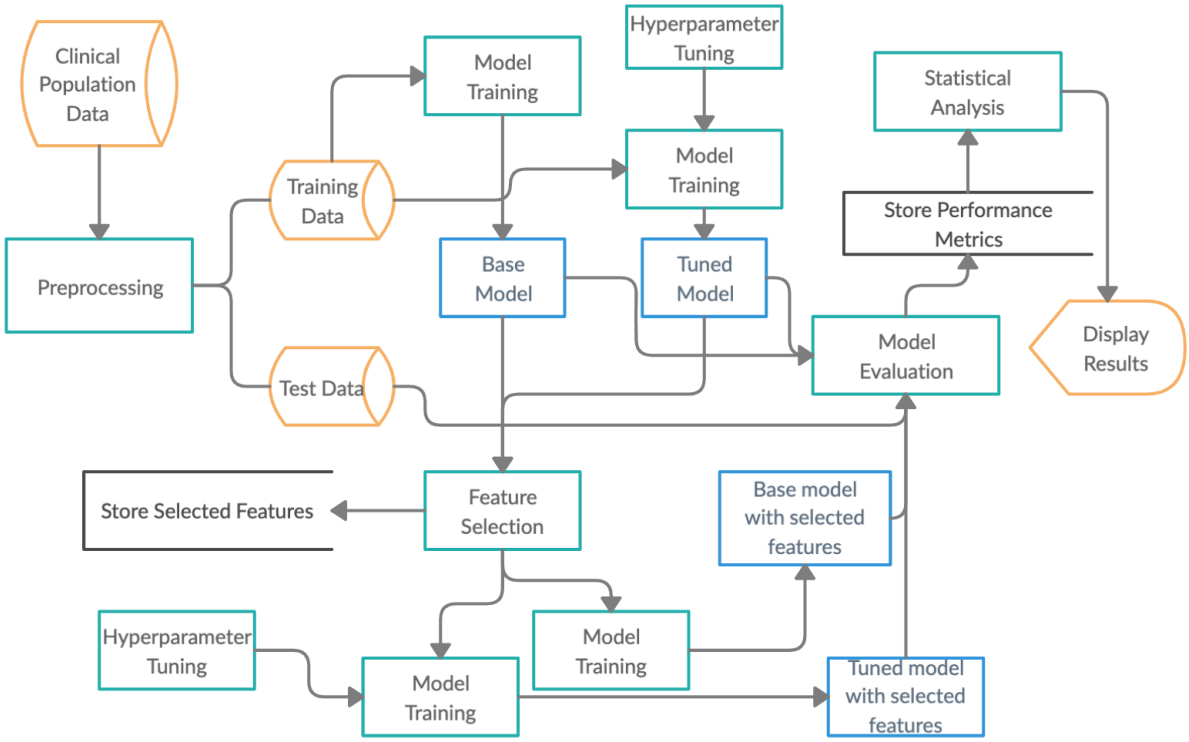

Figure 19 Data flow diagram of the system

Data pre-processing

In data pre-processing, the data will be obtained from the Shared Roots and DTI datasets will be pre-processed accordingly in order to allow the classification algorithms to operate as expected. The given "xlsx" files containing all the data can be read using the Pandas library (as a dataframe). The features that were specified not to be used in the classification will subsequently be dropped. Next steps will be to:
a) look for null records and remove them from the dataframe and
b) standardize the features that will be used for classification as most of the classifiers require the inputted data to be standardized.
Final steps in the preprocessing phase is to declare the different subsets of the data such as the PD patients and controls subset.

Data Exploration

Demographic Analysis of Clinical Population

Demographic analysis will show some characteristics for the clinical population which are useful to know prior to classification. It helps to develop an understanding of the age, sex and disease of each of the participants. Different graphs can be used to visually present the different aspects of the data such as showing the value counts of each disease or showing the number of patients and controls. Furthermore, the mean age per disease can be presented

as well as the male against the female population in the different diseases. That will give an idea whether a disease is strongly related with a gender or a group age.

Correlation Matrix, PCA, T-test, Data Distribution

Different methods will be used in order to gain further insight into the data. As described in the background section correlation matrix will be used to check the relationship between all the features in the data.

Additionally, PCA will indicate which way the data can be clustered better. One possible graph can be the plot of PD vs PTSD vs SC patients which will show if the diseases can easily be separated using PCA. The hippocampus dataset can be used as well in order to test how the different diseases can be separated with the same method. Furthermore, PCA can be used to plot the groups of patients and controls. PCA will help to visualize if and how different data groups can be clustered and thus can be a guide towards which groups would be more beneficial to classify as they will produce better performing models.

The last two data exploration methods that will be used are the t-test and data distribution. Firstly, t-test will determine whether statistical difference exists between the average age of the patients versus the average of the controls; it is expected that there will not be a statistical significance between the mean age of male and female groups. It will also show whether a particular variable of a group is normally distributed; an important aspect that most classifiers take as an assumption. Finally, plotting the data distribution of randomly selected variables from the data will enable interpretation of whether the selected features are normally distributed. As there are about 130 features in the data it would be impossible to plot all of them and thus the best practice that could be followed is to plot a subset of them.

The Two Classification Approaches

The Process

On both datasets that will be used there is an almost equal number of patients and controls. Based on that, one classification approach will be to test the classifiers into classifying the patients from the controls. Three different diseases can be found in the data so another classification target will be to classify the patients into the 3 diseases (PD, PTSD, SC). An overview of the classification approaches that will be followed can be seen in Figure 20.

In both methods the same 8 supervised machine learning classifiers will be used (including linear and non-linear classifiers);

- LDA
- Random Forest
- K-Nearest Neighbors
- Support Vector Machine (Support Vector Clustering)
- Naïve Bayes (Bernoulli)
- Logistic Regression
- Stochastic Gradient Descent
- Multilayer Perceptron

The classifiers stated above were all evaluated using the same metrics. Those metrics are:

- train accuracy
- test accuracy
- ROC AUC score
- Precision (for PD class)
- Recall (for PD class)
- nested cross validation score

1st classification approach - Disease Classification

In order to classify the diseases, only the patients will be used. First step will be to filter the data to get all the patients and then split them into train and test sets using the "train_test_split" from Sci-Kit learn library. It has to be confirmed that each class in both train and test set are equal in order to ensure fairness, otherwise, resampling has to be made to adjust the numbers of each class. Subsequently, the data will be classified using the 8 supervised machine learning classifiers stated. Each one will produce one model and then they will undergo hyperparameter tuning; a process which produced 8 new models. All of the 16 models that will be produced per classifier will be stored together with their performance metrics.

2nd classification approach - Patient – Control Classification

Prior to the classification data will be spited into the train-test sets similarly to the disease classification and resampling method will be applied as long as the dataset is imbalanced. At first, all participants from the clinical population will be used in order to classify them into patients and controls. Then, classification will be focused into classifying the PD patients against the PD controls. The same 8 classifiers will be used producing the 8 base models and then hyperparameter tuning process produced 8 more models. Similarly, to the other classification process, the results of each of the models were stored in a dataframe for further analysis.

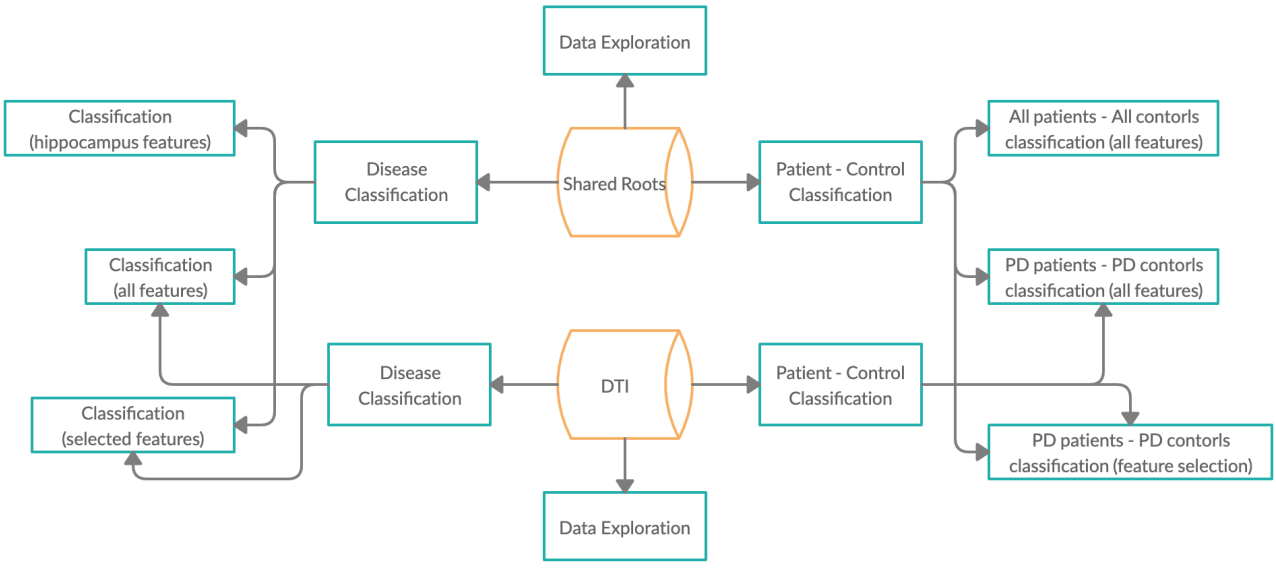

Figure 20 Classification approaches on the different datasets

Hippocampus Data Classification

The Hippocampus subset of Shared Roots dataset will be used in disease classification. Following the method described above 8 base models and 8 tuned models will be the output per classification approach. As the hippocampus is an area of the brain that is strongly related to memory it would be reasonable to observe how the classification performs using a small subset of the whole data. Hippocampus dataset consists of 24 features out of the 129 that are used in the base classification procedures.

Feature Selection

4 feature selection methods will be tested in order to identify which are the most important features in the dataset. The 4 methods of feature selection that will be used on both classification approaches are;

- Feature importance in random forest classifier
- Recursive feature elimination (RFE)
- Chi-Squared
- LASSO – SelectFromModel

Feature importance in random forest classifier will select all the features that are above the mean score calculated by the particular method while the rest of the features will select the 30 best features. It is not feasible to use all the models for feature selection methods since only certain classification models are eligible to obtain selected features. A new dataframe has to be set up in order for the results of the feature selection methods to be stored. The resulting dataframe will contain all the features of the dataset and it will show how many times each feature was selected by one of the methods. The features selected more than a threshold number of times by the different methods will then be used for classification. It is expected that results will be more accurate than previous attempts which were carried out without feature selection. Classification with selected features implies that only a fraction of the available data will be used and so a new train – test split will be required. As in previous classification approaches, the 8 classification algorithms will be tested producing 8 base and 8 tuned models. All of the models produced will be stored in the dataframe with the results that will be used for the classification analysis. At the end of the study there will be 2 lists with selected features from Shared Roots dataset classification approaches (patient – control and disease). From the DTI dataset there will be 2 more lists with the selected features. Each dataset will produce one list with selected features which will contain the common dataset variables from the two lists.

Classification Performance Analysis

During the different classification methods all the metrics of the models created will be saved in the particular classification's result data frame. From the Shared Roots dataset two different data frames with results will be created; one with the patient – control classification and one with the disease classification results. The same method for storing the classification results will apply for the DTI dataset, meaning that in total there will be 4 different data frames containing classification performances. In that dataframe the performance metrics of each classifier will be stored. The characteristics of each model will also be stored, for example whether it was a tuned model or not. The resulting data frames will contain all the data

needed in order to conclude which is the best performing classifier out of the 8 that have been used in the study. The analysis will be able to prove whether the features selected are actually the most important ones. In the analysis a number of different graphs will be produced that will represent the performance of each classifier under the different classification approaches. Using graphs, it will be possible to compare the classifiers and hence arrive to a fact-based conclusion on the best classification algorithm.

The classification analysis procedure can be summarized in Figure 21. The results obtained from the classification will be used to produce 3 main graphs each.

- A graph (or two for presentational purposes) presenting how each algorithm performed on average for each metric.
- A summarized version of the first graph so that there is one bar per classifier instead of 6 (allows better comparison)
- A graph that shows the performance of the classifiers over the different classification approaches (e.g. classification with selected features or not).
-

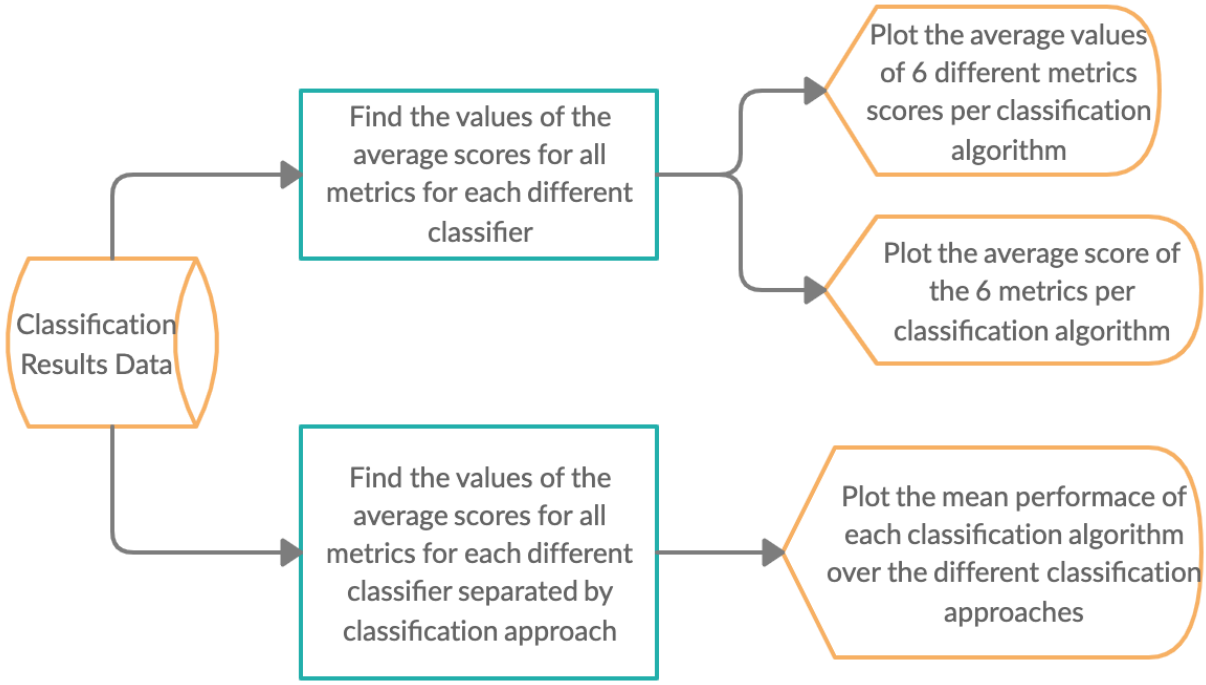

Figure 21 Classification analysis procedure

Implementation

Data Pre-processing

Shared Roots

The pre-processing state of the Shared Roots dataset was split into two sections; disease classification and patient – control classification.

The common pre-processing steps taken for both classification approaches were to load the Shared Roots dataset using the Pandas library and then merge the 2 datasets that were provided; one dataset including demographic information (Age, Sex, Patient or Control) and the other dataset containing the metrics from the MRI images. Then, features of the data that were not going to be used such as “Hippo_Comments” or “QC_Comment” were dropped. In total, 50 features were dropped as it was advised by the supervisor that it would be good practice not to be used for classification. Following that, the data was cleared from null records and then all the metric variables were standardized using the z-score; altering them in order to have a mean score of 0 and a standard deviation of 1 (code used for standardisation in Figure 22). Standardisation was essential due to the fact that classifiers work better with standardized values. Then, data was split into x and y in order to allow classification to occur, where the “x” subset included all the variables that were used for classification (Figure 23) and “y” included all the class labels. For the disease classification, y values hold the values for the 3 different diseases whereas for the patient – control classification the y values indicate the patient – control status.

A subset of Shared Roots, which included the hippocampus features was used in order to test whether the hippocampus alone can define the patient’s disease. For that purpose, a new dataframe was defined with all 29 hippocampus features identified in patients. By using the “Hippo score”, which indicates the quality of the readings from scale of 1 to 3, every reading above or equal to 2 was selected. Then by following the same standardization process, the hippocampus features were standardized with the z-score and then split into the x and y axis.

```
for variable in df[df.columns[0:119]]:
    df[variable] = zscore(df[variable])
```

Figure 22 Standardizing the data

```
[18] X.head(5)
```

	Left-Lateral-Ventricle	Left-Inf-Lat-Vent	Left-Thalamus-Proper	Left-Caudate
0	4.970711	4.109754	-0.174759	-2.378764
1	-0.168598	0.210404	-0.448193	-0.655294
3	2.722057	2.238224	-1.449177	1.073352
4	0.057782	-0.401048	-1.299043	-0.780202
5	0.259967	0.178482	-0.294095	0.932354

5 rows × 119 columns

Figure 23 The dataframe containing the X values showing only the first 5 rows and 4 columns out of the 119

For each of the two classification approaches stated above, one dataframe was created in order to store the different performance metrics for each individual classifier tested. For the patient – control classification, the data frame created was named "patient_control_results" and it consisted of the following fields;

- classifier_name (model's classification algorithm name)
- hippo_datset (Boolean, states if only hippocampus features were used)
- all_patient_all_control (Boolean, states if all diseases were used, false value mean that only PD patients and controls were used)
- tuned (Boolean, indicates whether the model is tuned or not)
- feature_selection (Boolean, indicates if the model used selected features)
- train_accuracy (Train accuracy of the model)
- test_accuracy (Test accuracy of the model)
- roc_auc_score (ROC AUC score of the model)
- precision (Precision score for classifying PD patients)
- recall (Recall score for classifying PD patients)
- nested_cross_val (Nested Cross Validation score)

The equivalent data frame for the disease classification was exactly the same but instead the "all_patient_all_control" field was dropped as there was no need to separate the patients from the controls due to the fact that no controls were used in this particular approach.

