National College of Ireland

Developing a Web Application with built in Predictive Models to perform real time predictions of Parkinson's disease

MSc Research Project
Data Analytics

Achinthya Ganesh
x17125600

School of Computing
National College of Ireland

Supervisor: Mr.Vikas Tomer

National College of Ireland
Project Submission Sheet – 2017/2018
School of Computing

Student Name:	Achinthya Ganesh
Student ID:	x17125600
Programme:	Data Analytics
Year:	2018
Module:	MSc Research Project
Lecturer:	Mr.Vikas Tomer
Submission Due Date:	13/08/2018
Project Title:	Developing a Web Application with built in Predictive Models to perform real time predictions of Parkinson's disease
Word Count:	XXX

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	
Date:	17th September 2018

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Developing a Web Application with built in Predictive Models to perform real time predictions of Parkinson's disease

Achinthya Ganesh
x17125600
M.Sc. Research Project in Data Analytics

17th September 2018

Abstract

Parkinsons disease is one of the most common progressive neurodegenerative diseases known to be highly fatal and incurable. It is one of the diseases which directly affects the human nervous system in turn affecting all the motor and non-motor functions of a human. Considering the fact that there is still no means to treat or eradicate the disease completely, a temporary solution can be bought in a way by making early predictions of the disease possible. This research focuses on developing a predictive web application which will perform the predictions of the disease based on the diseases most common symptoms and deliver the results to the user. The machine learning models used in this research are K-NN, logistic regression, support vector machine and decision tree out of which the model with the highest performance will be chosen to develop the web application. The built predictive model is published as a web service in Azure ML Studio which generates the API key enabling the model to be extended and be able to run in the backend of a web application. The predictive models are evaluated on the test data and decision tree stands out as the best performing model with 98% accuracy and 96% sensitivity. Thus, this model will be embedded within the web application that will perform real time predictions of the disease. This is being developed to provide an effective means for the user to predict the disease as early as possible so that necessary treatment measures can be taken.

Keywords : Parkinson's disease, Azure ML, machine learning, web application.

1 Introduction

Parkinsons is a disease which affects the entire central nervous system of a human leading to gradual deterioration of the motor functions of the person. There are many reasons as to why a person develops this disorder and above all, there is no cure for the disease Perlmutter (2009). The main aim of the research is to develop an interactive web application that will perform early predictions of the disease with the use of machine learning techniques. A web application is a client server software program that uses browsers and other web technologies to transmit and perform data operations. The most common

symptoms of the disease are researched and analysed based on which the occurrence of the disease is predicted. The required data is aggregated, cleaned and transformed after which four predictive models namely K-NN, logistic regression, SVM and decision tree are built. The model with the best performance metrics will be chosen with the help of a ROC curve after which the application will be built. Based on the symptoms, the models that are built will return the predictions either yes or no corresponding to the occurrence of the disease in the future. As much as its important to predict the disease accurately, it is even more important to deploy the same to the user from which they get can use out of. This has been identified as one of the major research gaps and has been explained in detail in the literature review section. Every other approach has considered and implemented techniques only to predict the disease but there has been no effort taken to deploy the same to the user. Thus, this gap in research will be analysed and provided a solution here in the proposed approach by developing a web application to perform the predictions which will permit the user to access it from anywhere. This has been the key motivation behind the implementation of this project. The research question the project intends to answer is:

How can we develop a simple interactive web application embedded with built in predictive models that will perform real time predictions of Parkinsons disease based on the diseases symptoms such as speech impairment, leg agility and tremors?
Another important aspect to be considered here is the user interface of the web application. It has to be as simple as possible embedded with questions regarding the symptoms that will enable the user answer those with ease. Based on those answers, the prediction will be done after which the results are presented to the user. The study covers in detail the related work, methodology, implementation, evaluation, conclusion and any possible future works of this study in the forthcoming sections.

2 Literature Review

As mentioned above, the various techniques and methodologies that will be implemented in this approach are discussed and justified with supporting evidence by analysing the existing theories as proposed by various authors. It is further broken down into several sections with each one analysing existing theories corresponding to the topic stated.

2.1 Parkinson's disease prediction

There has been quite a considerable amount of research done in this field where several techniques and methodologies have been implemented to predict the disease efficiently. The existing works are thoroughly analysed which will identify the research gaps and justify the use of appropriate techniques and methods that will be implemented in the proposed approach. This section which comprises of work entirely related to Parkinsons disease is again broken into different subsections based on the techniques and methods used.

2.1.1 Prediction based on symptoms

This section focuses on analysing the existing work done on the prediction of the disease by recording various parameters relating to the symptoms of the disease such as eye movement recording, speech and voice recognition, limb movements, tremor analysis etc.

