Detecting Parkinson's Disease using Data-Driven Classification Model

Maryam Arief

Abstract—Parkinson's disease (PD) is a long term neurodegenerative disease which affects the central nervous system and causes loss of motor function. Around 10 million people across the world suffer from Parkinson's disease. There is no definitive test to detect PD and symptoms overlap with various other conditions too which is why diagnosis is difficult. There have been a number of Machine learning techniques which have proven to be helpful in early detection of PD, using datasets related to voice, changes in gait, smell identification tests and SPECT scan data. In this project, we use Machine learning techniques and voice measurements to differentiate between patients who are suffering from PD and those who are not. We perform a comparative analysis of Logistic Regression, KNN and XGBoost methods, and achieve the highest accuracy of 97% and an MCC value of 0.93 with KNN.

Index Terms—Parkinson's Disease, PD, KNN, XGBoost, Logistic Regression, Principal Component Analysis, PCA, Classification, EDA

1 Introduction

As of 2020, Nearly 145000 people in the UK live with Parkinson's disease (PD) [1]. It is the second most common neurological disease after Alzheimer's, and men are 1.5 times more likely to be diagnosed with PD than women [2]. PD is the fastest growing neurological disorder in the world, it affects the central nervous system, and causes loss of dopamine in the brain. Dopamine is a neurotransmitter which helps in the transmission of signals in the brain. The motor symptoms of PD are cause by striatal dopaminergic neuron death in the substantia nigra, which is an area in the thalamic region of the brain, this causes degeneration and loss of mobility, muscle control, movement and other symptoms associated with this disease [3][4].It is still unclear what exactly causes PD, however there have been some studies indicating genetic mutations or exposure to toxins as a cause [2]. While there is no definitive test for PD, early detection can help immensely in providing appropriate treatment and management of symptoms. However, it is not easy to diagnose in the early stages since the symptoms of PD resemble many other conditions too. Timely detection at an early stage can help slow down the progression of the disease and since the disease has usually progressed a lot by the time it is correctly diagnosed (approximately 60% nigrostriatal neurons are already damaged by then), using Machine Learning techniques could be a huge help in this case.

Machine learning (ML) has proven to be useful in the past for detection of PD with relatively high accuracy. Previous works have applied ML techniques to a variety of PD data, [5] used patient questionnaire data to create predictive ML models and achieved high accuracy in doing so, [6] used SVM and KNN classifiers on speech data, [7] used image sequences of human silhouettes for gait analysis to predict PD, and [8] used single photon emission computed tomography (SPECT) data and SVM and Logistic Regression to achieve a high accuracy in even early PD cases.

Since speech problems can be one of the first indicators of PD even before other major symptoms become apparent, in this project we use biomedical voice data to predict PD. We try 3 different classification techniques and evaluate our results to observe which is the best performing to help predict PD.

2 Background

A number of previous works have used speech data to predict PD. About 89% of PD patients suffer from some form of speech related problems, many suffer from reduced loudness, a monotone and breathy voice. This could be due to a disordered motor system or due to changes in sensory processing. These problems diminish the confidence of the patient to participate in conversations and could cause a number of problems for both the patient and their caregiver, which is why early detection and speech therapy is so important [9].

Using a Neural Network for early detection and diagnosis of PD, [10] achieved 80% accuracy on the same dataset we use in our project, the Oxford Parkinson's Disease Detection Dataset. They used only 8 attributes from the dataset on a multilayer feed forward neural network that is trained with Back Propagation. It is a fully connected network with one hidden layer of 10 nodes and only one output neuron. They achieved 80% accuracy, 83.3% sensitivity and 63.6% specificity.

[11] used deep learning to predict PD severity using Unified Parkinson's Disease Rating Scale (UPDRS) which classifies the severity of the disease. They used the Parkinson's Telemonitoring Voice dataset from the UCI ML repository. After preprocessing the dataset, they built a Deep neural network with 16 units in input and 3 hidden layers with 10, 20 and 10 neurons in each hidden layer respectively.The

2 outputs of the neural network were 'severe' and 'non-severe'. They obtained a better accuracy of 81.66% for motor UPDRS as compared to 62.37% for total UPDRS.