The final action taken prior to both classification approaches was to split the data into train and test sets. Sci-Kit learn library's function "train_test_split" was used; the ratio for splitting between the train and test sets was 8:2 respectively. After separation each class for the patient – control classification had the values shown in Figure 24.

```
[ ] y_test.value_counts()

1.0    60
2.0    56
Name: Patient_Control, dtype: int64

[ ] y_train.value_counts()

2.0    232
1.0    228
Name: Patient_Control, dtype: int64
```

Figure 24 Value counts for train and test sets (Shared Roots)

However, due to the imbalanced class distribution the train – test split for the disease classification had to undergo the resampling procedure. That method concatenates the train and test sets back into one and then it resamples them based on the number of data points that the smaller class in population has. The code used to achieve that is shown on Figure 25. The "replace" parameter was set to false in order to prevent one data point to be chosen twice. After resampling method for the train sets the same method was applied to resample the x and y test sets. The value counts for each class is shown in Figure 26.

```
#as the dataset is imbalanced i have to make some adjustments using oversampling
# # Separate input features and target
X = df_patients.drop(['Cohort', 'Age', 'Sex', 'Patient_Control'], axis = 1)
y = df_patients['Cohort']

# # setting up testing and training sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=100)

# # concatenate our training data back together
X_new = pd.concat([X_train, y_train], axis=1)

# separate minority and majority classes
PD = X_new[X_new.Cohort=='pd']
PT = X_new[X_new.Cohort=='pt']
SC = X_new[X_new.Cohort=='sc']


pd_resample = resample(PD,
                       replace=False,
                       n_samples=len(PD),
                       random_state=27)
pt_resample = resample(PT,
                       replace=False,
                       n_samples=len(PD),
                       random_state=27)
sc_resample = resample(SC,
                       replace=False,
                       n_samples=len(SC),
                       random_state=27)

# # combine majority and upsampled minority
upsampled = pd.concat([pd_resample,pt_resample,sc_resample])

# # check new class counts
upsampled.Cohort.value_counts()
```

Figure 25 The resampling method to equate the classes

```
[ ] y_train.value_counts()

pd    46
pt    46
sc    37
Name: Cohort, dtype: int64
```

```
[ ] y_test.value_counts()

sc    11
pd     7
pt     7
Name: Cohort, dtype: int64
```

Figure 26 Value counts for train and test sets after resampling

Finally, inner and outer cross validation folds were set in order to achieve the nested cross validation. "KFold" from Sci-Kit learn was used and both inner and outer loops were defined with 4 splits. In summary, after the completion of the pre-processing phase, 4 sets of data had been created in order to train and validate the models. X_train and X_test stored the values of the different variables for training and testing while y_train and y_test stored the class labels for training and testing.

DTI dataset

The DTI dataset consists of a subset of the clinical population from the Shared Roots dataset. Similarly, to Shared Roots, the dataset was used for both disease and patient – control classification. The pre-processing phase started by dropping the columns that didn't have any significant information such as the subject id or additional comments for each participant. A problem that arose was that the decimal values were recorded using a comma (,) as a separator instead of a full stop (.). The problem was solved by creating a loop that swapped the comma with a full stop and so the numbers can be identified as floats and not as objects. Subsequently, by using the "FA QC score" (field to measure the quality of readings); the records with insufficient quality were dropped to eliminate anomalous readings and the data was filtered in order to use only patients. As done in the previous pre-processing procedure all the readings were standardized using z-score. Final step was to initialise 2 different data frames that would store the results for the two classification approaches. Similarly, to the previous data frames created for storing the results, the following fields were created:

- classifier_name (Stores the classification algorithm that the model is using)
- tuned (Boolean value that states if the model is tuned or not)
- feature_selection (Boolean value that states if the features that the model is using to classify are based on feature selection or not)
- train_accuracy (Train accuracy of the model)
- test_accuracy (Test accuracy of the model)
- roc_auc_score (ROC AUC score of the model)
- precision (Precision score for classifying PD patients)
- recall (Recall score for classifying PD patients)
- nested_cross_val (Nested Cross Validation score)

For both classification approaches the train-test split method had to follow the resampling method as the value counts of each class were not equal. The same resampling methods applied in Figure 24 were implemented, resulting in more evenly distributed classes. The downside for the disease classification was that the volume of the dataset was not as large as

the Shared Roots dataset one and so each class had about 20 data points per class for training and 6 for testing sets.
The resampling process for the patient – control classification had as a result 9 data points to be taken for each class as a train set and 3 data points for each class as a test set. Both resampling methods did not use the replacement method therefore all the data points could be selected only once. Finally, the resampling process inner and outer cross validation splits were set to 4 with shuffled enabled.

Data Exploration

The demographic analysis for both DTI and Shared Roots datasets involved the correlation matrix, PCA, data distribution and t-test.

Firstly, a t-test was performed in order to ensure that there was no significant difference between the PD patients and PD controls. Then, different PCA took place in order to present how the different groups in the data can be plotted in a 2-dimentional plane. The pairs of groups that were plotted are:

- PD patients vs PD controls
- PD vs PTSD vs SC (only patients)
- Patients vs Controls (all diseases)

The graphs were created twice for the Shared Roots; one with all features and one using only the hippocampus ones. For the DTI dataset, both PCA and T-SNE were used to plot the graphs stated.

Seaborn library was used in order to view the correlation between the different features in the dataset. A large correlation matrix was implemented for both Shared Roots and DTI datasets while in Shared Roots dataset a correlation matrix was created for the hippocampus features as well. For readability purposes, the correlation matrix for all the features in the Shared Roots dataset was split into two sections. Next part of the data exploration process was to plot randomly selected features in order to determine their distribution. As in the correlation matrix, Seaborn library was used to plot the distribution of 4 different variables. In addition to that, different plots were constructed that show the mean age of the participants for each of the 3 diseases. Finally, the mean age per disease and sex was plotted.

General Considerations for classification

Classification Procedure

In order to make classification comparison a fair test, a model was trained and tested 5 times and all the results were recorded in the particular classification approach results' dataframe. On each iteration, the model was declared with the inputted method of the particular classification algorithm and that process differs when the model was a base (using the default hyperparameters) or a tuned one. If a base model was declared, a variable was assigned with the particular classifier's default method. For example, in order to derive the base model of LDA, a variable named "model" was assigned the "LDA()" method. In contrast, when the model was a tuned one, a grid with the testing parameters was created and a grid search (or random search, according to the classifier) was conducted on the particular classifier's parameter grid. All the parameter grids that were used for tuning can be found in Appendix

C. The resulting model, either tuned or base, was trained using the “fit” method and predicted labels were calculated using the “predict” method. By using x and y train and test sets as well as the predicted y labels the six metrics were calculated.
All the metrics were calculated using the code snippet shown in Figure 27. Following the performance metrics, the characteristics of the model were recorded in order to allow comparison between classifiers and different subsets to be made. Finally, all the metrics and characteristics of the model were saved into the results’ dataframe.

```
[ ]  for i in range (5):
         model = LDA()
         model.fit(X_train, y_train)

         # mlp_grid = GridSearchCV(MLPClassifier(),mlp_param_grid, n_jobs = -1, cv=in
         # mlp_grid.fit(X_train,y_train)
         # model = mlp_grid.best_estimator_

         y_pred = model.predict(X_test)

         train_accuracy = model.score(X_train, y_train)
         test_accuracy = model.score(X_test, y_test)
         y_prob = model.predict_proba(X_test)
         roc_auc = roc_auc_score(y_test, y_prob, multi_class="ovo",average="macro")
         precision = precision_score(y_test, y_pred, labels=['pd'], average='micro')
         recall = recall_score(y_test, y_pred, labels=['pd'], average='micro')
         scores = cross_val_score(model, X_test, y_test, cv=outer_cv)
         nested_cross_val = scores.mean()

         hippo_datset = False
         classifier_name = 'lda'
         tuned = False
         feature_selection = True
         classification_results = classification_results.append({'classifier_name': c
```

Figure 27 An example of the loop that was used to test and store the results for each model

Feature Selection Procedure

The feature selection procedure was almost identical for all the classification approaches as it involved the 4 different feature selection methods stated in the approach section. The first method was the feature importance for random forests; every feature that scored a value above average was selected. The rest of the methods that were used elected the 30 best features. A minimax scaler was used in order to make all the X values positive as the following methods needed only positive numbers to operate. Then Chi-squared and Recursive Feature Elimination (using logistic regression base model) were implemented. The last feature selection method carried out was LASSO that used a number of different models that was different in every classification approach. Each one of the models used for the LASSO method selected its 30 most important features. The resulting dataframe of the above procedure can be seen in Figure 28.

Patient – Control Classification

Classification

Classification for the Shared Roots and DTI datasets was carried out using the 8 classification algorithms on each one of the 3 approaches described below. (each classifier used the same test and train sets)

- Classifying all patients against all controls (not used in the DTI dataset classification due to the poor classification performance)
- Classifying PD patients against PD controls
- Classifying PD patients against PD controls using feature selection.

Out of the first 3 classification approaches stated above a total of 240 different models (8 classifiers * 2 models each (base and tuned) * 5 repeats for each model * 3 classification approaches) were saved in the Shared Roots dataset classification results' data frame. The DTI dataset results' dataframe produced 160 different models as it used 2 of the 3 classification approaches.

Feature Selection

The feature selection procedure was followed as described in the *General considerations for classification* section. All the 4 feature selection methods were followed and the models that were implemented for the Shared Roots dataset in the LASSO method were the:

- Tuned model of LDA
- Base model of SGD
- Tuned model of Bernoulli Naïve Bayes
- Base and tuned models of logistic regression

The models used for LASSO (DTI dataset) were:

- LDA base model
- logistic regression tuned model
- base SGD model

The only difference with the method described in the *general considerations* was that the feature importance method for random forests was not used due to the poor performance of the classifier on the particular dataset. The features' data frame depicting which features were selected by each of the classifiers can be seen in Figure 28.

	Feature	Chi-2	Lasso_lda	rfe_lg	Lasso_rf	Lasso_sgd	Lasso_nb	Lasso_logistic_regression	Lasso_lg_tuned	Total
1	lh_rostralmiddlefrontal_thickness	True	True	True	True	True	True	True	True	8
2	Right-VentralDC	True	True	True	True	True	False	True	True	7
3	Right-Caudate	True	True	True	True	True	False	True	True	7
4	Left-Caudate	True	True	True	True	True	False	True	True	7
5	L_HATA	True	True	True	True	True	False	True	True	7
...	...	...	...	...	...	...	...	...	...	...
115	L_hippocampal-fissure	False	False	False	False	False	False	False	False	0
116	L_CA4	False	False	False	False	False	False	False	False	0
117	L_CA3	False	False	False	False	False	False	False	False	0
118	CC_Central	False	False	False	False	False	False	False	False	0
119	CC_Anterior	False	False	False	False	False	False	False	False	0

119 rows × 10 columns

Figure 28 A part of the dataframe used to store the features selected

Classification with selected features

From the Shared Roots a total of 23 features were selected from the 4 feature selection methods while 25 were selected from the DTI dataset. The method described in the *general considerations* was followed for each of the 2 datasets but with the only difference being that only selected features were used for classification.

Classification Analysis

Initially, the results' dataframe for each individual dataset was loaded; 720 records from the Shared Roots dataset and another 560 from the DTI dataset. The data frames used to store the results can be seen in Figure 28.

First step was to convert the Boolean columns into integer columns storing 1 for 'True' and 0 for 'False'. Then the "get_dummies" method was applied for the classifier name column in order to allow the aggregation of the data to occur. Data aggregation occurred for every 5 rows (as each model was tested for 5 times with the same parameters). Following that, new subsets of the aggregated dataset were created, one for each classifier and one for each classification approach. The code used for deriving the subsets of the different classifiers can be seen in Figure 29. The results obtained from Shared Roots and DTI datasets were used separately for creating graphs that are described in the '*Approach section'* for the results' analysis.

classifier_name	hippo_datset	all_patient_all_control	tuned	feature_selection	train_accuracy	test_accuracy	roc_auc_score	precision	recall	nested_cross
lda	False	True	False	False	0.717391	0.482759	0.482143	0.482759	0.482759	0.55
lda	False	True	False	False	0.717391	0.482759	0.482143	0.482759	0.482759	0.55
lda	False	True	False	False	0.717391	0.482759	0.482143	0.482759	0.482759	0.55
lda	False	True	False	False	0.717391	0.482759	0.482143	0.482759	0.482759	0.55
lda	False	True	False	False	0.717391	0.482759	0.482143	0.482759	0.482759	0.55
...	...	...	...	...	...	...	...	...	...	
mlp	False	False	True	True	0.835165	0.608696	0.613636	0.608696	0.608696	0.55
mlp	False	False	True	True	1.000000	0.521739	0.534091	0.521739	0.521739	0.48
mlp	False	False	True	True	0.835165	0.608696	0.613636	0.608696	0.608696	0.43
mlp	False	False	True	True	0.868132	0.521739	0.522727	0.521739	0.521739	0.30
mlp	False	False	True	True	1.000000	0.478261	0.484848	0.478261	0.478261	0.39

Figure 29 Example from the results' dataframe

```
[ ]  #get the mean scores for each classifier from all the different runs

     lda_scores = patient_control_results[patient_control_results['classifier_name_lda']==1]
     knn_scores = patient_control_results[patient_control_results['classifier_name_knn']==1]
     lg_scores = patient_control_results[patient_control_results['classifier_name_logistic_regression']==1]
     mlp_scores = patient_control_results[patient_control_results['classifier_name_mlp']==1]
     nb_scores = patient_control_results[patient_control_results['classifier_name_nb']==1]
     rf_scores = patient_control_results[patient_control_results['classifier_name_rf']==1]
     sgd_scores = patient_control_results[patient_control_results['classifier_name_sgd']==1]
     svm_scores = patient_control_results[patient_control_results['classifier_name_svm']==1]
```

Figure 30 Code used to define one dataframe containing all the models for each classifier

Disease Classification

Classification with all features

After the train-test split, the next step would be to run the 8 classification algorithms using the procedure declared in the *general considerations*. This classification approach involved the use of all the available features from both DTI and Shared Roots datasets. In specific, 129 features were used for Shared Roots dataset classification and 64 for DTI dataset. The difference with patient – control classification was that the y axis class labels represented the diseases.

Classification with hippocampus features

Classification with the hippocampus only occurred using the Shared Roots. By extracting the hippocampus subset, the data would be split into x train, x test, y train and y test. Those subsets would only involve the features that are located in the hippocampus of the brain for the X sets and the diseases on the y sets. This classification approach followed the methodology stated in the *general considerations* and results were also obtained using the procedure stated.

Feature Selection

Feature selection procedure for both datasets implemented by following the implementation steps stated at the *general considerations for classification* section. For the Shared Roots dataset, the models used in the LASSO method were the logistic regression base and tuned models.

In contrast, the models that were used for the LASSO method in the DTI dataset were the;

- SGD base model
- logistic regression base and tuned models
- Bernoulli Naïve Bayes tuned model

All the feature selection methods appended their selected features' in a dataframe (similar to Figure 28).

Classification with selected features

The features that were used from both datasets were the ones selected more than 3 times by the feature selection methods. Two new x-axes were created (one for each dataset) for classification which consisted of 26 features for the Shared Roots dataset and 22 features for the DTI dataset. Then, the resampling method was implemented in a similar way to the one

described in Figure 25. Subsequently, the 8 classifiers were tested again using the methodology described in the *general considerations*.

Classification Analysis

In total 720 models were imported for analysis from Shared Roots dataset and 320 from the DTI dataset. The disease classification analysis was similar to the one in the patient – control classification resulting in 11 new subsets for the Shared Roots dataset (Figure 31): The same occurred for the DTI dataset despite the fact that there were no hippocampus results. Subsequently, those subsets were used in order to plot the different graphs that have been described in the approach section for the results' analysis.

```
#get the mean scores for each classification method

hippo_results =  (classification_results[classification_results['hippo_datset']==1])
base_classification_results=classification_results[(classification_results['hippo_datset']==0)
                                        & (classification_results['feature_selection']==0)]
feature_selection_results = classification_results[(classification_results['hippo_datset']==0)
                                        & (classification_results['feature_selection']==1)]

#get the mean scores for each classifier from all the different runs

lda_scores = classification_results[classification_results['classifier_name_lda']==1]
knn_scores = classification_results[classification_results['classifier_name_knn']==1]
lg_scores = classification_results[classification_results['classifier_name_logistic_regression']==1]
mlp_scores = classification_results[classification_results['classifier_name_mlp']==1]
nb_scores = classification_results[classification_results['classifier_name_nb']==1]
rf_scores = classification_results[classification_results['classifier_name_rf']==1]
sgd_scores = classification_results[classification_results['classifier_name_sgd']==1]
svm_scores = classification_results[classification_results['classifier_name_svm']==1]
```

Figure 31 Code used to create the different subsets of the results' data frame

Extraction of diagnostic features

The diagnostic features were extracted based on the feature selection methods that occurred in the 4 classification approaches. From the Shared Roots dataset, in disease and patient – control classification 2 data frames were created containing the selected features for each classification approach. The diagnostic features were defined to be the features that were common in both lists. The same methodology was followed for the 2 DTI dataset classification approaches, resulting in 2 lists with 11 diagnostic features of PD in total.

Results and evaluation

Summary of the results

The results of this project are aiming to give answers to two research questions:

1. Which is the most accurate classical machine learning algorithm for the particular datasets.
2. Which are the features of the brain that have high correlation with the occurrence of PD

Many different plots illustrate with detail how each of the different classifiers behaved on the different classification methods and hence define the best one out of the 8. Additionally, the diagnostic features for PD were identified by the feature selection methods that were implemented in both disease and patient – control classification on the two datasets.