Śledzianowski et al. (2018) states that the disease can be predicted by recording the eye ball movements of a person. Szymański et al. (2017) supports Śledzianowski et al. (2018) by stating the same. While Śledzianowski et al. (2018) used existing recording of the eye ball movements to predict the disease Szymański et al. (2017) used instruments such as saccadometer and Eye Tribe to record the eye movements. They have used decision tree to perform the prediction of the disease achieving an accuracy of 93.3 and 85 percentage respectively which proves that Śledzianowski et al. (2018) has given good results when compared to Szymański et al. (2017) even though the methodology used are the same.
Another approach proposed by Aich, Choi, Park and Kim (2017) follows the same structure as followed by the approaches stated above with only one difference of recording the limb movements instead of eye ball movements of the patients which is also called as gait analysis. It proposes that by measuring the limb movements and changes, the disease can be predicted. Mazilu et al. (2013) and Pun et al. (2016) contribute to the work of Aich, Choi, Park and Kim (2017) by providing supporting evidence and proving that analysing the limb movements otherwise called as freezing of gait with the help of machine learning techniques can turn out to be effective. While Aich, Choi, Park and Kim (2017) and Mazilu et al. (2013) have used decision trees and regression analysis respectively to produce the results, Pun et al. (2016) visualizes the impact of the symptom over the disease. Similarly, Bhat et al. (2017) has performed prediction of the disease based on tremor analysis which refers to the involuntary muscle movements or shivering of the subjective body parts. It strongly states that tremor is one of the major symptom found to be very common among all patients with Parkinsons disease Bhat et al. (2017).
One of the most common symptom of Parkinsons disease is speech impairment Perlmutter (2009). The patient diagnosed with the disease either faces difficulties talking to others or finds themselves talking things out of their control i.e. not mentally aware of what they are talking about. Thus, analysing these in a person can help detect the disease effectively as it is said that they are generally common in all Parkinsons disease patients. Again, Chandrayan et al. (2017) claims speech analysis to be an effective tool in identification of the disease and has followed an approach of detecting the disease based on voice measurements using Support vector machines (SVM). Similarly, Agarwal et al. (2016) has followed the same approach of detecting the disease using voice measurements but with an innovative algorithm that is almost similar to neural networks called Extreme Learning Machine (ELM). Both the models have performed well with an accuracy of over 85 percent and has produced good results also.
Thus, to sum up, these approaches have used one of the above mentioned symptom each to carry out their analysis. It has been done with the help of several machine learning algorithms which have been discussed above. But, it cannot be concluded that the person has Parkinsons disease based on analysing only one symptom as it can be a sign of any other neurodegenerative diseases remotely related to Parkinsons such as Alzheimers disease etc. Even these diseases exhibit similar symptoms but some of them are curable unlike Parkinsons and this might result in wrong diagnosis of the disease. Considering this drawback, all the major symptoms which are common among most of the Parkinsons patients are considered here instead of performing the analysis with just one symptom. The major symptoms of the disease are said to be tremors, rigidity, speech impairment, intellectual impairment, difficulty in swallowing Perlmutter (2009). Thus, after much research these has been considered and will be carefully analysed in order to predict the disease.

2.2 Feature selection

Feature selection is an approach followed to increase the accuracy of a model by only having the important predictive feature variables of the given data Xiao and Zhang (2009). Aich, Sain, Park, Choi and Kim (2017) emphasizes on the fact that implementing feature selection can increase the performance of a model. They have implemented an innovative approach called Recursive Feature Elimination (RFE) algorithm on patients voice data where only the important predictive voice features out of a whole lot are selected based on the variance. After feature selection, linear classifiers are built to get the prediction results. Similarly, Tejeswinee et al. (2017) has performed feature selection on a data which has over thousand gene features collected to predict Parkinsons disease and even Chandrayan et al. (2017) has performed feature selection on their speech data before carrying out their analysis. It can be seen that most of the approaches have done feature selection in the motive of increasing the accuracy of the model. But it is also necessary to note that the feature variables in the data that has been used by them is vast. Hence, it is safe to conclude that only when there is a huge set of feature variables, feature selection is necessary. The proposed approach only carries up to ten important symptoms of the disease which after careful research has been finalized. Therefore, based upon the above evidences and research, it can be concluded that feature selection is not necessary for the current approach.

2.3 Use of multiple machine learning techniques

There is no one way as to decide the right machine learning model for a given problem. There are a number of conditions and prerequisites like data sample size, correlation level, number of classes etc that needs to be satisfied in order to get good accuracies and it sort of differs for a numerical type prediction problem and a categorical prediction problem. Yadav et al. (2012) encourages the building of multiple machine learning models and use the one with the most accuracy and out of three different models namely Support Vector Machine, Decision tree and Logistic regression. SVM has supposedly performed well because it was a binary class problem. Similarly, Challa et al. (2016) focuses on predicting the disease based on the non-motor symptoms of the disease. It has considered four algorithms namely Multilayer perceptron, Bayesian networks, Random Forest and Boosted Logistic Regression (BLR) out of which BLR has performed well compared to the others standing out with 97 percent accuracy because it was a voice recording data with multiple features. The approaches that have binary class problems majorly follow K-NN, SVM and Logistic Regression Charleonnan et al. (2016). Thus, it is important to note that the choice of the model to be implemented depends on several factors and henceforth it is safe to build multiple models out of which the one with the best accuracy metrics can be chosen. The proposed approach will consider implementing K-NN, Logistic Regression, Decision tree and Support vector machine as this is a binary classification problem and these models are proven to be good with problems of such kind Charleonnan et al. (2016). While logistic regression is typically meant for binomial classification problems, SVM uses the hyperplane to separate two classes hence proving to be effective for binary classification problems. Thus, these models will be considered to develop the web application.

2.4 Ensemble learning

As discussed in the previous section, multiple machine learning algorithms will be implemented in the proposed approach. Ensembling is a process of combining the results of two or more models and producing a single final output. It is an attempt made to increase the accuracy of the models. Bashir et al. (2016) has followed an innovative approach where multiple heterogenous classifiers are built and ensembled. This approach has been carried out to predict several diseases including Parkinsons, breast cancer, heart disease, diabetes, hepatitis etc. Bashir et al. (2016) believes that combining multiple classifiers can bring out better results and increase the performance of the models given the individual classifiers of the models have a significant level of disagreement in their error rate. Fayyazifar and Samadiani (2017) has performed ensembling methods such as bagging and Adaboost to predict Parkinsons disease and have achieved an accuracy of about 96 percent approximately. Similarly, Zhang et al. (2017) has argues that building an ensemble of multiple classical regression models has helped them capture the risk factors of Parkinsons disease more efficiently. But again, the data that has been used consists of lots of feature variables which are highly uncorrelated with each other. On the other hand, Kuncheva and Whitaker (2003) contradicts this by saying that the performance and the accuracy will not always increase with the use of ensembles. It totally depends on the data used. Thus, it is pretty clear that there are some constraints which needs to be satisfied for this to produce good results. As mentioned earlier, the proposed approach will focus on developing a predictive web application which will take the prediction results from the machine learning model built. The issue here is that the variables used in the model built must match the actual variables needed in the web application. If an ensemble classifier is built here, it will only have the combined results of all the models or the weighted average of the base classifier models but not the actual variables that have been used to perform the prediction in the first place. So, the web application will not work if an ensemble model is built. Moreover, it is not a compulsion to build an ensemble model if the accuracy achieved with one of the individual models is good enough to perform the predictions. The next section highlights the importance of developing web applications and why it needs to be implemented in the proposed approach.