Another study that used neural networks was by [12], they experimented with various feature set techniques. They used a dataset of recorded voice samples of 20 healthy and 20 PD patients and performed feature selection using a variety of techniques, Pearson's and Kendall's correlation coefficient, PCA and Self organizing maps (SOM). They then tested 5 different Artificial Neural Network (ANN) configurations, 2 with 1 hidden layer, 2 with 2 hidden layers having 5-5 and 10-10 neurons respectively and one with 3 hidden layers (5-10-5) and trained the ANN for each set of selected features. The best test accuracy achieved was with Kendall's correlation coefficient with ANN of 1 hidden layer of 10 neurons, with accuracy of 0.8133.

[13] used a voice dataset of 756 instances and 754 attributes from 188 PD patients. After preprocessing the data, they evaluated the performance on various ML algorithms Logistic Regression, Naive Bayes, KNN, Random Forest, Decision Tree, SVM, MLP, and XGBoost. The results showed XGBoost had the highest accuracy of 88.15% and Naive Bayes performed the worst out of all classifiers.

[14] used an iPhone application to collect voice data and analysed the performance of various ML Classifiers on this data. The raw audio was cleaned using the VoiceBox's Voice Activation Detection (VAD) algorithm, activlev, then preprocessed it using PyAudioAnalysis library in Python which resulted in 11 unique features. They used 2 methods, one from AVEC 2013 for preliminary audio analysis, the method of Minimum Redundancy Maximum Relevance (mRMR) which resulted in an array of ranked features. They also passed it into the The Geneva Minimalistic Acoustic Parameter Set (GeMaps) using the openSMILE toolkit set which extracted approximately 62 lower level features per audio sample. Both feature sets contain Mel Frequency Cepstral Coefficients (MFCC), they applied the features on Decision Trees (DT), Extra Trees (ET), Gradient Boosted Decision Tree (GBDT), Artificial Neural Network (ANN), Random Forest (RF) and Support Vector Machine (SVM).The results showed that AVEC features outperform the GeMaps features due to possibly more information encoded within them. The best performing classifier was the GBDT with an accuracy of 86%.
Another study by [15] used SVM with Radial basis function (RBF) kernel for classification of voice datasets of patients suffering from PD. They used 2 distinct datasets, one from the US and one from Germany. They performed 3 different approaches, in first they examined each country's dataset using Leave One Out Cross Validation SVM on it's own to search for the Feature Set that maximised generalisation results on the respective datasets. In the second approach they used the American dataset for training and German dataset for testing and vice versa, and they combined the 2 datasets in the third approach. They achieved the best results for the first approach and concluded that it appears from the results that optimal feature sets are language dependent.

3 METHODOLOGY

The following Steps were performed for Classification of the features to Parkinson's or No Parkinson's. We aim to achieve an accuracy of atleast 94% on the 3 classifiers, Logistic Regression, KNN and XGBoost. The entire project was completed in Python 3 on Google Colaboratory, a web IDE for Python code.

Step 1: Data Collection

The Data used in this project was the Oxford "Parkinson's Disease Data Set" obtained from the UC Irvine Machine Learning Repository [16]. The dataset consists of 197 rows and 23 attributes, each row consists of biomedical voice measurements from 31 people, 23 with Parkinson's disease. There are around 6 recordings per patient [17]. The dataset was read using the `read_csv` function from Python's pandas library. The attributes are name, MDVP:Fo(Hz), MDVP:Fhi(Hz), MDVP:Flo(Hz), MDVP:Jitter(%), MDVP:Jitter(Abs), MDVP:RAP, MDVP:PPQ, Jitter:DDP, MDVP:Shimmer, MDVP:Shimmer(dB), Shimmer:APQ3, Shimmer:APQ5, MDVP:APQ, Shimmer:DDA, NHR, HNR, RPDE, DFA, spread1, spread2, D2, PPE and status.The value of status is 0 for healthy subjects and 1 for Parkinson's.

Step 2: Exploratory Data Analysis and Visualization

We begin analyzing the dataset to observe if there are any missing values which need our attention, any corrections we could possible make, or any patterns we can spot in the dataset before we can start applying our classification algorithms.