Data Exploration Results

Data exploration was first applied in the Shared Roots dataset. As can be seen in Figures 32, 33 and 34 the PCA shows how the different groups of the datasets can be plotted in a 2-dimentional space. Those figures suggest that disease classification performs better; clustering can be performed to separate the classes while in patient-control plots the data points are mixed. Additionally, when the PD patients and controls are compared, the result seems to be better in clustering compared to the same graph with all the diseases.

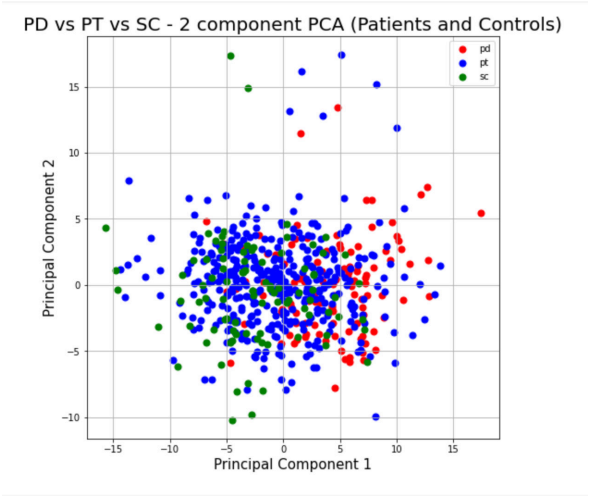

Figure 32 PCA for the 3 diseases on Shared Roots

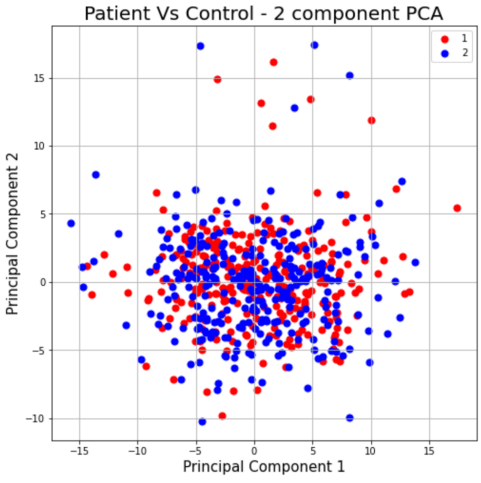

Figure 33 PCA for patients Vs controls on Shared Roots

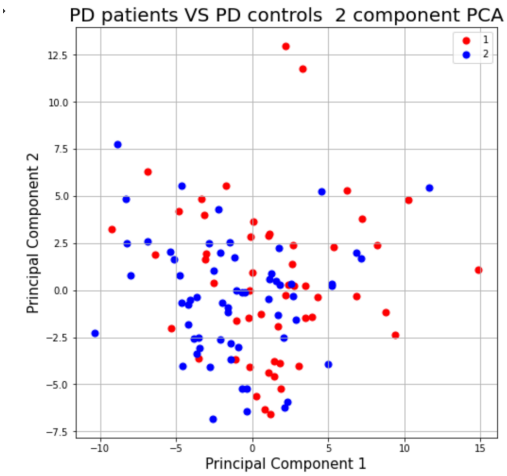

Figure 34 PCA for PD patients and controls on Shared Roots

The same 3 graphs (Figures 35, 36, 37) were recreated using only the features that were part of the hippocampus and the outcomes imply the same observations as previously. However, it seems that the clustering for the disease classification could be worse using the hippocampus features rather than using all the features. Patient – control classification using all the diseases again implies that the performance of the classifiers would not be as high as classifying the PD patients with the PD controls.

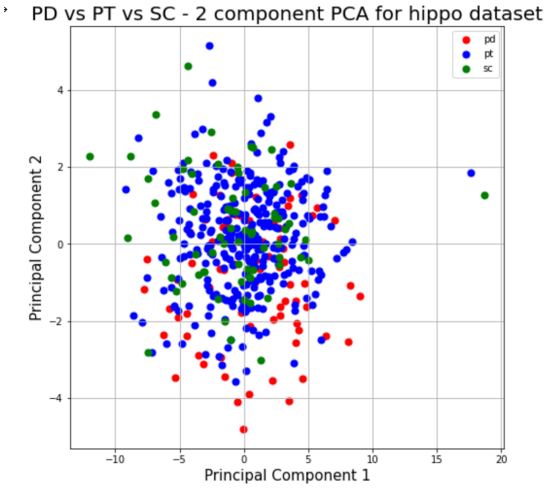

Figure 35 PCA for the 3 diseases using the Hippocampus features from Shared Roots

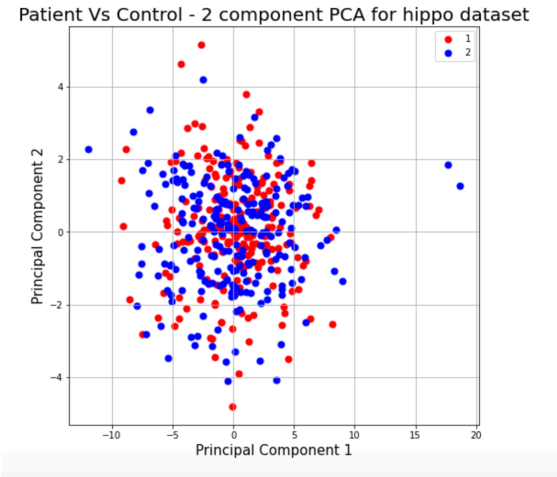

Figure 36 PCA for patients Vs controls using the hippocampus features from Shared Roots

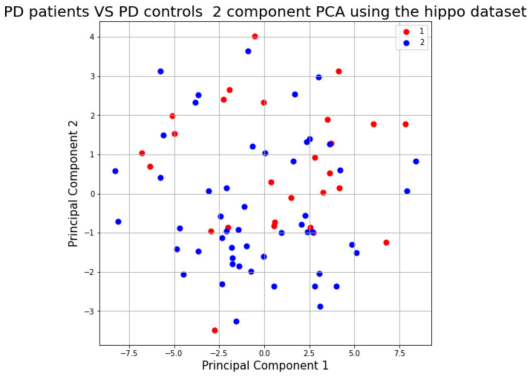

Figure 37 PCA for PD patients Vs controls using the hippocampus features from Shared Roots

Next step was demographic analysis where different plots for the various groups of the clinical population were created. Figure 38 shows that PD patients are the oldest on average within the clinical population compared to schizophrenia patients which are the youngest. Figure 39 shows that there are more females than males and relates to the fact that more females in the population suffered from PTSD. On the other hand, males are the majority of patients for in PD and SC groups.

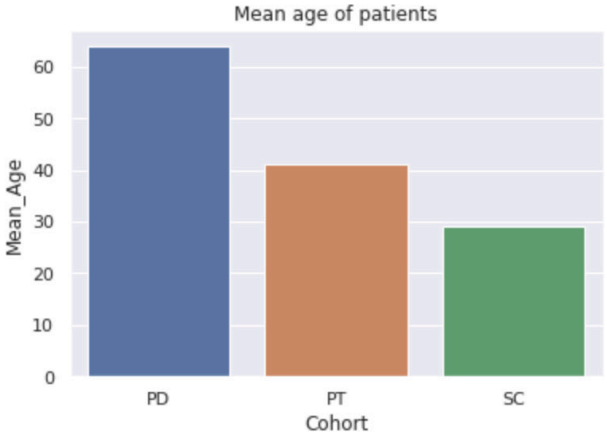

Figure 38 Bar plot for the disease count in Shared Roots

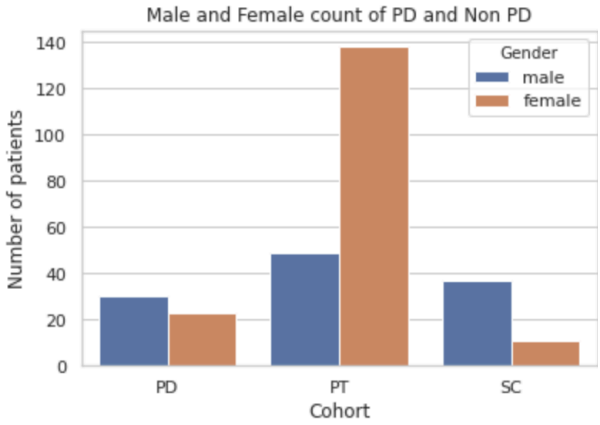

Figure 39 Bar plot showing the ages of patients according to their disease using Shared Roots

The correlation matrix on Figures 40 and 41 shows the correlation between each feature in the Shared Roots dataset. The data was sliced into two figures due to the large volume of the data. It can be seen that there are clusters of features that have strong correlation between them while there are other areas in the graph that show that features have a very weak correlation. That implies that classification after feature selection would possibly improve the classification performance. Additionally, Figure 42 shows the correlation matrix with the features that belong to the hippocampus. All the features have a strong correlation between them; the only fields that are not correlated with the rest are age and sex which will not be used in the classification.

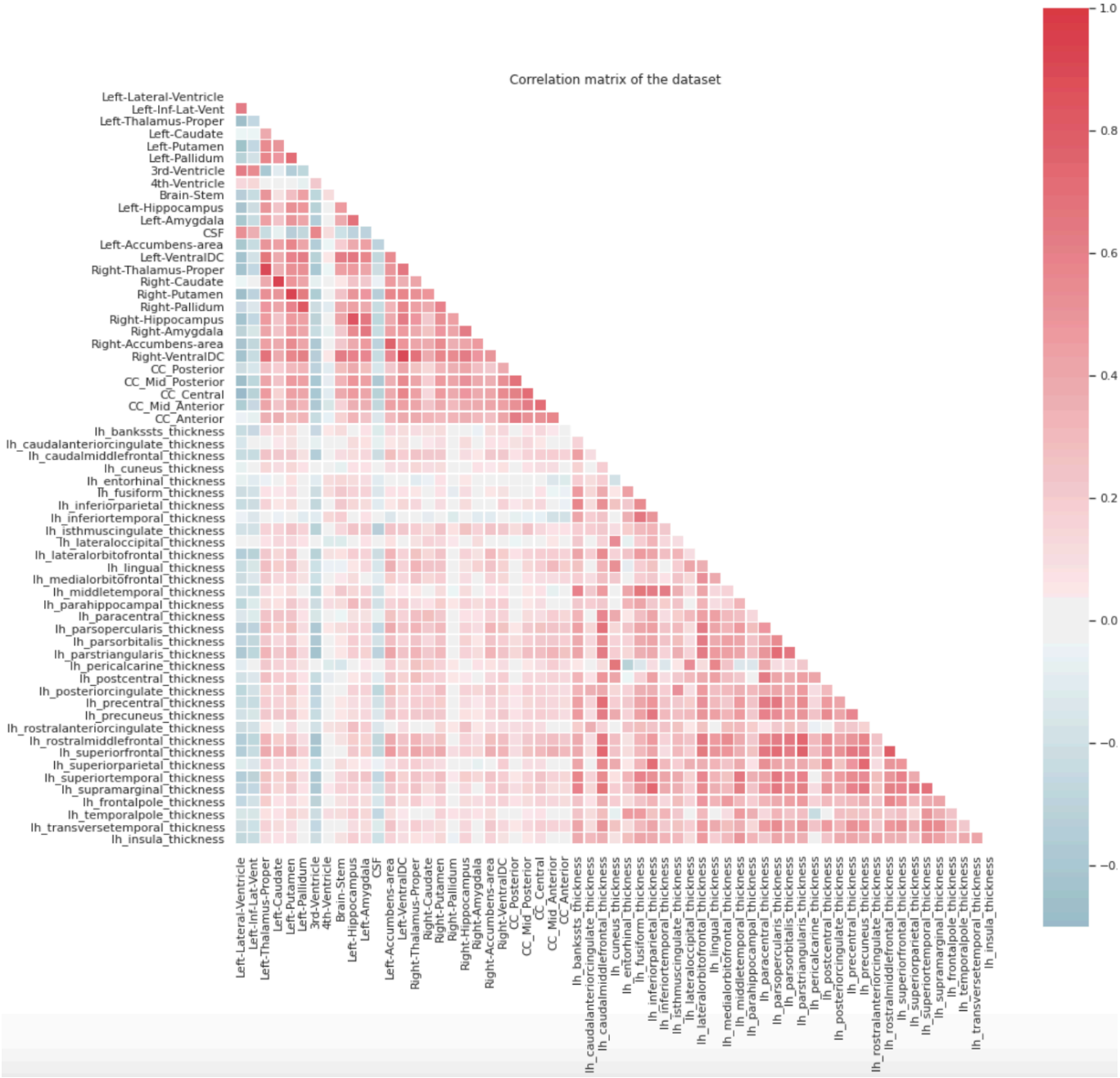

Figure 40 Correlation Matrix for Shared Roots dataset (first 62 features)

Figure 41 Correlation Matrix for Shared Roots dataset (last 62 features)

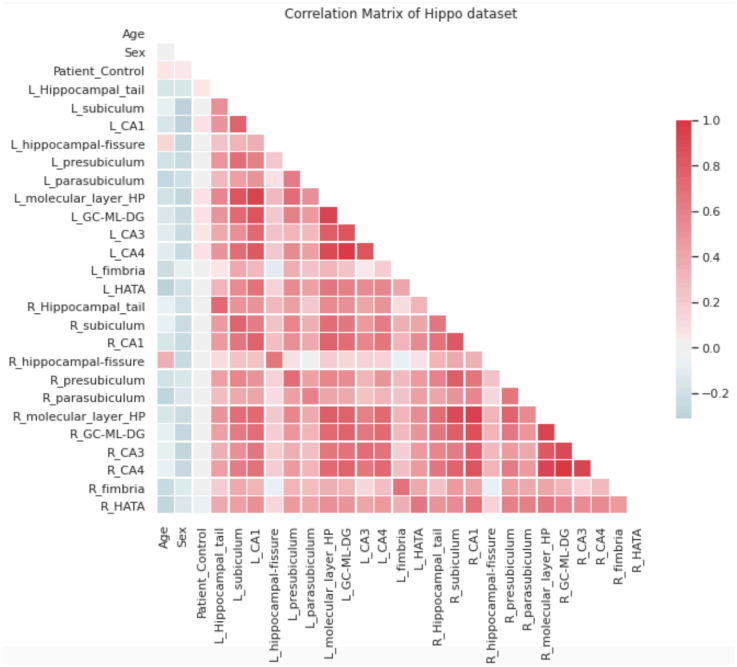

Figure 42 Correlation matrix for the hippocampus features in Shared Roots

Figure 43 presents the distribution of 4 randomly selected variables, where their distribution has been observed not to be completely normal. That could cause problems in classification as some of the classifiers are assuming that the data is normally distributed. The fact that the distribution is almost normal implies that the consequences from that could be minimal.

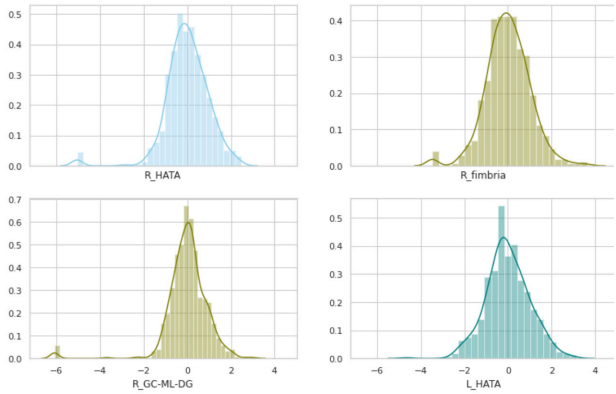

Figure 43 Distribution of randomly selected variables from Shared Roots

Last part of the data exploration for the Shared Roots dataset was to perform a t-test to check difference in mean age between the PD patients and controls (Figure 44). Controls' age distribution is not normal and that could cause anomalies on the patient-control classification. Furthermore, the average age difference between the patients and controls is significantly different implying that there could be possible anomalies in the patient – control classification. The large t-statistic value also indicates the large difference between the means of the two groups.

```
Sample mean age of Parkinsons patients:  63.90766141716043
Sample mean age of Parkinsons controls:  59.18762671001832
Difference:  4.72
P-Value: 0.0022718193277579016
T-statistic: 3.1239753625145346
Is Patient and Control age difference significant?  [ True]
Patients age is normal?  True
Controls age is normal?  False
```

Figure 44 T-test results for age difference in Shared Roots

The second dataset that was used for classification was the DTI dataset where the same data exploration process was followed. Firstly, the different group counts that appear on the dataset were plotted using bar plots. As shown in Figure 45, approximetly half of the population are patients and the other half are controls. The disease count separated by gender is similar to the Shared Roots dataset as a large fraction of the patients suffer from PTSD which are mainly females. In addition, the number of the total PD patients and controls is small and that could cause problems in the respective classification.

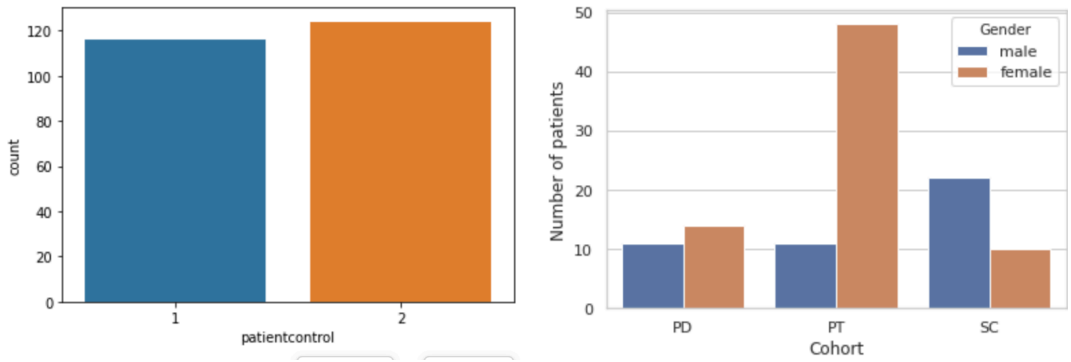

Figure 45 Patient - Control count (left) and disease count separated by gender (right) for DTI dataset

Next part of the data exploration was to create 2-component PCA graphs. In Figure 47 it is clear that the diseases can be clustered and that could imply that a linear classification method can be really beneficial for the disease classification. On the other hand, the separation between the patients and controls (Figure 48) does not show any possible clustering, thus, a possible classification between patients and controls would not have good results. Finally, Figure 46 shows that there is a small number of PD patients and controls but despite that there could be some separation between the classes.