2.5 Predictive web applications

One of the areas where a huge research gap has been found is the reach of these applications or models to the end user. So many authors have done lots of research in this field particularly for the prediction of Parkinsons disease but there has been no effort taken to have this easily accessible by the end user. There are only applications developed to monitor patients after the onset of the disease. Patel et al. (2010) and Memedi et al. (2011) have developed applications to monitor patient's activities diagnosed with the disease. The main purpose of this research is to help the user predict if he/she will have the disease or not in the future but the existing models that have been developed are raw and they need to have some kind of a simple running interface where its easily accessible to the user. That is why, developing a web application that can be directly accessed by users is encouraged. This has been implemented in several fields particularly in the medicinal field where it is most used. Alves et al. (2018) and Przednowek et al. (2018) have followed an approach of designing a web application for the prediction of chemical toxicity and designing an expert system for race training programs respectively. Alves et al. (2018) has implemented the Flash framework to develop the app which is

suitable for python language coding while Przednowek et al. (2018) has used the Shiny package in R to implement the same. While the proposed approach follows R programming, using Flash framework might not be the best choice here and the Shiny package is found to be suitable for data visualizations more than building predictive models Seal and Wild (2016). Uwagbole et al. (2017) advocates the use of Azure Machine Learning Studio where powerful and simple predictive models are built and deployed. The process is simple and user friendly which advocates the use of most of the machine learning algorithms. Above all, it also allows the model to be extended and published as a web service after which an Application Programming Interface (API) gets generated allowing the model to be used in any platform. This has been implemented by Uwagbole et al. (2017) to predict SQL Injection attack. Given the purpose of the proposed approach, it would be suitable and efficient to develop a web application using Azure ML to predict the disease.
The methods and techniques previously implemented has been discussed. Even though several methods to predict the disease has been implemented, there has been no effort taken to deploy the same to the user which is considered as a huge research gap since the main motivation behind these attempts to benefit the user is still unsolved. The goal is to develop a web application that is simple with a user friendly interface which can benefit the users and its not about creating attractive visualizations or graphs which the user might not even understand. This is being developed for a social cause and the only motivation behind this it to benefit the users who might incur the disease later in the future. These drawbacks have been understood and studied thoroughly and will be given a suitable solution through the implementation of the proposed approach.

3 Methodology

This section covers the technique and methods that are needed to implement the project. The basic aim of the project is to develop a predictive web application that will perform real time predictions of Parkinsons disease. A predictive model is created which is then extended and published as a web service which can then be used in a request response web application to perform the predictions. There are a few prerequisites that needs to be satisfied for this research to be counted as a valuable contribution:

- The data must consist of all the important symptoms as explained and justified as the most common symptoms of Parkinsons disease in the literature review section.
- The predictive model to be built will have to boast an extremely high accuracy with a good specificity rate since wrongly classifying negative classes for this particular rate can be dangerous.
- The predictive web application must have a very simple and user friendly UI since it has to be beneficial for people who might not have much knowledge about these technologies and applications.

The process of data acquisition and the usage along with the technologies and methods implemented to execute the project are explained in the forthcoming sections.