After creation of our Pandas dataframe (df) we use df.describe() which gives us an idea of various statistical parameters of the dataset for example the count, mean, standard deviation, minimum and maximum values and quartiles etc. so we can know the approximate range in which our values lie.

Once again we re-confirm if there are any missing values using the isna() function, even though the counts showed no missing values for all the features in our previous step.

Before we proceed with the next steps, we ensure that the entire data frame is numeric. Now we visualise the counts of the status column (the column we have to predict) where 1 indicates the patient has Parkinson's and 0 indicates that they do not. Upon observing we notice that there is a data imbalance with 147 records with status 1 and only 48 with status 0.

We normalize our dataset using the MinMaxScaler to scale values between -1 and +1.

The number of features in the data frame is on the higher side, we have 22 features. We need to look for ways to reduce dimensionality and improve our understanding of this data frame. First we try to use a Linear Regression Method to rank our top important features, which we will use for visualising. We choose the top 9 ranked features (all positive values greater than 0) from this step and create an additional data frame with only these features.

Using seaborn's pair plot function we plot pair plots for both the original data frame and the newer one we

have just created. We also create correlation plots for both data frames. Both these plots help show us the correlation between each pair of features and identify relationships between them, pair plot helped us visualise it and correlation plot correspondingly helped us quantify the correlation. As expected the features that appear to be highly positively correlated had corresponding higher positive values in the correlation plot and similarly for features that appear to be negatively correlated or not be correlated at all.

Having observed the relationships between pair of features, we would like to observe:
i) The distribution of values of individual features
ii) How similar or different are they with respect to the 2 classes.

We plot the graphs of every single feature and try to observe if it's possible to linearly separate the graph using any one feature. None of the features are linearly separable. We also plot and observe the distribution plots of every feature, and how they differ with each class using FacetGrid from seaborn. The distributions mostly follow a Normal distribution and the values for each feature are higher and lower per status and differ for every feature, there is no clear trend which could be sufficient to classify the 2 labels. We move on to prepare our dataset before using a Classification Algorithm.

PCA or Principal Component Analysis is a dimensionality reduction method used in datasets where they have a large number of features, but which can be reduced considerably. PCA ensures that reducing the dimensionality results in minimum loss of information by converting the dataset into a set of principal components derived from the eigenvectors of the covariance matrix [18]. We split our main data frame into Training and Testing sets in an 80/20 ratio with 80% data for training set and 20% for testing, and set the stratify parameter in the splitting of the train and test sets (considering the imbalance we had found previously). We then perform PCA, and plot and observe our scree plot. The scree plot tells us that more than 90% of the information can be explained by using 5 Principal Components, we keep this in mind for our next step.

Step 3: Training the dataset and Hyperparameter tuning

Since this is a binary classification problem, we have a plethora of algorithms to choose from. The aim of the project is to achieve at least 94% accuracy so we shall try a few algorithms which have been proven to provide good results.

We begin with a simple Logistic Regression classifier, which is one of the most commonly used classifiers for binary classification problems and provides good results. By setting the number of components for PCA as 5 (as we had found out in the previous step), we create the train PCA and test PCA datasets and calculate the accuracy. We also calculate accuracy for the datasets without any feature selection. Both times we achieve an accuracy of 84.6% which is considerably good but not above our aim of 94%.

Next we try K Nearest Neighbours or KNN, and once again calculate accuracies for both regular and PCA sets. KNN is a classification method that uses distance function as a similarity metric to classify new cases. The results showed that both PCA accuracy and without PCA accuracy is pretty good and better than Logistic Regression, we achieved values of 92.3% and 89.7% respectively. While they are great results, they are still around 2% shy of the target accuracy of 94%. We go ahead and tune the hyper parameters in hopes that since the accuracy is so close to our target, we might be able to achieve the target by adjusting our hyper parameters. Using GridSearchCV which helps us choose the best parameters, the new parameters we receive which are considered to give us better results are leaf size of 1 (leaf size effects speed and memory of the tree), number of neighbours is 1 and p is 2 (power parametric of Minkowski distance measure), we then substitute these values and train and test the dataset on this new model again. We find that we have achieved an improved accuracy of 97.4% on the PCA dataset and 92.3% on the regular dataset.