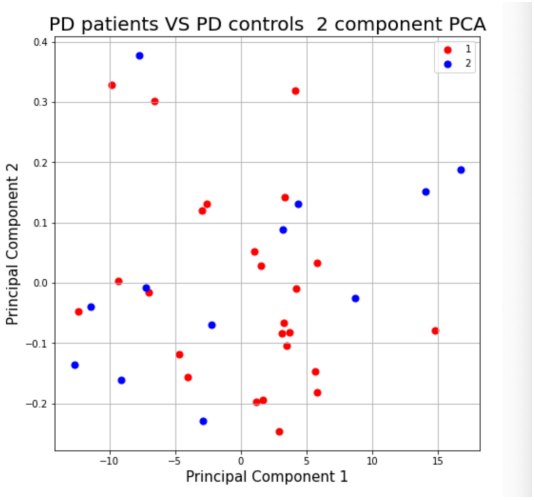

Figure 46 PCA for PD patients and controls for DTI dataset

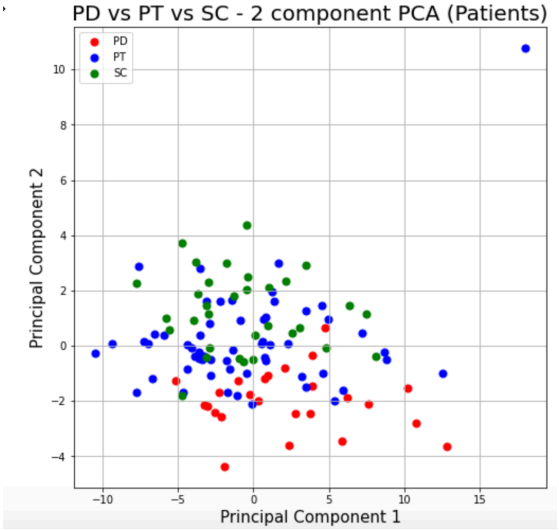

Figure 47 PCA for the 3 diseases on DTI dataset

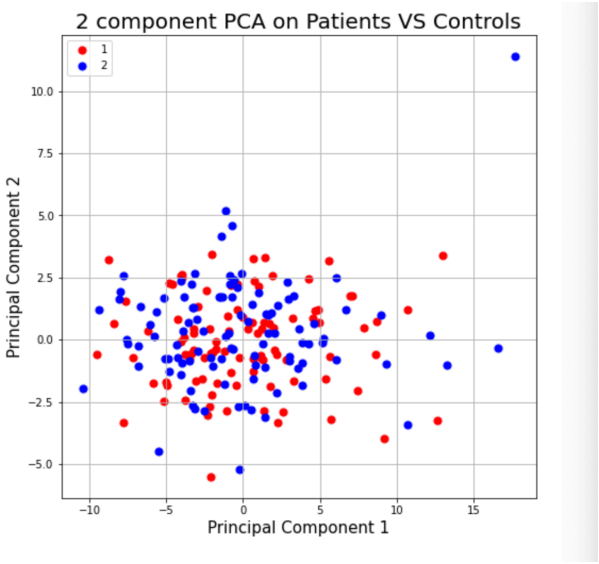

Figure 48 PCA for Patients - Controls on DTI dataset

The same graphs were then reconstructed but that time T-SNE analysis was used instead of PCA (Figures 49, 50). The results were about the same as PCA analysis implying that disease classification can be done with accuracy where the patient – control classification could have less accuracy, close to chance level. Despite that, PCA overall achieved better separation between the classes especially when the classes were diseases.

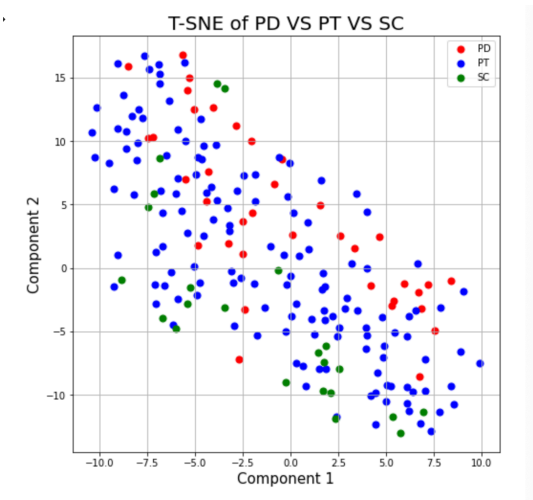

Figure 49 t-SNE analysis for the 3 diseases on DTI

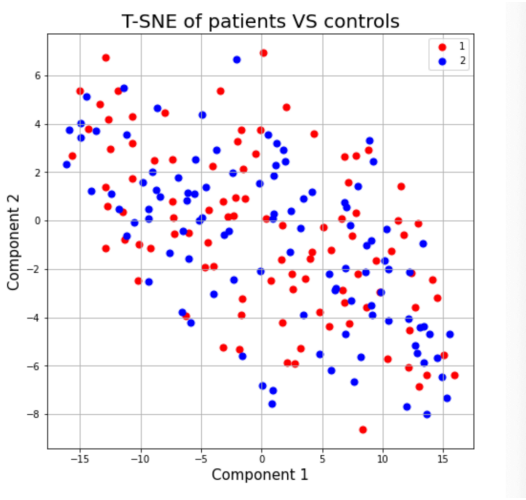

Figure 50 t-SNE analysis for patients - controls on DTI

A correlation matrix was used to plot all the features of the DTI dataset. As it can be seen in Figure 51, the features of the brain are strongly correlated between them where the rest of the features have a weak correlation with the other of the variables.

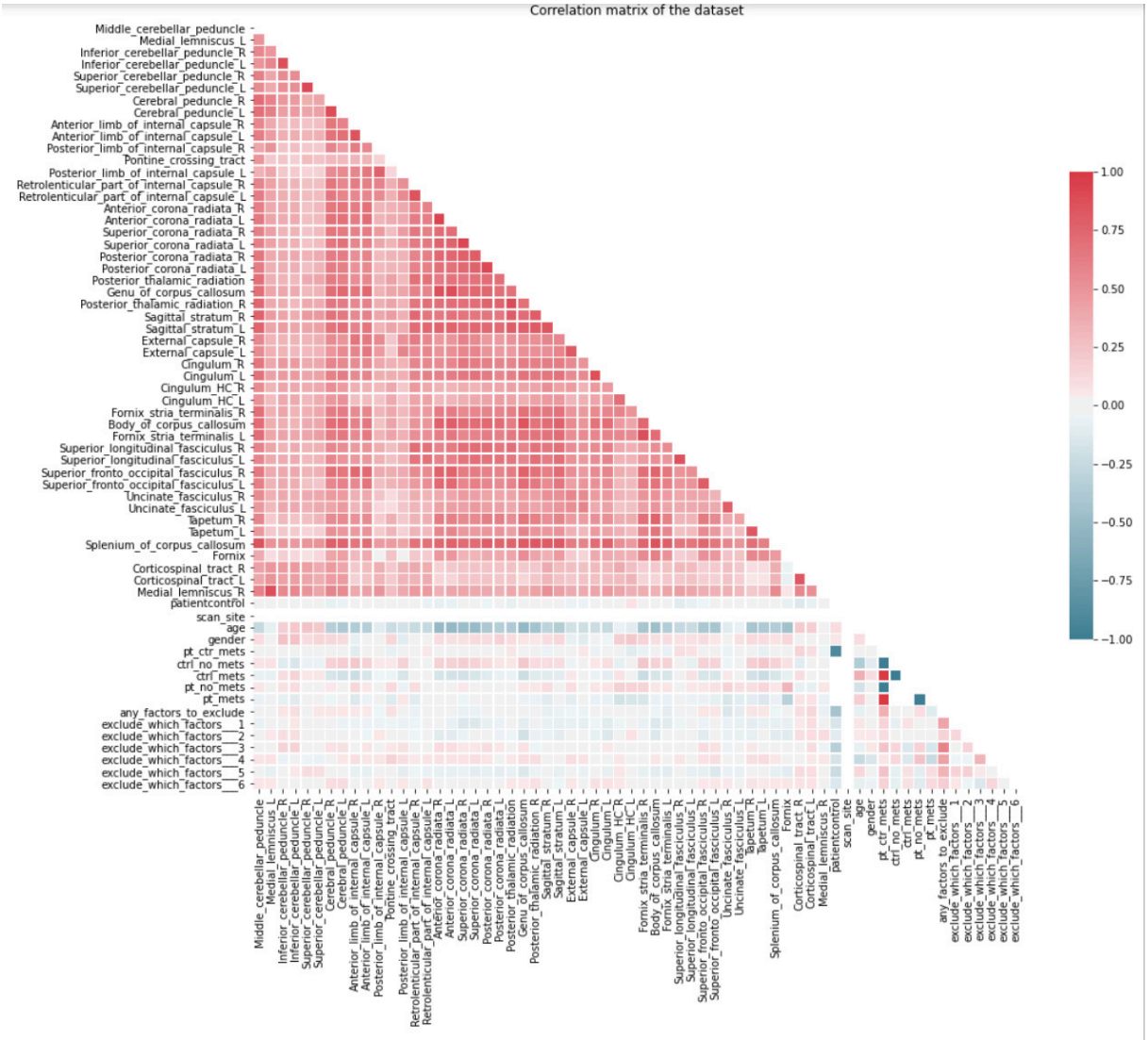

Figure 51 Correlation matric for all the features in DTI dataset

In Figure 52, 4 randomly chosen variables have been plotted and similarly, to the Shared Roots dataset distribution of the 4 variables form the DTI dataset are not completely normally distributed and that could cause anomalies to the classification.

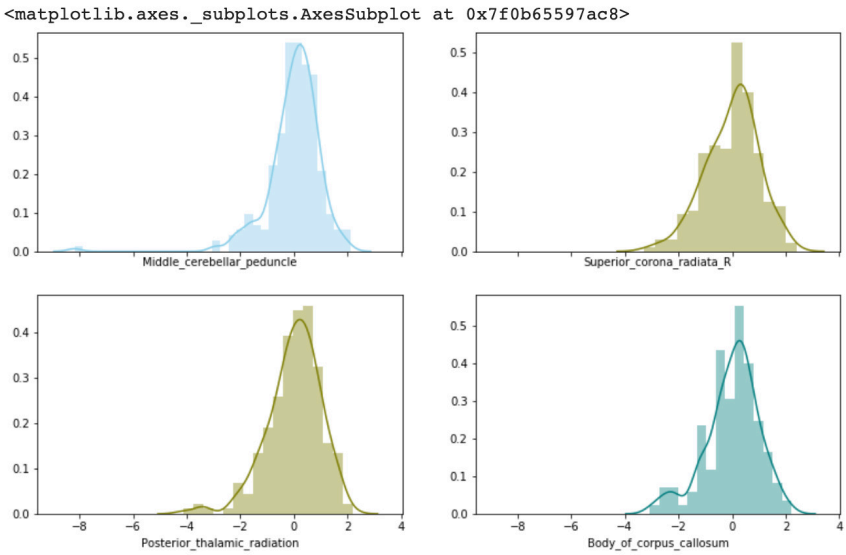

Figure 52 Distribution of randomly selected features form DTI dataset

The last part of the data exploration on the DTI dataset was to perform a t-test on the age difference between the PD patients and controls. Figure 53 shows that both PD patients' and controls' age was normally distributed and the observing difference between the means of the two groups was not significant. The small t-statistic value also implies the insignificant difference between the mean ages of the two groups

```
Sample mean age of Parkinsons patients:  63.681965026855465
Sample mean age of Parkinsons controls:  62.919623057047524
Difference:  0.76
P-Value: 0.7748443683869779
T-statistic: 0.2882647428595105
Is Patient and Control age difference significant?  [False]
Patients age is normal?  True
Controls age is normal?  True
```

Figure 53 T-test for age difference in DTI dataset

Disease Classification

Shared Roots

From the disease classification on the Shared roots dataset a lot of useful information can be derived. As it can be seen in Figure 54, the best performing algorithms are MLP and Naïve Bayes. On the other hand, it is clear that random forest is overfitting but its performance on train and nested cross validation does not seem to be affected. Naïve Bayes scored the highest in terms of test accuracy whereas MLP scored the highest nested cross validation score. Moving on to Figure 55, the highest precision and recall score is achieved by logistic regression, whilst random forest classifier scored the highest ROC AUC score. The dotted line

on both graphs represents the chance level of accuracy. The high recall score from all of the classifiers implies low false positive rate but the relatively low (to the recall) precision implies high false positive rate. In addition, the results imply that many of the predicted labels were not correct when compared to the training labels.

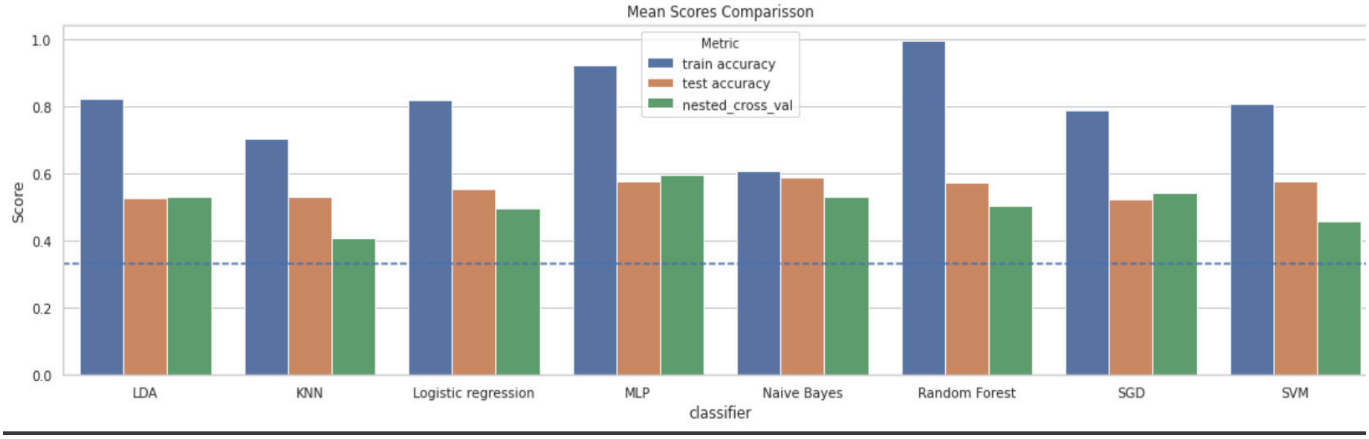

Figure 54 Mean Scores for all the classifiers for Shared Roots dataset disease classification(part 1)

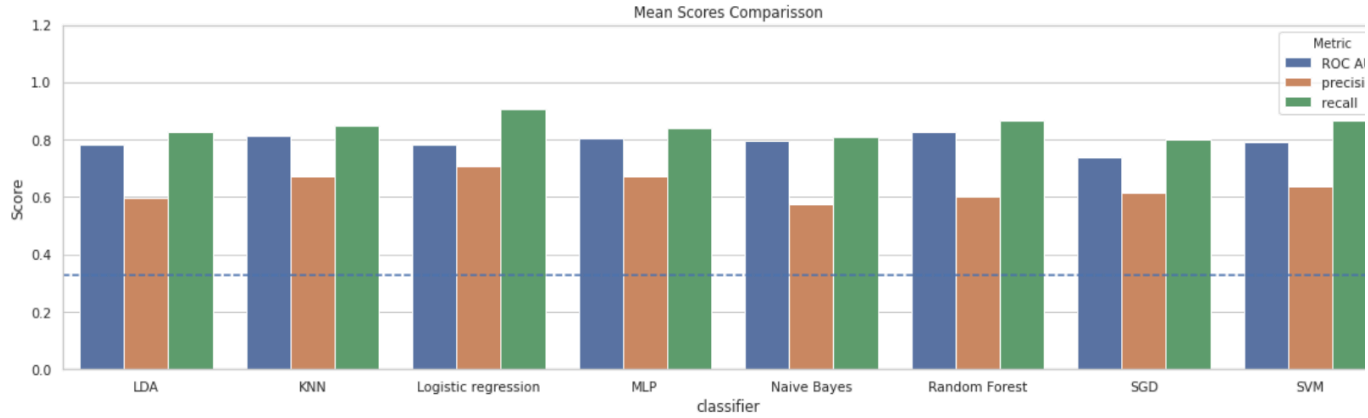

Figure 55 Mean Scores for all the classifiers for Shared Roots dataset disease classification (part 2)

Due to the fact that there are 6 different metrics that can measure the performance of the algorithms, another method was applied to identify the best classifier. The method was described in the approach section and the results obtained are shown in Figure 56. The graph shows that the best algorithms overall are MLP (0.736) and random forest (0.729) where all the classifiers have high scores way above the chance classification level.