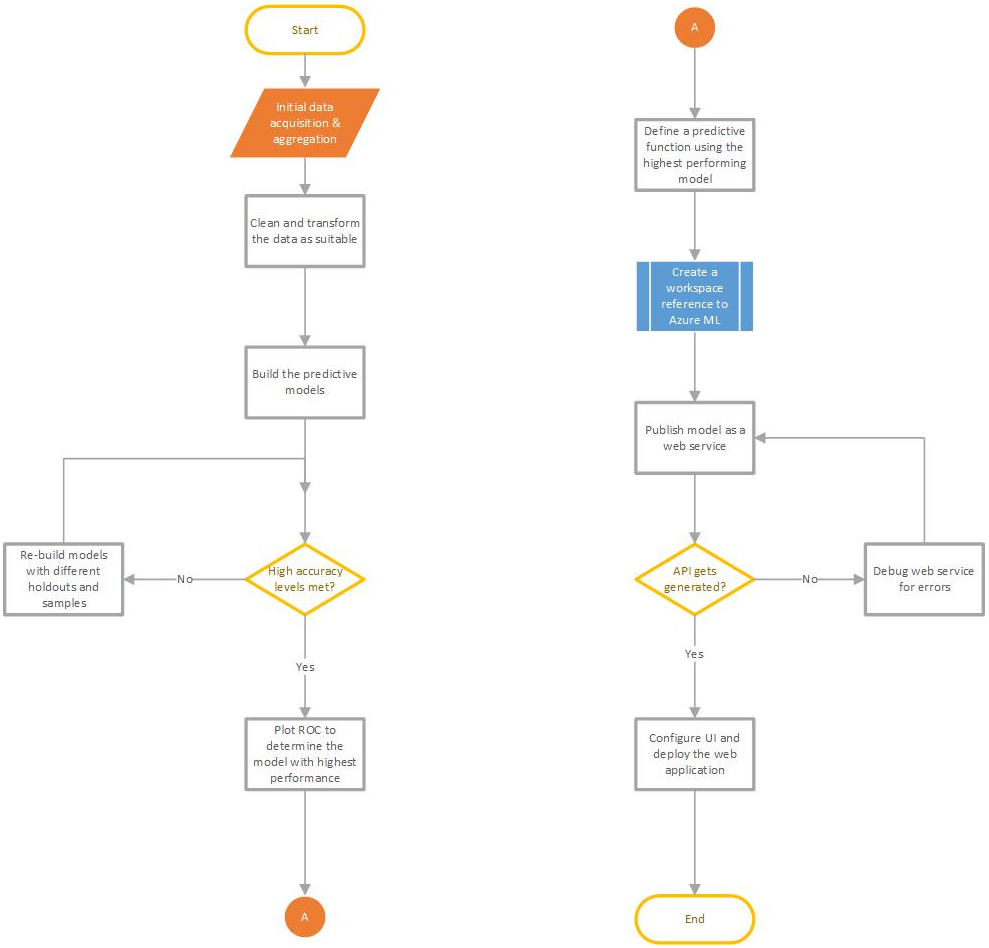

Figure 1: Process flow diagram

3.1 Data acquisition

The symptoms of the disease cited and justified earlier as the most common symptoms of the disease include speech impairment, excessive salivation, tremors, leg agility, rigidity, difficulty in swallowing and unintelligible speaking and additionally, the history of the disease in the persons family is also considered. These are the variables that are required in order to perform the analysis. The data as required has been acquired from two sources namely UCI Machine Learning repository and DataWorld. These are real datasets which has around 6000 instances approximately. Raw data has been collected from these repositories and will be transformed as suitable for the project based on the prerequisites mentioned in the literature review.

3.2 Machine Learning

The predictive models that needs to be built to perform the predictions are built with the help of Machine Learning techniques. The predictive models are initially trained using the existing data after which the new data called the test data will be given for it to make appropriate predictions. As justified in the literature review section, multiple models will be built here and the model with the most accuracy will be chosen further to develop the web application. This is one of the most important prerequisites to be satisfied to execute the project. Based on the data acquired, the models needs to be chosen and built. As

the proposed approach is a binary classification problem, models such as K-NN, SVM, Decision trees and Logistic regression will be built here with evidences and justification supporting the above decision being given in the literature review section. These models are well suitable for binary class problems and especially decision tree can handle robust data well and are built using R programming in the RStudio environment. All the necessary data cleaning and transformation will be done in the same environment before building the predictive models. Several measures are undertaken in order to increase the accuracy of the model after which the most accurate model will be considered to develop the web application.

3.3 ROC curve

The Receiver Operating Characteristic (ROC) is a broadly accepted performance measure for evaluating accuracy metrics of a model [28]. The ROC is a 2D graph where the x-axis is the measure of true positive rate while y-axis is the measure of false positive rate. A classifier is said to be perfect if its in the (0,1) point in the 2D space which is highly unlikely [28]. Thus, a model that is the closest to the (0,1) point is said to have the highest accuracy. The idea is to maximize the Area under the curve (AUC) which in turn will increase the accuracy of the model. This is one of the most important performance evaluation measure used in this project. A combined ROC plot consisting of four curves one for each model will be designed and the model with the highest accuracy will be chosen for the development of web application.

3.4 Web application development

The process of development of the web application begins once the predictive models are built and the one with the maximum accuracy is chosen. Considering the research gap and the drawbacks of the existing methods, the aim here is to build a simple, easy to use application which is not too overwhelming or complicated for the people to use. The web application consists of simple questions regarding the symptoms of the disease which the user will answer. This is done with the help of Microsoft Azure ML Studio and the Request Response Web application. The development of a web application involves the following steps:

- Defining a prediction function that is responsible for performing the predictions. The relevant input parameters are fed to the model with the highest accuracy which in turn returns the prediction value which in our case is the result of whether the person will have the disease or not.
- Once the prediction function is successfully defined, the model can be extended and published as a web service which then helps generate an API location and key enabling the model to be able to run on any platform.
- After publishing the web service, the API location and key is generated which is then fed to the Request Response Web application. Once the connection between the web application and the web service is successfully established, the user interface of the application can be designed. Again, the web application must have a simple user interface keeping in mind the needs and requirements of the user.