We now try XGBoost or eXtreme Gradient Boost, which is a fast and powerful ensemble classification method. The idea of an ensemble classification method is that instead of using just 1 classifier, we choose an ensemble of classifiers and by employing a voting system which chooses the best prediction [19]. This time we achieve 89.7% accuracy for the PCA dataset and 94.8% accuracy for the dataset without feature selection.

4 Results

The results of our experiment showed that KNN and XGBoost are the best performing classifiers for our problem and we have achieved accuracy above our target of 94% on the test set. Other than accuracy, we have used a number of other measures to evaluate our accuracy. In this project since we are dealing with an imbalanced dataset, we choose to use the Matthews Correlation Coefficient (MCC) which shows the quality of binary classification, keeping data imbalance in mind.It outputs a value between –1 and +1. If the coefficient value is closer to +1 it implies the classifier makes near perfect predictions, and –1 implies the opposite scenario, and 0 implies that the classification is equivalent to random guessing.

Matthews Correlation Coefficient:

$$\frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

where TP stands for True positive, FN stands for False negative, TN stands for True Negative and FP stands for False Positive values.

The Results are summarised in the following tables in the next page:

Accuracy Results of the Classification

Classification Method	PCA	Regular	PCA (after hyperparameter tuning)	Regular (after hyperparameter tuning)
Logistic Regression	84.6	84.6	NA	NA
KNN	92.3	89.7	97.4	92.3
XGBoost	89.7	94.8	NA	NA

Fig. 1. Accuracy comparison of the classifiers

Measures	Logistic Regression	KNN	XGBoost
Sensitivity	0.70	0.90	1.00
Specificity	0.89	1.00	0.93
Precision	0.70	1.00	0.80
Negative Predictive Value	0.89	0.96	1.00
False Positive Rate	0.10	0.00	0.06
False Discovery Rate	0.30	0.00	0.20
False Negative Rate	0.30	0.09	0.00
F1 Score	0.70	0.95	0.88
Matthews Correlation Coefficient	0.5966	0.9369	0.8651

Fig. 2. Comparison of other performance metrics

5 Discussion

The results of this experiment showed that KNN with hyper parameter tuning and XGBoost were the best performing classifiers. In the case of XGBoost, since we achieved the target accuracy with the default hyper-parameters, there did not seem to be any need for hyper-parameter tuning. PCA while definitely reduces dimensionality, does not seem to have any direct effect on the accuracy, where we have seen 2 cases where in one case the PCA test set achieves a higher score and another where it achieves lower than the dataset with no feature selection. The MCC values helped us see that KNN is indeed the best performing algorithm, and inspite of the seemingly high accuracy of logistic regression the values of MCC are closer to 0.5 in that case which indicates that the Logistic Regression method is not any better than random guessing.

6 Conclusion

The speech data proved to be very useful in classifying a patient with Parkinson's or not. The results are promising for early diagnosis, considering how Parkinson's is usually not very easy to diagnose. Our top scoring methods are of KNN and XGBoost, in the future, their efficiency can be tested on a larger dataset perhaps. Future datasets which are more balanced could also be used to observe if these results are reproducible. Maybe further data underlying the gender/age of the patients could be helpful in identifying any special or interesting patterns and relationships which could help improve diagnosis too. It would also be interesting to observe whether there are any differences in performance of algorithms with datasets in different languages as observed in the previous study of the German language vs English. In conclusion the project helps confirm that detection of Parkinson's disease is possible using Machine Learning techniques, gives promising results and could prove useful in health care and early detection and diagnosis.

REFERENCES

[1] Parkinson's UK. 2021. *Parkinson's UK*. [online] Available at: <https://www.parkinsons.org.uk/>

[2] de Lau, L. M., & Breteler, M. M. (2006). Epidemiology of Parkinson's disease. *The Lancet. Neurology*, *5*(6), 525–535. https://doi.org/10.1016/S1474-4422(06)70471-9

[3] DeMaagd, G., & Philip, A. (2015). Parkinson's Disease and Its Management: Part 1: Disease Entity, Risk Factors, Pathophysiology, Clinical Presentation, and Diagnosis. *P & T : a peer-reviewed journal for formulary management*, *40*(8), 504–532.