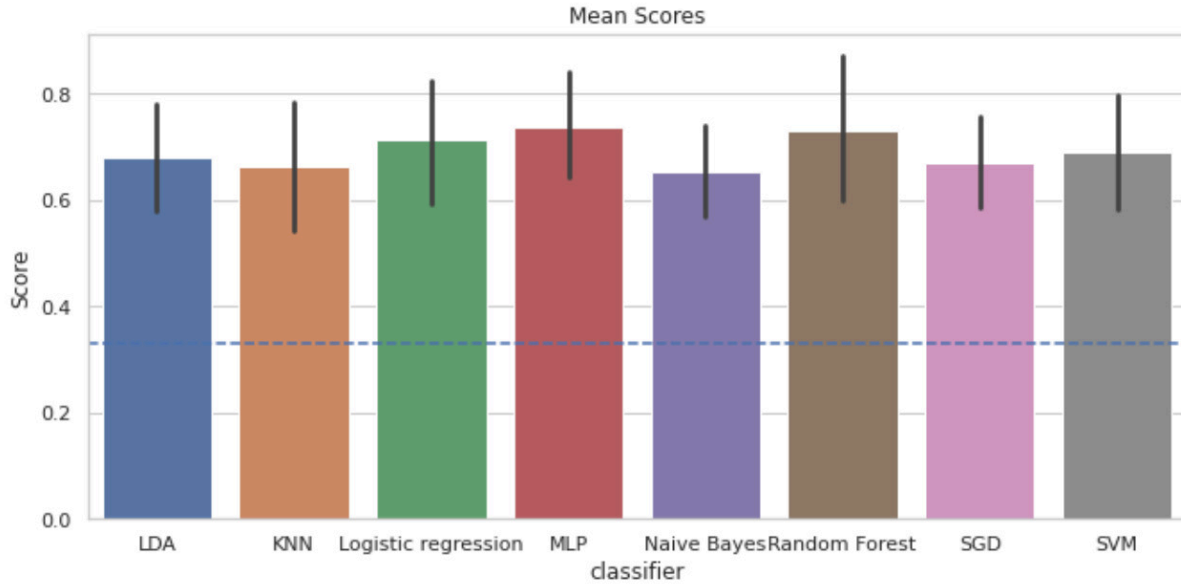

Figure 56 Average score for all the classifiers for disease classification on Shared Roots

Since all the classifiers have scored results from 0.65 to 0.74, a graph was used to show their performance on the different datasets (base dataset, hippocampus and dataset with selected features). As it can be seen from Figure 57, the first eight bars for each group represent one algorithm, whereas the seventh bar represents the average of all the other eight algorithms in that particular group. Overall, the hippocampus dataset had the worst results with only the random forest performing better on that subset. The overall performance stays the same when feature selection is applied, implying that the features that have been dropped from the base dataset did not help the decisions made by the algorithms. It can therefore be concluded that the feature selection methods have selected features that can be more effective in the classification of PD.

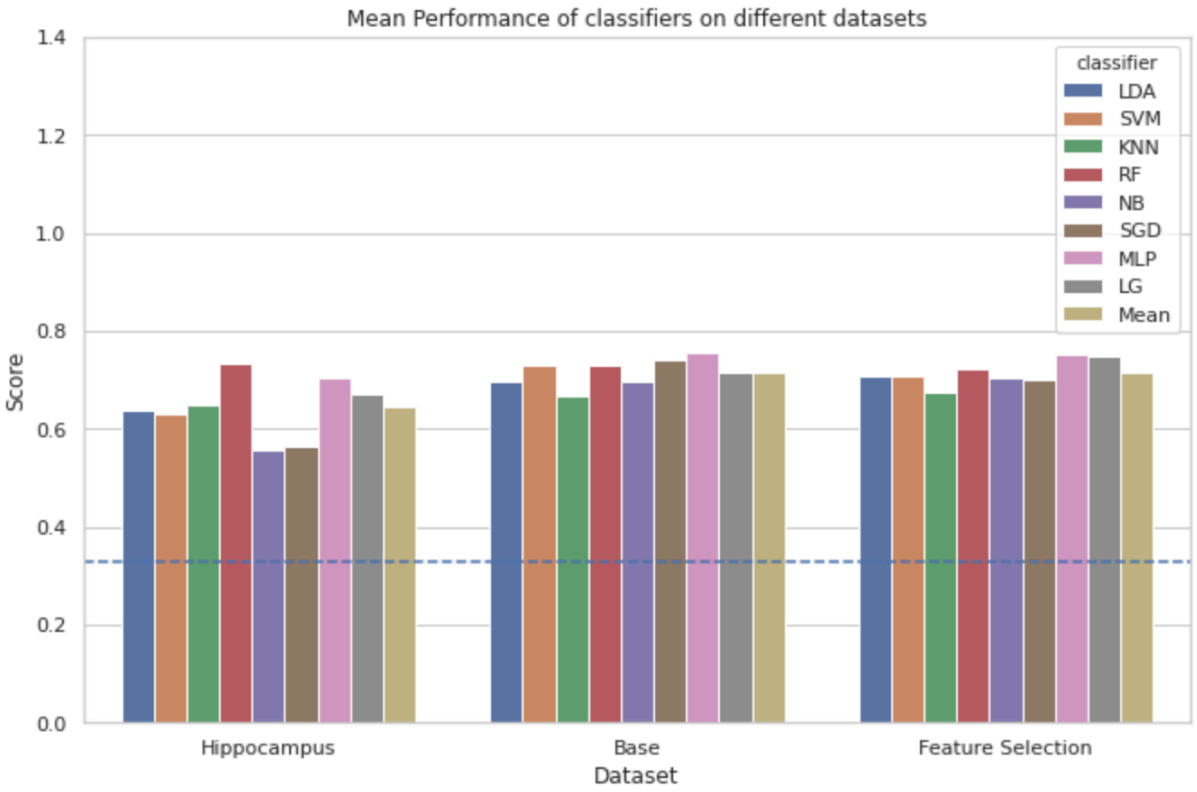

Figure 57 The performance per classifier on the different subsets of Shared Roots

DTI

Disease classification on the DTI dataset performed much better than the classification on Shared Roots. Despite the fact that there was much less data in the DTI dataset, the features that were used may have been more significant in classifying the diseases. As shown in Figures 58 and 59, the analytic average performance of each algorithm implies that in many occasions there was model overfitting as train accuracy was most of the times close to 1. MLP and logistic regression scored the highest test accuracy and nested cross validation score. Similarly, logistic regression scored the highest average ROC AUC and recall score, while random forest scored the highest average precision for the PD class. In contrast with the Shared Roots dataset results, the difference between precision and recall was not present in most occasions in the DTI dataset.

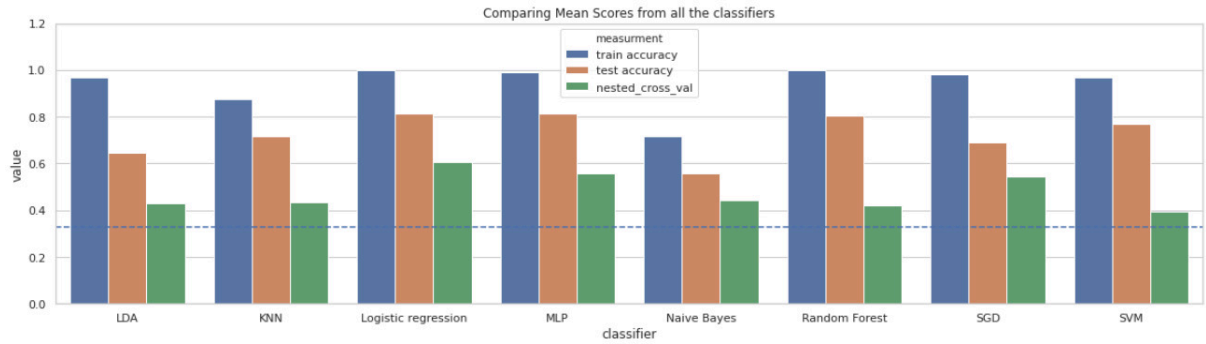

Figure 58 Mean performance results from all the classifiers on DTI dataset disease classification (part1)

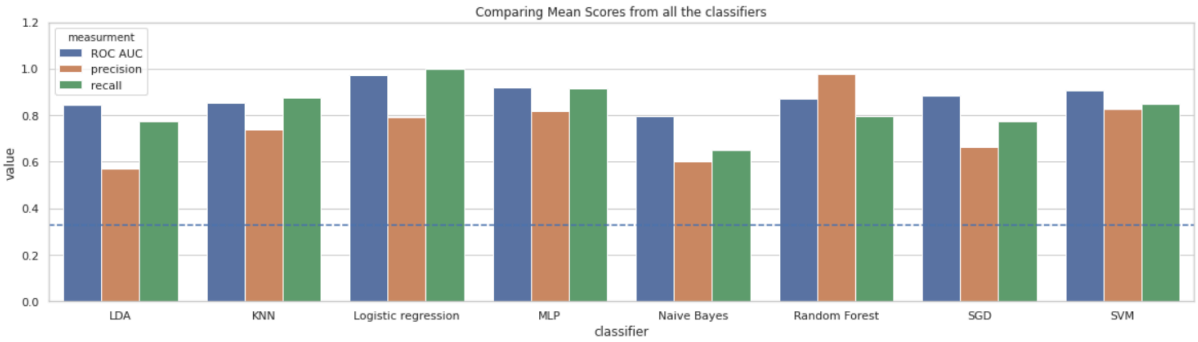

Figure 59 Mean performance results from all the classifiers on DTI dataset disease classification (part 2)

Figure 60 is used to summarize the results of the classifiers by plotting the average from all the metrics that are used to measure the performance of the classifiers. Logistic regression had the best overall performance (0.921) where random forest (0.882) and MLP (0.883) are again amongst the best classifiers. Bernoulli Naïve Bayes scored the worst results out of the 8 classifiers used similarly to the diseases classification on the Shared Roots dataset. Overall, diseases classification using the DTI dataset produced the best classification results from all the others; that could imply that much more useful information can be extracted from the DTI dataset.

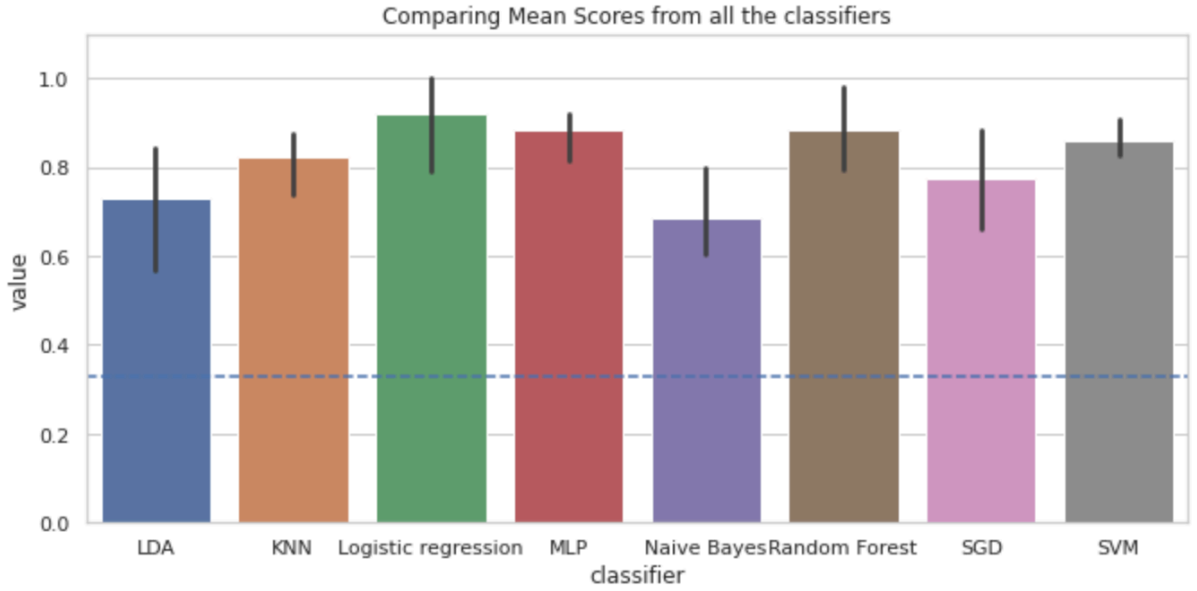

Figure 60 Mean overall results from disease classification on DTI dataset

Figure 61 shows the difference of the mean performance of classifiers over the different datasets. It can be concluded that the mean performance from all the classifiers (7th bar from each group) has been increased after feature selection. This implies that the features that were dropped from the base dataset were just adding noise to the data. LDA had the larger increase in performance after the feature selection while the performance of KNN dropped after feature selection.

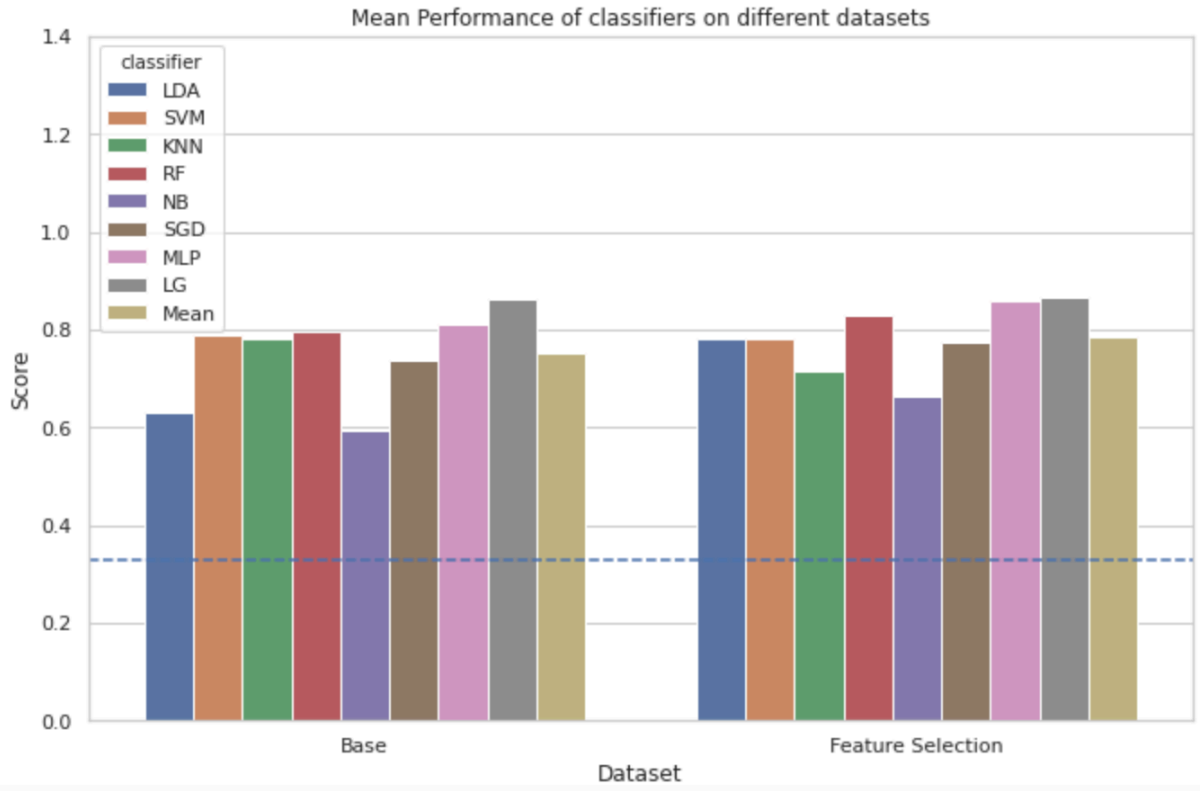

Figure 61 Average performance of the classifiers on different subsets of DTI dataset

Patient – Control Classification

Shared Roots

From the patient – control classification on Shared Roots dataset it has been observed that the feature selection methods improve the performance of the classifiers. The reason for this could be the fact that the first classification approach uses all patients and controls, omitting their disease status. Classifiers have performed very differently compared to the disease classification. As it can be seen from Figure 62, classifiers do not perform well enough when the classification involves all the 3 diseases (Base). In contrast, when only one disease (PD) is used to classify the patients from the controls the results are always better than the chance classification and the results are becoming even better when feature selection methods apply. The "Mean" bar indicates the average performance from all the classifiers for the particular classification dataset. Overall, algorithms perform much better when they are using the selected features so that implies that those features can differentiate the PD patients form controls much better than chance level. Despite that, the results of the classifiers were not as good as the disease classification from the same dataset. The reason for that could be the lack of data as the PD patients and controls were a third of the whole dataset whereas in the diseases classification half of the dataset was used (patients).