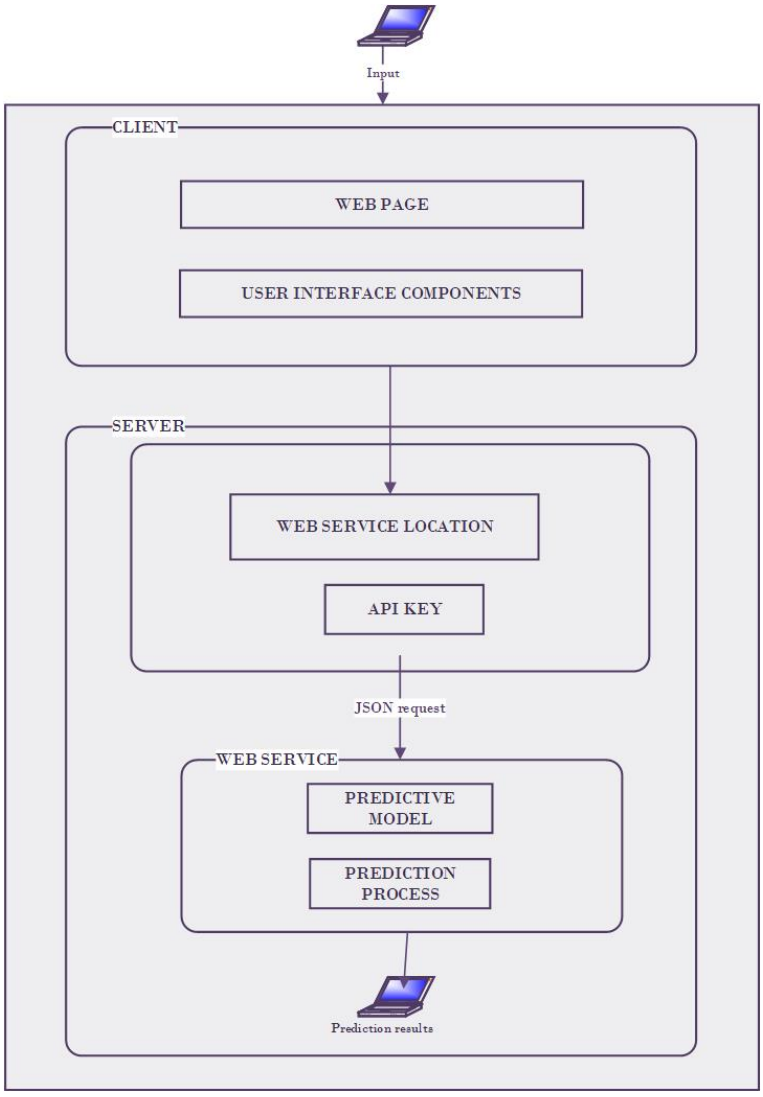

Figure 2: Architecture of the web application

These are the steps involving the development of a web application. As mentioned earlier, the proposed approach must make sure the necessary conditions are met before the execution of the project. The next section involves the implementation of the project where each and every step of the implementation of the project is explained in detail with supporting diagrams.

4 Design and solution development

This section explains the implementation of the project in depth. The various phases of implementation along with the different platforms and environments used, data structure and pre-processing, the flow of data from one platform to another etc are all discussed in detail with supporting evidences and justification for the chosen techniques being provided already. The various phases include:

4.1 Initial data acquisition and aggregation

As mentioned earlier, the data needed has been acquired from two sources namely UCI machine learning repository and DataWorld. The symptoms of the diseases are aggregated from the two sources with the dependent variable being the status of the disease which is categorical with values Yes and No. Similarly, the independent variables tremor, family history, leg agility are categorical while the rest are numeric variables. No data imbalance has been observed here and thus the results of the predictive models will not be biased. Predictions will be returned as either Yes or No based on the values of the symptom variables.

Name	Description	Data type
Sex	Sex of the person	String
Age	Age of the person	Integer
History	History of the disease in the person's family with values	String
Leg agility	Problems with the functioning of limbs and feet	String
Swallowing	Difficulty experienced while swallowing solid food	Integer
Tremor	Experiencing involuntary quivering movements	String
Unintelligible speaking	Involuntary talking and use of words	Integer
Motor	Difficulties faced with the entire motor functioning of the body parts	Integer
Salivation	Excessive involuntary salivation	Integer
Speech impairment	Frequent stuttering and stammering	Integer
Rigidity	Rigidity of the body parts	Integer

Figure 3: Data variables and description

4.2 Data cleaning and transformation

The initial aggregation of the data is completed but every dataset needs to be cleaned and transformed in a such a way that its suitable for the models to be built. The final accuracy and performance of the model mostly depends on this phase [27]. The data acquired from the second source has missing values i.e. number of instances were comparatively less than the first source. Thus, about 300 rows from the first source were deleted instead of imputing values as this is a highly sensitive data and imputing values might decrease the overall accuracy of the model. After cleaning, the data needs to be transformed in such a way that its suitable for the models to be built. Data transformation is done using R involving the following steps:

- Initially, all the character variables are converted to factors except for the Sex variable which will be dummy encoded. The values yes and no are converted to factors with labels 1 and 0 respectively.
- The numeric variables have to be normalized so that all the values fall under the same range because values under different ranges can impact the prediction accuracy of the model i.e. higher ranging values can have a greater impact over the accuracy which can lead to a biased prediction result Charleonnan et al. (2016). Min-Max normalization method has been implemented here to normalize the numeric data.

```
(X - min(X))/(max(X) - min(X))
```

where min(x) refers to the least value of the column while max(x) is the highest value. The character variable Sex should be dummy encoded since models such as K-NN calculates the distance between the variables to perform the prediction and character variables will cause the return of NA values. Thus, with the help of dummies package in R, this column is dummy encoded after which its separated into two columns one for male and the other one for female which will help the users give an input easily when the web application is designed. These transformation techniques have to be performed for models especially like K-NN since it does predictions by calculating the distance between variables Charleonnan et al. (2016). As all the required transformations have been done, the models can be built now.

4.3 Building predictive models

After much research, four models have been chosen to perform the prediction with the evidences and justification given earlier. This study uses a stratified 75/25 holdout and a training control is defined to control the parameters of the train function. This is done with the help of the caret library in R using the createDataPartition function and train control is defined where resampling is done with a 3 fold cross validation. All the models expect for SVM are built and evaluated using the same test and training samples with the same training control. Support vector machine uses a different training control with a repeated cross validation method. Now that all the initial pre-processing and transformations have been done, the models can be built. K-Nearest Neighbors performs predictions by calculating the closest distance between the training tuples and the unknown test tuple. The model is built and trained using the stratified training sample which is resampled with a 3 fold cross validation after which the predictions are obtained by testing the model with the test set. The optimal value of k was found to be 5. The same test and train set along with the same training control are used here to build the decision tree and the logistic regression models. The decision tree model is built with the help of c50 and libcoin packages in R where the training data along with the training control is defined. SVM is a model when used on the right data with the appropriate kernels can return the best results [25]. SVM requires the package e1071 to be installed before building it. A linear kernel along with a separate training control involving a repeated cross validation method with 10 folds is defined for SVM after which the model is built. The prediction results of each of the models have been obtained to plot the ROC curve only after which the web application can be developed.