[4] Emamzadeh, F. N., & Surguchov, A. (2018). Parkinson's Disease: Biomarkers, Treatment, and Risk Factors. *Frontiers in neuroscience*, *12*, 612. https://doi.org/10.3389/fnins.2018.00612

[5] Prashanth, R., & Dutta Roy, S. (2018). Early detection of Parkinson's disease through patient questionnaire and predictive modelling. *International journal of medical informatics*, *119*, 75–87. https://doi.org/10.1016/j.ijmedinf.2018.09.008

[6] B. E. Sakar *et al*., "Collection and Analysis of a Parkinson Speech Dataset With Multiple Types of Sound Recordings," in *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 4, pp. 828-834, July 2013, doi: 10.1109/JBHI.2013.2245674.

[7] Chien-Wen Cho, Wen-Hung Chao, Sheng-Huang Lin, You-Yin Chen,
A vision-based analysis system for gait recognition in patients with Parkinson's disease,
Expert Systems with Applications,Volume 36, Issue 3, Part 2, 2009, Pages 7033-7039,ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2008.08.076.

[8] R. Prashanth, Sumantra Dutta Roy, Pravat K. Mandal, Shantanu Ghosh,
Automatic classification and prediction models for early Parkinson's disease diagnosis from SPECT imaging, Expert Systems with Applications, Volume 41, Issue 7, 2014, Pages 3333-3342, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2013.11.031.

[9] Parkinson's Foundation. 2021. *Speech Therapy and Parkinson's*. [online] Available at: <https://www.parkinson.org/pd-library/fact-sheets/Speech-Therapy>

[10] R. F. Olanrewaju, N. S. Sahari, A. A. Musa and N. Hakiem, "Application of neural networks in early detection and diagnosis of Parkinson's disease," *2014 International Conference on Cyber and IT Service Management (CITSM)*, South Tangerang, 2014, pp. 78-82, doi: 10.1109/CITSM.2014.7042180.

[11] Srishti Grover, Saloni Bhartia, Akshama, Abhilasha Yadav, Seeja K.R.,
Predicting Severity Of Parkinson's Disease Using Deep Learning,
Procedia Computer Science, Volume 132, 2018, Pages 1788-1794, ISSN 1877-0509,
https://doi.org/10.1016/j.procs.2018.05.154.

[12] Berus, L., Klancnik, S., Brezocnik, M., & Ficko, M. (2018). Classifying Parkinson's Disease Based on Acoustic Measures Using Artificial Neural Networks. *Sensors (Basel, Switzerland)*, *19*(1), 16. https://doi.org/10.3390/s19010016

[13] Iqra Nissar, Danish Raza Rizvi, Sarfaraz Masood, Aqib Nazir Mir
Voice-Based Detection of Parkinson's Disease through Ensemble Machine Learning Approach: A Performance Study
DOI: 10.4108/eai.13-7-2018.162806

[14] Wroge, T.J., Özkanca, Y., Demiroglu, C., Si, D., Atkins, D.C., & Ghomi, R.H. (2018). Parkinson's Disease Diagnosis Using Machine Learning and Voice. *2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, 1-7.

[15] H. Hazan, D. Hilu, L. Manevitz, L. O. Ramig and S. Sapir, "Early diagnosis of Parkinson's disease via machine learning on speech data," *2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel*, Eilat, 2012, pp. 1-4, doi: 10.1109/EEEI.2012.6377065.
[16] 'Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection', Little MA, McSharry PE, Roberts SJ, Costello DAE, Moroz IM. BioMedical Engineering OnLine 2007, 6:23 (26 June 2007)

[17] Archive.ics.uci.edu. 2021. *UCI Machine Learning Repository: Parkinsons Data Set*. [online] Available at: <https://archive.ics.uci.edu/ml/datasets/parkinsons>

[18] Jolliffe Ian T. and Cadima Jorge 2016 Principal component analysis: a review and recent developments *Phil. Trans. R. Soc. A.***374**20150202

[19] Bramer, M. (2013). Ensemble Classification. Principles of Data Mining. London, Springer London: 209-220.