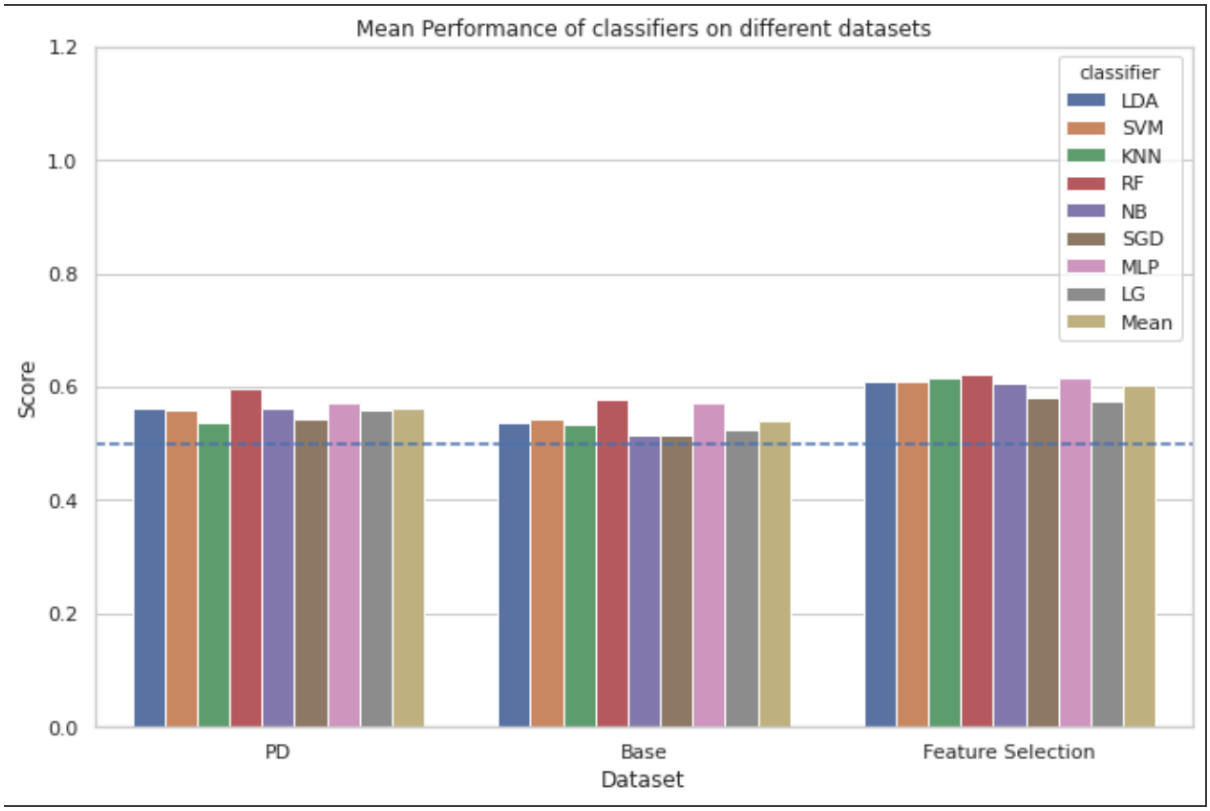

Figure 62 Mean performance of the classifiers on different sets for Shared Roots

Since feature selection is improving the performance of the classifiers the mean performances of all the classifiers after feature selection were plotted in Figure 63. As it can be seen, similarly to the disease classification, the best performing classifiers were MLP (0.614), random forest (0.622) and KNN (0.617) while the worst performing algorithm was logistic regression and SGD. It is important to state that while K-NN was the worst algorithm in disease classification, it has been proven that it works much better in patient control classification.

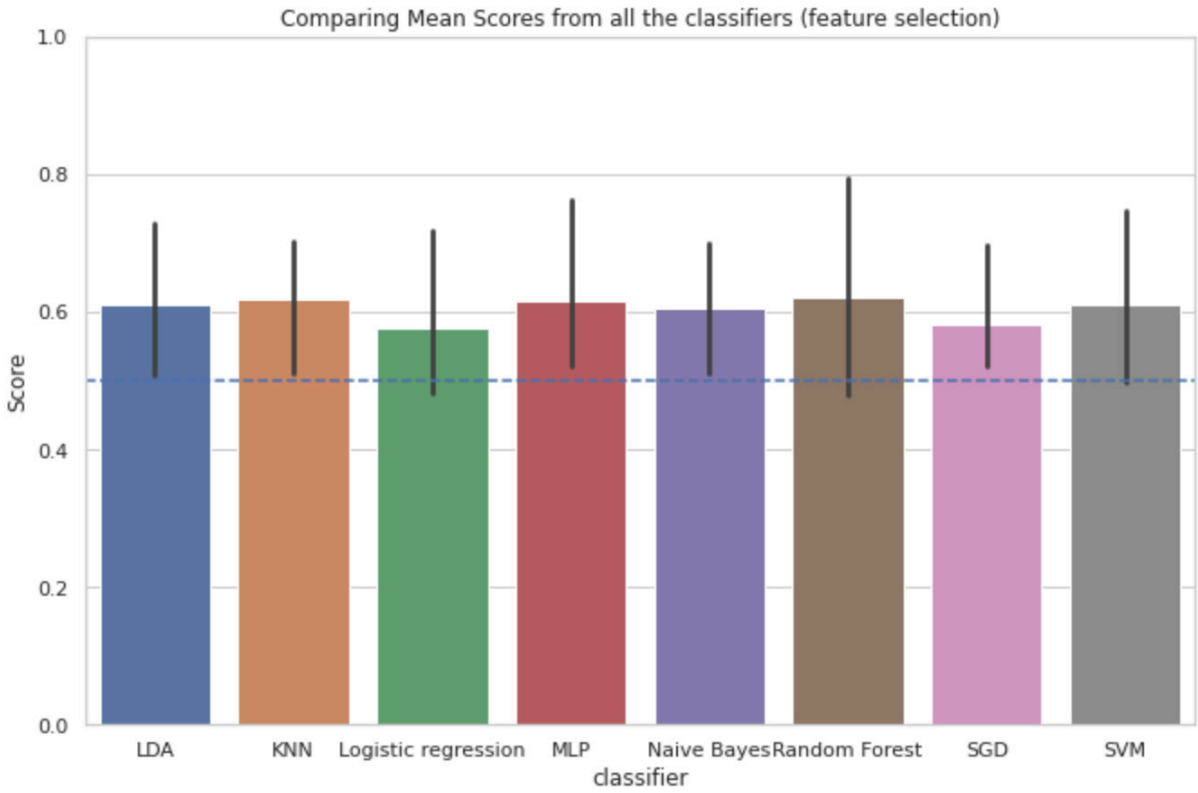

Figure 63 Mean performance of classifiers after feature selection on PD patient – control classification using Shared Roots

DTI dataset

The worst classification approach was the PD patient-control classification as the classification performance of all the classifiers was often below the chance classification level. Figure 64 shows that half of the algorithms were overfitted and the results on the test accuracy were the lowest compared to other classification approaches. Furthermore, the test accuracy was more than 0.5 on LDA and MLP only. Despite that, nested cross validation was relatively high with MLP and Naïve Bayes being the best performing algorithms on that metric.

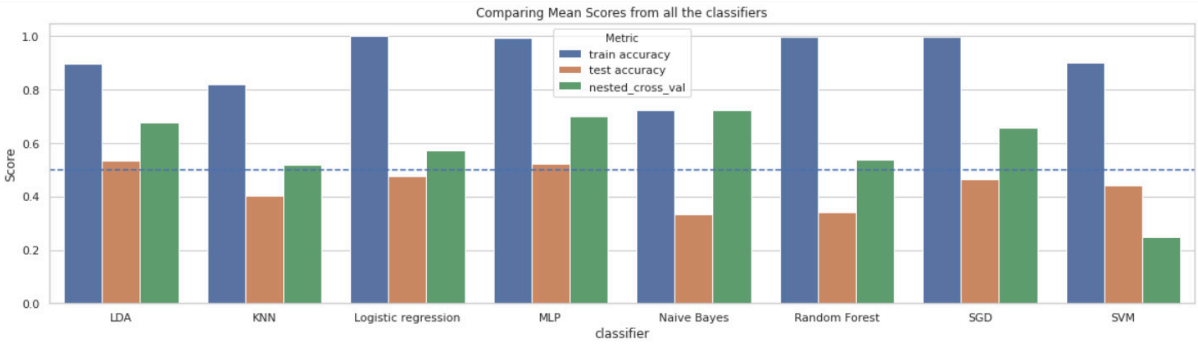

Figure 64 Average performance (3 metrics) for the PD patient - control classification for DTI dataset

The overall classification performance of all the classifiers is shown in Figure 65. As in the previous classifications, the MLP classifier performed better (0.630) than the rest whereas the LDA (0.620) was the second-best performing algorithm. However, 4 classifiers performed below the chance classification level. The reason for the poor classification results was the fact that there was not sufficient data for the models to train and so they overfitted. The high performance on the disease classification using the DTI dataset could imply that if more data become available for the particular dataset, the results for the patient – control classification could have been improved dramatically.

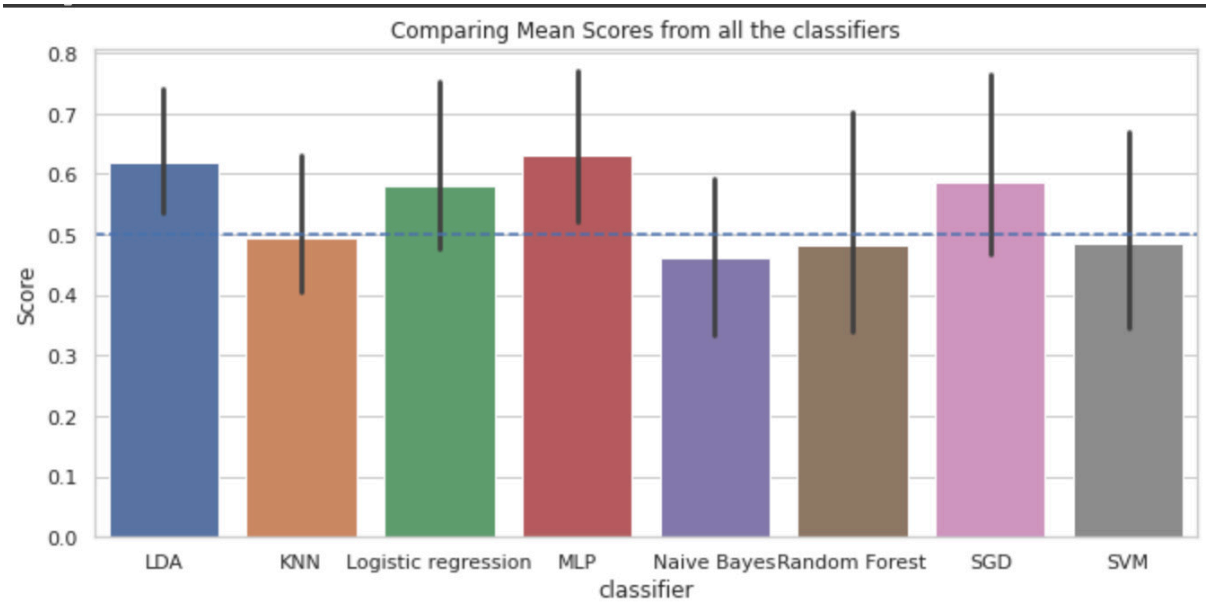

Figure 65 Mean performance of all the classifiers on PD patient - control classification

Last part of the analysis of the PD patients – controls classification was to plot the average performance of all the classifiers before and after the feature selection methods. Figure 66 shows that the mean performance of all the classifiers was decreased after the feature selection. The underlying reason could possibly be that there was not enough data for training and so the feature selection methods failed to identify the features that could improve the performance of the classifiers.

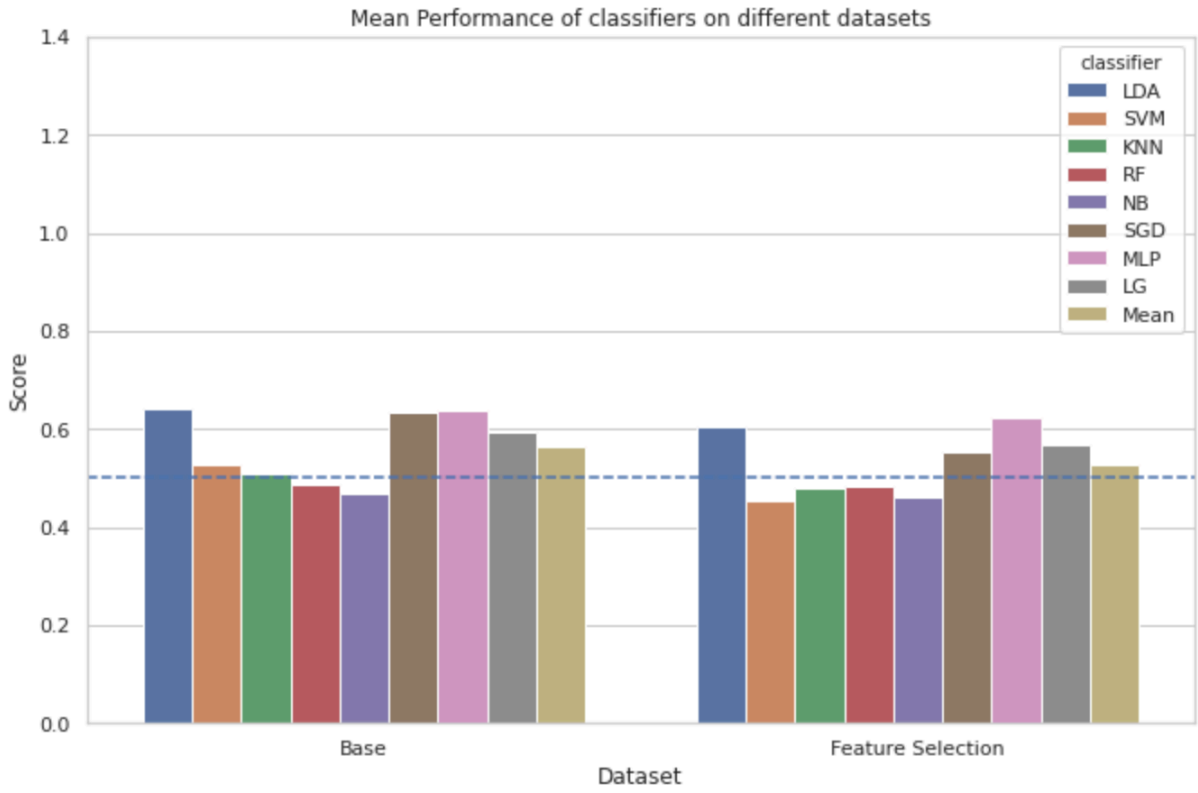

Figure 66 Mean performance of the classifiers with different features using DTI dataset

Diagnostic Features for PD

The study defined two different lists with the diagnostic features of PD. The features obtain form the Shared Roots dataset can be seen in Figure 68. As many studies state; "PD is a predominantly disorder of the basal ganglia". [36] Basal ganglia's largest nuclear complex is striatum which is composed of the caudate and putamen. One of the main functions of putamen is to regulate different movements; an action often impaired in PD. One of the features that this study identifies as a diagnostic feature is the left putamen. As shown in Figure 67, [37] the putamen is located above the amygdala, at the point where amygdala is connected to the basal ganglia. The left amygdala has been also identified to be one of the diagnostic features. Next feature found was the 3rd ventricle which is a fluid-filled cavity that carries cerebrospinal fluid. The 3rd ventricle expands around the amygdala and putamen as shown in Figure 67. The average values of the diagnostic features when comparing the PD patients and controls (Figure 69) come to prove that PD patients have extreme readings compared to the PD controls. A similar research by Liana G. Apostolova et al. , [38] suggests that there was "significant enlargement of all portions of the lateral ventricles". Likewise, to this outcome, this research project has concluded to the observation that there was significant enlargement of the 3rd ventricle which is part of the lateral ventricles as well as 2 more areas which are enclosed by the 3rd ventricle, the left putamen and amygdala.

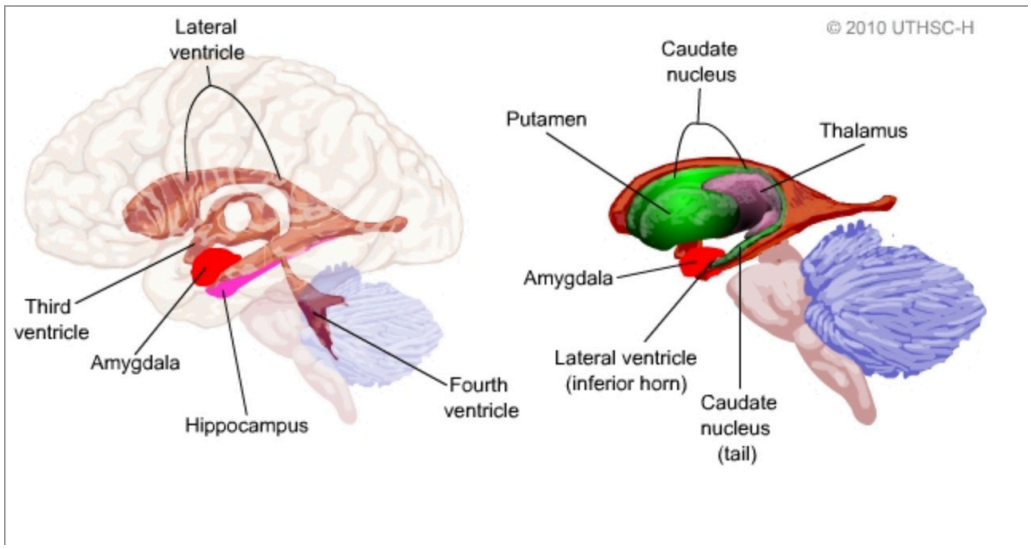

Figure 67 Relationship of amygdala to other brain structures [37]

	Feature
2	Right-VentralDC
6	3rd-Ventricle
13	Left-Putamen
15	Left-Amygdala

Figure 68 Diagnostic features of PD obtained from Shared Roots

```
Right-VentralDC   -0.812615
3rd-Ventricle      1.066055
Left-Putamen      -0.881546
Left-Amygdala     -0.950375
dtype: float64
```

```
Right-VentralDC   -0.258900
3rd-Ventricle      0.547853
Left-Putamen      -0.349212
Left-Amygdala     -0.277445
dtype: float64
```

Figure 69 The average values of the diagnostic features of PD for PD patients (left) and PD controls (right)

The same method applied in the DTI dataset and the features that were found are shown in Figure 70. The three major parts of the cerebellar peduncle were included (middle, superior and inferior). According to a paper by Andreea C. Bostan and Peter L. Strick [39]; that studies the relationship between the cerebellum and basal ganglia, concluded that those two are interconnected at the subcortical level meaning that the features discovered from the Shared Roots dataset are related with the features found from the DTI dataset. Furthermore, corpus callosum, is used for connecting the two parts of the brain (left and right cerebral hemispheres) and two of its four parts (body and genu) have been identified as diagnostic features. In Figure 71, corpus callosum is shown to be connected with the basal ganglia which is strongly-associated feature in PD. Finally, anterior corona radiata in Figure 72 appears to be connected to both body and genu of corpus callosum. Corona radiata is a bundle of nerve

cells that are used to transfer information for both motor and sensory nerve pathways, which are also affected by PD [40]