4.4 ROC curve analysis

It is highly necessary to evaluate the models once before developing a web application since only one model can be used to define the prediction function that will do the predictions and present the result through the web application. A combined ROC has been plotted with one curve for each model based on which the model with the highest performance can be chosen to define the prediction function. As mentioned earlier, the model that is the closest to the (0,1) point is said to have the highest accuracy. The ROC

curve, that is plotted with the help of ROCR package requires all the predictions of the models to be converted to numerical values. The predictions are combined in a single list and then the curve is plotted which is shown below

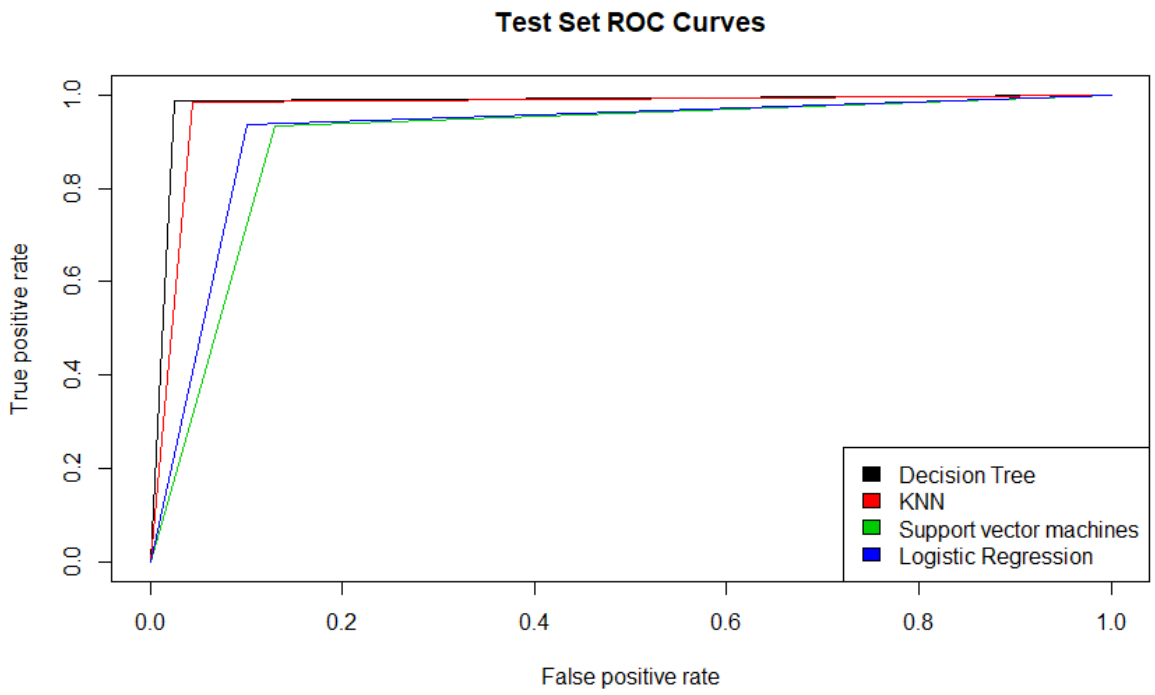

Figure 4: ROC curve

It can be seen from the above plot that decision tree and K-NN have the highest area under the ROC curve but only one model can be chosen to proceed further. A precise model performance metric needs to be obtained here to choose between those two models. Thus, the area under the ROC curve is determined which is presented below

Model	Area under the curve
K-NN	0.9697
Decision tree	0.981

Figure 5: Area under the ROC curve

Now, its evident that decision tree has performed the best and hence will be chosen to develop the application.

4.5 Defining a prediction function

As mentioned earlier, a prediction function will have to be defined here that will do the predictions in the web application. A function named predictdisease has been defined here which will do the predictions with the help of decision tree model built and all the other

input parameters except the dependent variable needs to be fed into it. It is important to note that without parsing a proper working model and all the input parameters correctly, the function will not work or at least will not return the appropriate prediction value which in this case is the status of the person. After defining the function successfully, it can be published as a web service which will generate the API key and the location.

4.6 Publish model as web service

To publish the model as a web service, the Azure ML package in R needs to be installed and loaded. It also requires an Azure subscription and an Azure ML workspace which can be created with the use of a Microsoft ID. Initially, to publish a web service from R, a workspace reference has to be created. This is done by obtaining the workspace ID and the authentication token from the settings section in the Azure ML workspace. Once this is obtained, the reference can be created using the following syntax:

```
ws <- workspace(id = wsID, auth = wsAuth, .validate = TRUE)
```

where id is the default workspace ID and auth is the default authorization token obtained from Azure ML Studio. The model is ready to be published as a web service once the reference to the Azure ML workspace is created. The predictdisease function along with the workspace reference and the inputschema which consists of all the input parameters are defined for the model to be published as a web service. This also requires an external zip program which will compress this information before transmitting them to Azure. This is done by installing RTools from which the zip program can be loaded and installed. This tool converts the inputschema into a data frame after which it is published as a web service which can be seen in the Azure ML workspace.

4.7 Configuration and deployment of web application

Once the model is published as a web service, an API key and location gets generated which can be seen in the R console. These are used to create and deploy the application which after creation uses this API to run the model in the backend and produce the prediction results in the web application to the user. Initially, a request response web application is deployed in Azure. After successfully deploying the application, the API key and the location is given here after which the user interface of the application can be configured i.e. the minimum and maximum value of each input is defined here as required and the web application is deployed. Now, as the deployment is completed, the input values can be given which will return the prediction values i.e. if the person will have the disease or not. As all the variables of the models are transformed into numerics for prediction purposes, entering exact numeric value for each input would be difficult for the user. Thus, a slider button is designed enabling the user to only give the approximate values as input after which the prediction results will be given. The application has been

designed in a simple, elegant way where the user has to answer very simple questions to get the results which is the whole point of this research anyway.

5 Evaluation

This section involves the evaluation of the performance of all the individual models and the web application. The performance can be evaluated by testing the trained models with the test set. A ROC curve was already plotted since it was required to evaluate the model performance before developing the web application. The rest of the performance metrics such as accuracy, sensitivity, specificity and kappa statistic will be discussed in depth in this section. Before beginning, a summary of each model is presented below.

C5.0

```
4407 samples
  15 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 2938, 2938, 2938
Resampling results across tuning parameters:

  model  winnow  trials  Accuracy   Kappa
  rules  FALSE    1      0.9752666  0.9497387
  rules  FALSE    10     0.9784434  0.9563134
  rules  FALSE    20     0.9807125  0.9608926
  rules  FALSE    30     0.9813932  0.9622587
  rules   TRUE    1      0.9752666  0.9497387
  rules   TRUE    10     0.9791241  0.9576602
  rules   TRUE    20     0.9825278  0.9645557
  tree    TRUE    10     0.9773088  0.9540262
  tree    TRUE    20     0.9795779  0.9586152
  tree    TRUE    30     0.9820740  0.9636477

Accuracy was used to select the optimal model using the
largest value.
The final values used for the model were trials = 20, m
odel = rules and winnow = TRUE.
```

Support Vector Machines with Linear Kernel

```
4407 samples
  15 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold, repea
ted 3 times)
Summary of sample sizes: 3966, 3967, 3966,
3967, 3966, 3966, ...
Resampling results:

  Accuracy   Kappa
  0.9078709  0.8122343

Tuning parameter 'C' was held constant at a
value of 1
```

Generalized Linear Model

```
4407 samples
  15 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 2938, 2938, 2938
Resampling results:

  Accuracy   Kappa
  0.9153619  0.8281971
```

k-Nearest Neighbors

```
4407 samples
  15 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 2938, 2938, 2938
Resampling results across tuning parameters
:

  k   Accuracy   Kappa
   5  0.9750397  0.9493175
   7  0.9736782  0.9465467
   9  0.9707284  0.9405839
  11  0.9707284  0.9405509

Accuracy was used to select the optimal mod
el using the largest value.
The final value used for the model was k =
5.
```

Figure 6: Model summary

For K-NN, the optimal value for k was found to be 5 by the model i.e. model has achieved the highest accuracy when the value of k is 5 where k is the number of Nearest Neighbors found. Since the control parameters for the train function have been defined, the model finds the optimal value by itself. Similarly, all the models have been tuned and resampled with a 3 fold cross validation and the optimal solution has been deduced for each one of them. This is beneficial because there is no need to change the optimization parameters such as the k value for K-NN or the tuning parameters for SVM such as c or sigma.

5.1 Accuracy, sensitivity and specificity.

All the models have performed reasonably well with good overall accuracies and kappa measure. Decision tree has achieved an accuracy of 98 percentage which is the highest among the four models. K-NN is the second best with 97 percentage accuracy. The accuracies of SVM and logistic regression are comparatively low although not bad with 90 percentage and 92 percentage respectively. Sensitivity and specificity can be defined as the proportion of positive and negative classes classified correctly. A table is presented below comparing the accuracy, sensitivity, specificity and kappa value of all the models.

Model	Accuracy	Sensitivity	Specificity	Kappa statistic
K-NN	0.9714	0.9795	0.9653	0.9419
Decision tree	0.9782	0.9769	0.9792	0.9558
Support vector machine	0.906	0.8706	0.9341	0.8085
Logistic regression	0.9203	0.9182	0.9219	0.8381

Figure 7: Accuracy, sensitivity and specificity

It is evident from the above table that decision tree has outperformed the rest of the models with respect to all the performance metrics. It is also important to note that the specificity value i.e. proportion of negative classes correctly classified of all the models are extremely high. This is good because in this particular case, sometimes it is okay to have a wrong positive class prediction, but the consequences of an incorrect negative class prediction might be catastrophic. Even the kappa statistics of all the models are reasonably good with decision tree being the highest.

5.2 Computational time and Area under ROC curve (AUC)

As mentioned earlier, AUC is a metric deduced to determine the performance of a model. The greater the area under the ROC curve, the better the performance of the model. The AUC of K-NN and decision tree have already been shown in the section 4.4. Another important aspect that needs to be evaluated is the computational time of the model. The lesser the computational time, more efficient the model is. These metrics of all the predictive models built are presented below

Model	Area under the curve	Model running time
K-NN	0.9697	1.97s
Decision tree	0.981	16.06s
Support vector machine	0.9023	18.37s
Logistic regression	0.9182	0.81s

Figure 8: Computational time and AUC

Thus, the above table again shows that decision tree has the highest AUC hence being the best performing model of the lot but has been a bit slow compared to K-NN which is the second best model with respect to AUC. SVM has been the slowest because it has to perform resampling using a 10 fold repeated cross validation method. K-NN is also good enough to be embedded in the web application but decision tree has been given the edge here considering its accuracy and robustness. If these work fine, the web application will automatically function properly and return the correct values.

6 Conclusion and future work

The main aim of the research was to develop an application that a user can use to predict the occurrence of Parkinsons disease. Gap in research has been identified and necessary steps have been taken to provide an effective solution for those drawbacks observed. To summarize, four effective predictive models have been built based on the diseases symptoms and among them decision tree has been chosen given its high accuracy and performance to develop the web application that will enable the user perform predictions with ease. The whole idea was to develop a simple means through which a user can efficiently predict the occurrence of this disease otherwise very hard to diagose. It is important to note that all conditions mentioned in section 3 as required to execute this project are met. A simple web application has been designed with basic questions for the users to be able to answer and get the prediction results.

6.1 Future work

The proposed approach can be improved in the future in the following ways:

- More number of symptoms can be considered instead of just ten so that more effective predictions can be performed.
- The resulting predictions just indicate if the person will have the disease or not. It can be improved in such a way that the prediction results provide in which stage of the disease they are likely to be diagnosed and what sort of treatments are to be undertaken.

Acknowledgement

I would like to thank my supervisor Mr Vikas Tomer for the continuous support, knowledge, advice throughout the entire process. His guidance towards this project is highly acknowledged.

References

Agarwal, A., Chandrayan, S. and Sahu, S. S. (2016). Prediction of parkinson's disease using speech signal with extreme learning machine, *Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on*, IEEE, pp. 3776–3779.