	Feature
2	Pontine_crossing_tract
4	Body_of_corpus_callosum
5	Anterior_corona_radiata_R
7	Superior_cerebellar_peduncle_R
10	Middle_cerebellar_peduncle
11	Inferior_cerebellar_peduncle_L
12	Genu_of_corpus_callosum

Figure 70 Diagnostic features for PD extracted from DTI dataset

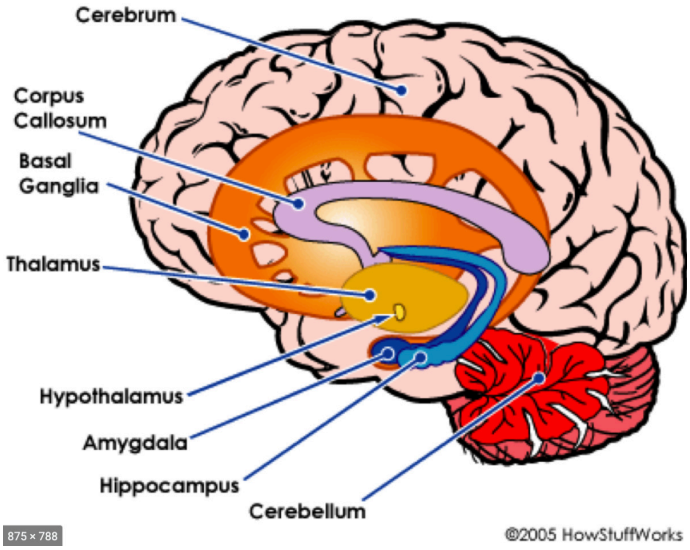

Figure 71 Corpus Callosum connection with Basal Ganglia

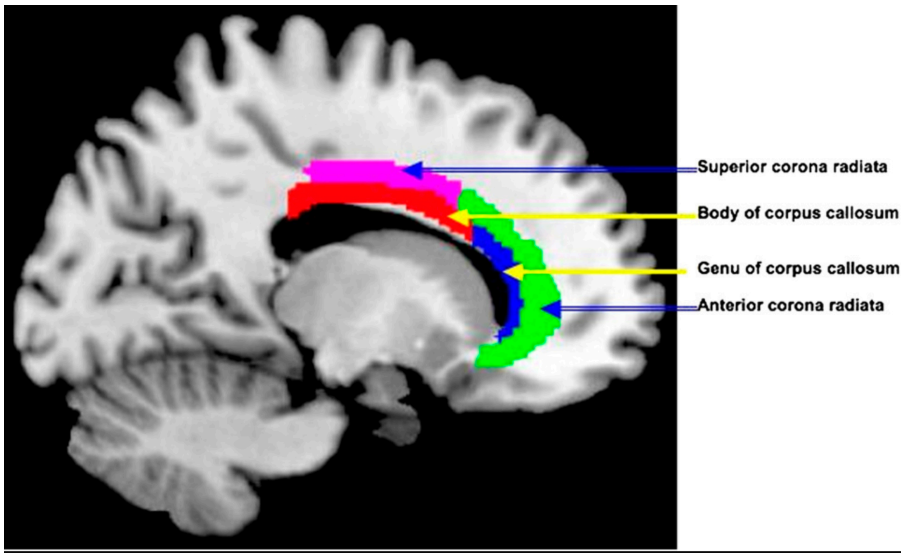

Figure 72 Connection of anterior corona radiata with corpus callosum [41]

Key Results

Summarizing the results obtained from the 4 different classification methods, key information can be derived. Firstly, in 3 out of the 4 methods the feature selection improved the performance of the algorithms. Since the feature selection methods have improved the performance in most occasions, the diagnostic features stated above are strongly related with the occurrence of PD in the clinical population. Furthermore, it has previously been described in the scientific literature that the parts identified as the diagnostic features are related with the occurrence of PD.

The primary aim of the thesis was to achieve a comprehensive comparison between the classifiers. It is clear that some classifiers perform better than others on all the datasets tested. In the Shared Roots dataset, the best performing classifier overall was the MLP (average score 0.675) and then random forest (average score 0.676) whereas in the DTI dataset the best performing classifier was again the MLP (average score 0.757) and then logistic regression (average score 0.750). Overall, the classifiers were much more successful in classifying the diseases rather than classifying the patients from the controls. The best classification performance occurred at the disease classification for the DTI dataset. The results also imply that more data is needed for the DTI dataset as it shows promising potentials due to the high accuracy in diseases classification despite the reduced data volume compared to Shared Roots. The collective results' comparison (Figure 73) allows for some general conclusions to be made:

- MLP is the best performing classifier overall with an average score of 0.716
- Logistic regression is the best of the classical machine learning algorithms used with an average score of 0.697
- Random forest is the third best performing algorithm with an overall average score of 0.678
- Bernoulli Naïve Bayes was the worst performing classifier with an average score of 0.601
- Neural networks seem to be performing better in the particular datasets but since only one classifier of that type was tested (MLP), further investigation needs to be made.

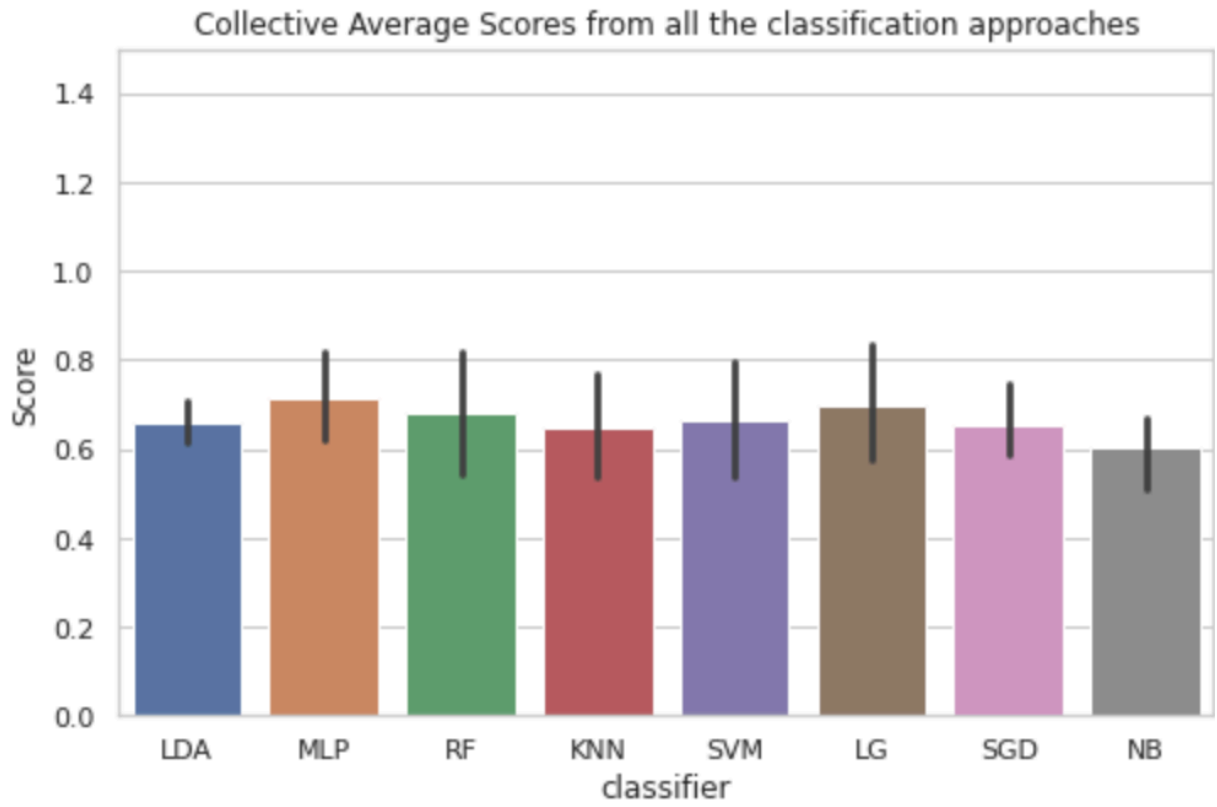

Figure 73 Collective average scores from all the classifiers from all the 4 classification approaches

Result Evaluation

Disease Classification over Patient – control classification

While analysing the results from the classifications, it was clear that disease classification performed much better than the patient – control classification despite the lower chance – level classification. The major reason was the lack of data as the disease classification dealt with half of the population (patients) of the dataset while the patient – control classification dealt with less than a third of the population (PD patients and controls). Furthermore, another possible reason for the less accurate results of the patient – control classification was that the age difference between the PD patients and PD controls in Shared Roots dataset was significant as their p-value was less than 0.05. The combination of those two factors contributed to the classification of patient – control performing worse; that could possibly be eliminated if there was more data available to train the model.

The best performing algorithm overall

One of the reasons explaining the superiority of MLP over the rest of the classifiers is that it does not require the data to be normally distributed. Additionally, neural networks work better than classical machine learning classifiers as they can better model heteroscedasticity (data has non-constant variance and has high volatility).

The best performing classical machine learning algorithm

As shown in Figure 73, logistic regression was ranked as the second-best classification algorithm. This is probably because it performs better when the features used are not too

correlated to each other (values < 0.9). Generally, logistic regression is considered to be one of the best algorithms for classification.

The third best performing algorithm

Random forest outperforms every other classifier on 2 of the 4 classification approaches while it appears to be second best on the DTI dataset disease classification. Random forest appears to perform better than the rest on those occasions as it can cope with non-linear solutions, similarly to the MLP. In addition, high dimensionality in the data gives an advantage to random forest as no dimensionality reduction is required.

The worst performing algorithm

The reasons behind the failure of the BNB algorithm to outperform the rest of the classifiers could vary. First of all, BNB makes the assumption that all the features are independent between them but as the correlation matrix has proven, most of the features tested are correlated between them with values around 0.7. Furthermore, another reason could be the data scarcity as in some occasions (PD patient – control classification) there was sufficient data for training and that could have as a result the probability of certain features to be highly weighted towards 1 or 0.

Methodology Evaluation

Evaluation of methods

Overall, the methods applied in the research were based on testing each classifier multiple times using different parameters in order to allow a fair comparison between the algorithms. In the results' section different bar plots show the performance of classifiers, enabling the comprehensive comparison and thus allowing conclusions to be drawn about the best performing classifier tested. Despite that, the methods applied in the study could be altered in a way to make it more reliable. One of the most important weaknesses of the methodology was that the feature selection procedure didn't involve repeat measurements. Each time the feature selection procedure run, a new set of selected features was defined and the old one was replaced.

A small change in the methodology of selecting the features could result in a more reliable conclusion to be made as the current solution is based on a single execution where the results of the classifications are based on multiple runs of the system. In addition, the models that were used for feature selection on each of the 4 classification approaches vary. The decision upon which models were used as estimators relied on the performance of the particular model on the given data. That means that features were not selected based on the exact same process and this may imply that the feature selection process was biased from that perspective.

Furthermore, the decision on which classifiers tested could be done differently as not all the classification algorithms that were used are able to provide feature importance feedback. By choosing algorithms that can give feature importance the feature selection methods would be more accurate.

The approach that has been followed and could have been done differently was the methodology for the patient – control classification (Shared Roots dataset). Initially, a patient – control classification occurred without separating the participants into diseases resulting in models with very poor results close to the chance classification level. Since it was known from the PCA that the patients and controls from all the diseases together could not be clustered then that classification approach did not need to be tested.

Evaluation of metrics used

In the particular study, 6 different metrics were used to measure the performance of each classifier. Those metrics allowed the comprehensive comparison to be made as each one of them gave valuable information about the behaviour of each individual model. The different metrics helped to create an unbiased test for all the classifiers so no algorithm is benefited from biased readings from a single metric.

Train and test accuracy

Firstly, the simplest and most common metrics that were used are the train and test accuracy. They both help to get an overall idea of what are the true predictions achieved by the model on different subsets of the data. Their score helped to summarise the overall performance of the model but it sometimes led to false conclusions to be made. Despite their simplicity and valuable readings, those metrics could give misleading results of the performance of the model if the dataset is imbalanced. The reason is that the correct predictions made are not separated into classes so if the dataset is imbalanced and the model does not perform well on classifying the imbalanced class, the train and test accuracy will not be able to identify that.

Precision and Recall

Other metrics that were used are precision and recall; they were used in order to eliminate the drawbacks of the train – test accuracy as they can calculate the accuracy in which the model predicts the value of each class separately. Those two metrics were two of the most important as they showed the ability of a particular model to distinguish between the PD patients and the rest of the classes. Their disadvantage is that they do not use in their readings all the available information from the confusion matrix as TN are not taken into consideration. Due to the fact that the study was not focused on the value of negative class, the particular drawback of precision and recall did not affect the conclusions.

ROC AUC score

ROC AUC score was used in order to show the area under the curve of the true positive rate against the false positive rate. The basic use of this metric is to tell the ability of the model to distinguish the data points between the classes. It was a vital reading that helped to enable the deeper understanding of the performance of the models. ROC AUC score could be avoided in the readings as it is better to be used when the research question focuses equally on both true positives and true negative classes but in this particular occasion, the project was aiming to separate the PD patients from the rest of the classes.

Nested Cross Validation score

The last metric included in the readings was the nested cross validation score which was used in order to train models which would undergo hyperparameter tuning. Its main use was that

it can calculate the generalization error (prediction of unseen data points) which gave further insights on the performance of each model.

However, Nested CV was used in all the occasions were model selection occurred after grid search and in those cases the same data was used for both tuning and evaluation of the models. That was probably the reason why in the PD patient – control classification the nested CV score was much higher compared to the rest of the metrics - as the information that was used may have overfitted the data. In order to avoid that from happening during the model training and evaluation, the inner loop occurred during the grid search by using multiple train – test splits. The model's score was maximised by fitting the model to each of the different training sets and was subsequently maximized even more in the hyperparameter selection that occurred at the validation set. The outer loop occurred using the "corss_val_score" and it estimated the generalization error by calculating the mean of the multiple test set scores.

Future Work

While the aims of the study have been met successfully, there is still a lot of potential for further development that could improve the quality of the research. Some of the most important improvements involve more classification approaches to be followed in order to draw better conclusions on which algorithms perform better and also perform further analysis upon the identification of the diagnostic features of PD.

Use hippocampus features for further classification

Despite the fact that classification using the hippocampus features was tested for the disease classification on Shared Roots, it was not tested as expected on the patient – control classification due to the poor results obtained on the first test. Examining the classification performance on the patient – control classification, could provide further useful information. First of all, it would provide more runs for each of the algorithms and thus gain valuable data in order to evaluate them with different subsets of the data. If the classification was proven to be better than previous attempts, then the hippocampus could provide more meaningful information about the diagnostic features of PD and hence allow better conclusions to be made.

Altering the feature selection procedure

As discussed in the methodology evaluation section, the feature selection procedure contained a 'bug'. In order to ensure the validity of the selected features, repeats of each of the feature selection methods should occur on every run of the system in order to eliminate any anomalous decisions. That could be the reason that in one of the classification approaches, the feature selection classification failed to perform better than the base one. Repeats in the feature selection methodology will ensure that the obtained features are actually the ones that are more important in the decisions taken by the algorithms.

Increase the classification algorithms tested

In this project, 8 different classification algorithms were extensively tested under different datasets in order to achieve a comprehensive comparison. Despite that, the system is already set and many more algorithms could easily be added and therefore allow a greater variety of algorithms to be tested. That will enhance our understanding on how the classification algorithms work on pre-processed information from MRI images and therefore allow the better identification of the diagnostic features for more NPDs. One of the most promising algorithms that could be tested in future development is linear regression as it is an algorithm that can provide feature significance.

Research on the PTSD and SC

In both datasets used there were 3 different diseases but the research was focused on finding the diagnostic features for one of them. A more detailed study on all of the 3 diseases will allow the better separation of the diagnostic features for each of the diseases and hence allow the discovery of features that can be related to the occurrence of more than one disease. Furthermore, as the presence of each disease in the data showed, there were far more PTSD participants in the clinical population and therefore greater volume of data for training and

evaluation could result in far more accurate results. Finally, the classification of the patient – control for the other 2 diseases could give more information on which algorithm performs better on the given data and therefore prove or disprove the results that this study has concluded.

Use a greater variety of metrics

The more metrics in the research, the more complete view an individual can draw about the performance of a classification algorithm as it provides a different viewpoint. Some of the metrics that could be included in future work for the particular study are logarithmic loss, F1 score and Mean absolute error.

Logarithmic Loss

Since the disease classification is a multiclass problem then logarithmic loss would be suitable. It can be considered a reliable metric as it penalizes every false prediction made by the classifier based on how much it varies from the actual loss. Logarithmic loss could take values from 0 to ∞ with scores closer to 0 being the most desirable ones. Scores near to 0 indicate high accuracy of the model and its formula is shown in Figure 74. y_{ij} indicates if sample i belongs to class j and p_{ij} indicates the probability that sample i belongs to class j. N represents the number of samples and M the number of different classes. [42]

$$LogarithmicLoss = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} * \log(p_{ij})$$

Figure 74 Logarithmic loss formula [41]

F1 score

F1 score is calculated (Figure 75) using the scores of precision and recall so it represents the ability of the classifier to distinguish a particular class. It can take values from 0 to 1 and the greater the score the more accurate the model is. Most importantly, the F1 score can show the weighted average between precision and recall as that it takes both FP and FN into account. [42]

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

Figure 75 F1 score formula [41]

Mean Absolute Error (MAE)

The last important metric that could be included into future work of the particular study is the Mean Absolute Error. As the name implies, it is used to measure the error between predicted and original values. Their difference is added up and then they are divided with the number of metrics in order to conclude the mean value. The drawback of this metric is that it

cannot identify which classes the classifier has failed to classify in comparison to the other classes.