Aich, S., Choi, K., Park, J. and Kim, H.-C. (2017). Prediction of parkinson disease using nonlinear classifiers with decision tree using gait dynamics, *Proceedings of the 2017 4th International Conference on Biomedical and Bioinformatics Engineering*, ACM, pp. 52–57.

Aich, S., Sain, M., Park, J., Choi, K.-W. and Kim, H.-C. (2017). A mixed classification approach for the prediction of parkinson's disease using nonlinear feature selection technique based on the voice recording, *Inventive Computing and Informatics (ICICI), International Conference on*, IEEE, pp. 959–962.

Alves, V. M., Braga, R. C., Muratov, E. and Andrade, C. H. (2018). Development of web and mobile applications for chemical toxicity prediction, *Journal of the Brazilian Chemical Society* **29**(5): 982–988.

Bashir, S., Qamar, U., Khan, F. H. and Naseem, L. (2016). Hmv: a medical decision support framework using multi-layer classifiers for disease prediction, *Journal of Computational Science* **13**: 10–25.

Bhat, M., Inamdar, S., Kulkarni, D., Kulkarni, G. and Shriram, R. (2017). Parkinson's disease prediction based on hand tremor analysis, *Communication and Signal Processing (ICCSP), 2017 International Conference on*, IEEE, pp. 0625–0629.

Challa, K. N. R., Pagolu, V. S., Panda, G. and Majhi, B. (2016). An improved approach for prediction of parkinson's disease using machine learning techniques, *Signal Processing, Communication, Power and Embedded System (SCOPES), 2016 International Conference on*, IEEE, pp. 1446–1451.

Chandrayan, S., Agarwal, A., Arif, M. and Sahu, S. S. (2017). Selection of dominant voice features for accurate detection of parkinson's disease, *Biosignals, Images and Instrumentation (ICBSII), 2017 Third International Conference on*, IEEE, pp. 1–4.

Charleonnan, A., Fufaung, T., Niyomwong, T., Chokchueypattanakit, W., Suwannawach, S. and Ninchawee, N. (2016). Predictive analytics for chronic kidney disease using machine learning techniques, *Management and Innovation Technology International Conference (MITicon), 2016*, IEEE, pp. MIT–80.

Fayyazifar, N. and Samadiani, N. (2017). Parkinson's disease detection using ensemble techniques and genetic algorithm, *Artificial Intelligence and Signal Processing Conference (AISP), 2017*, IEEE, pp. 162–165.

Kuncheva, L. I. and Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, *Machine learning* **51**(2): 181–207.

Mazilu, S., Calatroni, A., Gazit, E., Roggen, D., Hausdorff, J. M. and Tröster, G. (2013). Feature learning for detection and prediction of freezing of gait in parkinsons disease, *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, Springer, pp. 144–158.

Memedi, M., Westin, J., Nyholm, D., Dougherty, M. and Groth, T. (2011). A web application for follow-up of results from a mobile device test battery for parkinson's disease patients, *Computer methods and programs in biomedicine* **104**(2): 219–226.

Mund, S. (2015). *Microsoft azure machine learning*, Packt Publishing Ltd.

Patel, S., Chen, B.-r., Buckley, T., Rednic, R., McClure, D., Tarsy, D., Shih, L., Dy, J., Welsh, M. and Bonato, P. (2010). Home monitoring of patients with parkinson's disease via wearable technology and a web-based application, *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, IEEE, pp. 4411–4414.

Perlmutter, J. S. (2009). Assessment of parkinson disease manifestations, *Current protocols in neuroscience* **49**(1): 10–1.

Przednowek, K., Wiktorowicz, K., Krzeszowski, T. and Iskra, J. (2018). A web-oriented expert system for planning hurdles race training programmes, *Neural Computing and Applications* pp. 1–17.

Pun, U. K., Gu, H., Dong, Z. and Artan, N. S. (2016). Classification and visualization tool for gait analysis of parkinson's disease, *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, IEEE, pp. 2407–2410.

Seal, A. and Wild, D. J. (2016). Netpredictor: R and shiny package to perform drug-target bipartite network analysis and prediction of missing links., *bioRxiv* p. 080036.

Śledzianowski, A., Szymański, A., Szlufik, S. and Koziorowski, D. (2018). Rough set data mining algorithms and pursuit eye movement measurements help to predict symptom development in parkinsons disease, *Asian Conference on Intelligent Information and Database Systems*, Springer, pp. 428–435.

Szymański, A., Szlufik, S., Koziorowski, D. M. and Przybyszewski, A. W. (2017). Building classifiers for parkinsons disease using new eye tribe tracking method, *Asian Conference on Intelligent Information and Database Systems*, Springer, pp. 351–358.

Tejeswinee, K., Shomona, G. J. and Athilakshmi, R. (2017). Feature selection techniques for prediction of neuro-degenerative disorders: A case-study with alzheimers and parkinsons disease, *Procedia Computer Science* **115**: 188–194.

Uwagbole, S. O., Buchanan, W. J. and Fan, L. (2017). Applied machine learning predictive analytics to sql injection attack detection and prevention, *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*, IEEE, pp. 1087–1090.

Xiao, D. and Zhang, J. (2009). Importance degree of features and feature selection, *Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on*, Vol. 1, IEEE, pp. 197–201.

Yadav, G., Kumar, Y. and Sahoo, G. (2012). Predication of parkinson's disease using data mining methods: A comparative analysis of tree, statistical and support vector machine classifiers, *Computing and Communication Systems (NCCCS), 2012 National Conference on*, IEEE, pp. 1–8.

Zhang, J., Xu, W., Zhang, Q., Jin, B. and Wei, X. (2017). Exploring risk factors and predicting updrs score based on parkinson's speech signals, *e-Health Networking, Applications and Services (Healthcom), 2017 IEEE 19th International Conference on*, IEEE, pp. 1–6.

Mund (2015)