Conclusions

Though this report, a large volume of knowledge has been extracted from the particular datasets. Useful information was provided on how supervised machine learning algorithms behave on tabular preprocessed MRI data. Based on the performance of those algorithms and by implementing a variety of feature selection methods, some of the possible diagnostic features were identified in this study.

The first aim of the project was to achieve a comprehensive comparison between the 8 classification algorithms tested. After exhaustive testing and a multitude of different models created for each of the classifiers, the conclusions that were derived are significant and can define with accuracy the best performing algorithms. According to the results' analysis, the best performing classifier overall was the MLP, then logistic regression and the third best was the random forest classifier. MLP performed better than the rest of the algorithms due to the fact that it's a neural network and doesn't follow the classical machine learning approach and, it does not require the features' distribution to be completely normal. Logistic regression performed almost as good as the MLP and proves why it is considered one of the best classical machine learning classification algorithms as the simplicity of its algorithm led to excellent results. Additionally, random forest classifier which preformed almost as high as logistic regression, yielded good results due to the fact that the data used had high dimensionality that gave an advantage to the particular algorithm. Contrary, the worst performing algorithm was the Bernoulli Naïve Bayes algorithm possibly due to data scarcity, causing algorithm to weight some features' probability close to 0 or 1 and therefore obtain the same value each time a feature was tested. Generally, most of the classifiers, even BNB performed above the chance classification performance on almost every occasion; proving the validity of the outcomes that this study suggests. The fact that each of the classifiers was tested over different datasets and under different hyperparameters on every occasion helps to give a spherical view of the real capabilities of each classifier tested.

The secondary aim of the project was to compare different feature selection methods in order to derive which are the diagnostic features of PD. In total four different models were compared and each method 'voted' for the features that were most important. The system used that information to select the features with the most 'votes' in order to perform the classification with selected features on each occasion. In total, 11 different features were identified as the diagnostic ones for PD from both datasets analysed in this study. The results yielded appear to have high validity as most of the identified features have a strong relation with the basal ganglia; the brain area affected in PD. The features that were identified by this study can help scientists to explore the pathophysiology of PD even further and hence prove or disprove the results of this study.

Despite the promising results of the study, there were areas of the project that could have been implemented better in order to increase the accuracy of the results. Changes to the algorithm that is used to find the diagnostic features should have taken place in order to allow repeat measurements. Additionally, in order to improve the quality of the conclusions, more metrics could have been taken in order to gain valuable data on the performance of each individual model trained. Finally, the hippocampus features were not tested into classifying the PD patients from the PD controls; a classification approach that could conclude some

valuable information about the role of the hippocampus in the diagnosis of PD. On the other hand, the feature selection methods did not choose any hippocampus-related feature; this could mean that the hippocampus does not carry useful information for the identification of PD.

The most challenging aspect of the study was to build a system that will enable a fair experiment to be made and hence achieve a comprehensive comparison between the involved classification algorithms. Furthermore, another challenging part was deciding how to collect the results of each classifier and subsequently analyse them in a meaningful manner such that they provide a clear image of the performance of the classifiers. Fortunately, the majority of those challenges were overcame and the results can identify with validity which is the best performing algorithm on the particular data and hence identify the diagnostic features of PD.

Reflection on Learning

From the beginning of this module, I ensured that I remained focused and goal-driven to achieve the timely completion of my scientific report. This project attracted my attention as it was related to machine learning, a topic which although I haven't had the change to study in depth before, I really wanted to explore further. I am a person who is driven by new challenges as I feel that they give me the chance to develop and educate myself. Another major reason for choosing a data science related project was my curiosity to explore this field of computer science before deciding if it is something that I want to pursue further at higher level. This experience helped me understand that it is one of my preferred topics in computer science and hence I have applied for a Data Science postgraduate degree.

During the semester, I have always tried to push myself towards improve my analysis as much as possible by working consistently and methodically in order to avoid having pressure towards the end of the semester. The organised working plan helped me to maximise my productivity as I am the type of person who keeps on working until there is nothing else to be done. Unfortunately, the unpreceded events secondary to the Covid-19 pandemic, including the lockdown, had a negative impact on the development of the solution. After my repatriation to Cyprus, I was enforced in a compulsory quarantine in a hotel for two weeks. During that time, the hotel's Wi-Fi wasn't working properly and on top of that I was unable to exit my room for any reason as part of the quarantine process. I tried to minimise the negative impact of quarantine by focusing on my work, but as one can expect the circumstances had a negative effect on my mind-set.

The timeline that was created in the initial plan helped me to stay on track. There were times though that I had to compromise a lot of my free time in order to keep up with the expectations of the project. One of the most useful skills that I have developed during the development of this project was task prioritization and time-management. When the workload started increasing, I had to prioritise the tasks in order to allow the flow of the project to continue smoothly. In addition, time-management and self-motivation was really important as the whole project relied on a single person. Finally, I have improved my self-discipline while ensure that such big project was completed within the time set. Due to that combination it was easy to ignore the first weeks of the semester but I ensured that I worked consistently throughout the semester rather than rushing to finish work close to the deadline. This allowed me some valuable time towards the end of the semester to develop the report with a clear mind without having the time pressure.

To conclude, in my opinion, the most valuable asset that I have personally gained from this research is that I have extended my knowledge in a field that intrigues me so much to study. Most importantly, I have enjoyed and learned a lot from this study and I am sure I will apply the knowledge and experience that I have gained in the near future.

Appendices

Appendix A

The hippocampus

Similar to a computer's CPU, our brain is responsible for almost every action performed by our bodies. The three main areas of the brain are the cerebrum, brainstem and cerebellum. The brain is divided into two hemispheres and the corpus callosum is used to connect them. The hippocampus is located in the cerebrum and it has an important role in learning and memory. Research suggests that damage to this brain structure is related to different neurological and psychiatric disorders. [43]

Appendix B

Within – class and between – class scatter matrices calculation

The within class scatter matrix Sw is calculated using $Sw = \sum_{i=1}^{c} Si$ where $Sw = \sum_{x \in Di}^{c} (x - m_i)(x - m_i)^T$. mi is the mean vector such that $m_i = \frac{1}{ni} \sum_{x \in Di}^{n} xk$.

The between class scatter matrix S_B is calculated using $\sum_{i=1}^{c} N_i (m_i - m)(m_i - m)^T$ where m is the overall mean and m_i is and N_i are the sample mean and size of the respective classes. [17]

Appendix C

Parameter grids used for each of the classifiers

LDA parameter grid:

```
#hyperparameters
 lda_param_grid = {'n_components':[0,1,2,3,4,5,6],
                   'shrinkage':[0.2,0.4,0.6,0.8],
                   'solver':['svd', 'lsqr', 'eigen']}
```

Random forest parameter grid:

```
# Random forest Parameter Grid
n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]
max_features = ['auto', 'sqrt']
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)
min_samples_split = [2, 5, 10]
min_samples_leaf = [1, 2, 4]
bootstrap = [True, False]
```

K-Nearest Neighbors parameter grid:

```
leaf_size = list(range(1,50))
n_neighbors = list(range(1,15))
p=[1,2]
weights = ['uniform', 'distance']
algorithm = ['auto', 'ball_tree', 'kd_tree', 'brute']
```

Support Vector Machines – Support:

```
 svm_param_grid = {'C': [0.1,1, 10, 100],
                   'gamma': [1,0.1,0.01,0.001],
                   'kernel': ['rbf', 'poly', 'sigmoid', 'linear'],
                   'shrinking':[True, False],
                   'max_iter':[1000,100,10,5],
                   'probability':[True, False],

                   }
```

Bernoulli Naïve Bayes parameter grid:

```
#hyperparameters to be tuned
nb_param_grid = {'alpha':[0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1],
                 'binarize':[0.0,0.2,0.4,0.6],
                 'fit_prior':[True, False]
                 }
```

Logistic Regression parameter grid:

```
# grid search on logistic regression
lg_param_grid = {
    'C': [1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1e0],
    'penalty': ['l2', 'l1', 'elasticnet', 'none'],
    'n_jobs': [-1],
    'solver':['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']
}
```

SGD parameter grid:

```
sgd_param_grid = {
    'alpha': [1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3],
    'loss': ['log', 'modified_huber', 'squared_hinge'],
    'penalty': ['l2', 'l1', 'elasticnet'],
    'n_jobs': [-1],
    'max_iter':[10000, 1000, 100, 10, 5]
}
```

Multilayer Perception parameter grid:

```
mlp_param_grid = {
    'hidden_layer_sizes': [(50,50,50), (50,100,50), (100,)],
    'activation': ['tanh', 'relu'],
    'solver': ['sgd','lbfgs'],
    'alpha': [0.0001, 0.001, 0.1, 1],
    'learning_rate': ['constant','adaptive'],
}
```

References

[1] B. Voytek, 20 05 2013. [Online]. Available: https://www.nature.com/scitable/blog/brain-metrics/are_there_really_as_many/. [Accessed 8 4 2020].

[2] A. Bhande, "What is underfitting and overfitting in machine learning and how to deal with it.," 11 03 2018. [Online]. Available: https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76. [Accessed 11 4 2020].

[3] C. K. Firoz. et al., "An overview on the correlation of neurological disorders with cardiovascular disease," 1 2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4281592/. [Accessed 9 4 2020].

[4] O. Cohen, "Data-science? Agile? Cycles? My method for managing data-science projects in the Hi-tech industry.," 3 1 2019. [Online]. Available: https://towardsdatascience.com/data-science-agile-cycles-my-method-for-managing-data-science-projects-in-the-hi-tech-industry-b289e8a72818. [Accessed 9 4 2020].

[5] Johns Hopkins Medicine, "Can Environmental Toxins Cause Parkinson's Disease?," [Online]. Available: https://www.hopkinsmedicine.org/health/conditions-and-diseases/parkinsons-disease/can-environmental-toxins-cause-parkinson-disease.

[6] Collaborators, GBD 2016 Parkinson's Disease, "Global, regional, and national burden of Parkinson's disease, 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016," 1 10 2018. [Online]. Available: https://www.thelancet.com/journals/laneur/article/PIIS1474-4422(18)30295-3/fulltext#%20. [Accessed 09 04 2020].

[7] Chien Tai Hong et al., "Prevalent cerebrovascular and cardiovascular disease in people with Parkinson’s disease: a meta-analysis," 4 9 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6130276/. [Accessed 5 5 2020].

[8] Synopsys Editorial Team, "Top 4 software development methodologies," 28 3 2017. [Online]. Available: https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/. [Accessed 9 4 2020].

[9] D. L. Hermoye, "Diffusion Tensor Imaging (DTI) - Fiber Tracking," [Online]. Available: https://www.imagilys.com/diffusion-tensor-imaging-dti/. [Accessed 8 5 2020].

[10] Wikipedia, "White matter," 12 4 2020. [Online]. Available: https://en.wikipedia.org/wiki/White_matter. [Accessed 6 5 2020].

[11] Wikipedia, "Machine learning," 9 4 2020. [Online]. Available: https://en.wikipedia.org/wiki/Machine_learning. [Accessed 10 4 2020].

[12] Wikipedia, "Supervised learning," 8 4 2020. [Online]. Available: https://en.wikipedia.org/wiki/Supervised_learning. [Accessed 11 4 2020].

[13] W. Kenton, "Overfitting," 2 7 2019. [Online]. Available: https://www.investopedia.com/terms/o/overfitting.asp. [Accessed 11 4 2020].

[14] Wikipedia, "Overfitting," 31 3 2020. [Online]. Available: https://en.wikipedia.org/wiki/Overfitting. [Accessed 11 4 2020].

[15] Wikipedia, "Eigenvalues and eigenvectors," 29 4 2020. [Online]. Available: https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors. [Accessed 4 5 2020].

[16] A. Violante, "An Introduction to t-SNE with Python Example," 29 8 2018. [Online]. Available: https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1. [Accessed 5 5 2020].

[17] S. Raschka, "Linear Discriminant Analysis – Bit by Bit," 3 8 2014. [Online]. Available: https://sebastianraschka.com/Articles/2014_python_lda.html. [Accessed 12 4 2020].

[18] A. Navlani, "Understanding Random Forests Classifiers in Python," 16 5 2018. [Online]. Available: https://www.datacamp.com/community/tutorials/random-forests-classifier-python. [Accessed 12 4 2020].

[19] Wikipedia, "k-nearest neighbors algorithm," 5 4 2020. [Online]. Available: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm. [Accessed 12 4 2020].

[20] R. Gandhi, "Naive Bayes Classifier," 5 5 2018. [Online]. Available: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c. [Accessed 12 4 2020].

[21] R. Gandhi, "Support Vector Machine — Introduction to Machine Learning Algorithms," 7 6 2018. [Online]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47. [Accessed 12 4 2020].

[22] A. Pant, "Introduction to Logistic Regression," 22 1 2019. [Online]. Available: https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148. [Accessed 12 4 2020].

[23] M. Joshi, 4 4 2019. [Online]. Available: https://medium.com/@maithilijoshi6/a-comparison-between-linear-and-logistic-regression-8aea40867e2d. [Accessed 6 5 2020].

[24] A. V. Srinivasan, "Stochastic Gradient Descent — Clearly Explained !!," 7 9 2019. [Online]. Available: https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31. [Accessed 12 4 2020].

[25] Wikipedia, "Multilayer perceptron," 26 10 2019. [Online]. Available: https://en.wikipedia.org/wiki/Multilayer_perceptron. [Accessed 13 4 2020].

[26] J. M. Ashfaque, 20 6 2016. [Online]. Available: https://www.researchgate.net/figure/Diagram-of-k-fold-cross-validation-with-k-10-Image-from-Karl-Rosaen-Log_fig1_332370436. [Accessed 5 5 2020].

[27] Geeks for Geeks, "Confusion Matrix in Machine Learning," [Online]. Available: https://www.geeksforgeeks.org/confusion-matrix-machine-learning/. [Accessed 5 5 2020].

[28] A. Sharma, "Confusion Matrix in Machine Learning," [Online]. Available: https://www.geeksforgeeks.org/confusion-matrix-machine-learning/.

[29] K. McNulty, "Trash or treasure — how to tell if a classification algorithm is any good," 7 9 2018. [Online]. Available: https://towardsdatascience.com/trash-or-treasure-how-to-tell-if-a-classification-algorithm-is-any-good-cb491180b7a6. [Accessed 14 4 2020].

[30] V. Zhou, "A Simple Explanation of Gini Impurity," 29 3 2019. [Online]. Available: https://victorzhou.com/blog/gini-impurity/. [Accessed 13 4 2020].

[31] W. Koehrsen, "An Implementation and Explanation of the Random Forest in Python," 30 8 2018. [Online]. Available: https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76. [Accessed 13 4 2020].

[32] Delloitte, "Understanding a Random Forest Model through Feature Importance," [Online]. Available: https://www2.deloitte.com/nl/nl/pages/innovatie/artikelen/understanding-a-random-forest-model-through-feature-importance.html. [Accessed 10 5 2020].

[33] R. Agarwal, "The 5 Feature Selection Algorithms every Data Scientist should know," 27 7 2019. [Online]. Available: https://towardsdatascience.com/the-5-feature-selection-algorithms-every-data-scientist-need-to-know-3a6b566efd2. [Accessed 13 4 2020].

[34] F. Valeria, "Feature Selection using LASSO," 30 3 2017. [Online]. Available: https://beta.vu.nl/nl/Images/werkstuk-fonti_tcm235-836234.pdf. [Accessed 13 4 2020].

[35] The SciPy community, "What is NumPy?," 26 7 2019. [Online]. Available: https://docs.scipy.org/doc/numpy/user/whatisnumpy.html. [Accessed 14 4 2020].

[36] Robert A Hauser, "Parkinson Disease," 29 4 2020. [Online]. Available: https://emedicine.medscape.com/article/1831191-overview#a3. [Accessed 2 5 2020].

[37] A. Wright, "Chapter 6: Limbic System: Amygdala," [Online]. Available: https://nba.uth.tmc.edu/neuroscience/m/s4/chapter06.html. [Accessed 2 5 2020].

[38] Liana. G. Apostolova. et al., "Hippocampal and ventricular changes in Parkinson's disease mild cognitive impairment," 1 7 2014. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4077346/. [Accessed 3 5 2020].

[39] P. L. S. Andreea C. Bostan1, "The basal ganglia and the cerebellum: nodes in an integrated network," 1 6 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6503669/. [Accessed 2 5 2020].

[40] J. Vega, "The Corona Radiata and Stroke," 6 2 2020. [Online]. Available: https://www.verywellhealth.com/what-is-the-corona-radiata-3146130. [Accessed 2 5 2020].

[41] Yi-Yuan Tang et al., "Mindfulness meditation improves emotion regulation and reduces drug abuse," 1 6 2016. [Online]. Available: https://www.researchgate.net/publication/303799121_Mindfulness_meditation_improves_emotion_regulation_and_reduces_drug_abuse. [Accessed 5 5 2020].

[42] A. Mishra, "Metrics to Evaluate your Machine Learning Algorithm," 24 2 2018. [Online]. Available: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234. [Accessed 3 5 2020].

[43] Kuljeet Singh Anand and Vikas Dhikav, "Hippocampus in health and disease: An overview," 1 10 2012. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3548359/. [Accessed 5 5 2020].

[44] C. G. Goetz, "The History of Parkinson's Disease: Early Clinical Descriptions and Neurological Therapies," 09 2011. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3234454/.

[45] Wikipedia, "History of Parkinson's disease," 04 2020. [Online]. Available: https://en.wikipedia.org/wiki/History_of_Parkinson%27s_disease#cite_note-pmid15568171-3. [Accessed 09 04 2020].

[46] Mayo Clinic Staff, "Parkinson's disease," 30 06 2018. [Online]. Available: https://www.mayoclinic.org/diseases-conditions/parkinsons-disease/symptoms-causes/syc-20376055. [Accessed 09 04 2020].