**Parkinson’s Disease Vocals and Machine Learning**

**CS506 - Rough Draft**

**Daniella DeWeerd**

Background and Motivation

Parkinson’s Disease (PD) is a neurodegenerative disorder well known by the tremors it causes. It is also known to cause slowed movements, rigid muscles, impaired balance and posture, writing changes, loss of automatic movements, and speech changes [1]. While this is a well-known disease it is not clear exactly how it develops. Due to this, there isn’t any proven ways to prevent it. As such, it is super important to diagnose it at an early stage so treatment can begin as early as possible. Currently, diagnosis methods are as follows: looking at your medical history, reviewing your symptoms, imaging tests to rule out other disorders, utilizing current Parkinson’s medications to determine if they help [2]. Putting all these together as well as other similar tests, a doctor will give you a certain percentage of certainty that you have or don’t have Parkinson’s. This uncertainty can lead to either a correct diagnosis or a misdiagnosis. A misdiagnosis can lead to the actual disorder going untreated to a point where the progression is unable to be slowed or stopped which is why it is important for a neurologist to have as many tools available to be able to have a correct diagnosis.

The motivation of this project lies in the last sign of PD, speech changes. As PD progresses, the muscle of the vocal cord becomes thinner and less taught and changes the person’s voice as well as their intelligibility which is a change that could hopefully be noticed in the early stages of PD to help with diagnosis. This occurs in approximately 89% of people with PD and therefore it would be helpful for this difference in speech to be identifiable [3]. As such, the idea of measuring people with PD and without PD’s vocals came about. Max Little of the University of Oxford released a public dataset in 2008 containing data from 31 people, 23 of which with PD [4]. To best help doctors with diagnosing PD, I thought a computational outlook may help and so machine learning is the focus of this project. The major questions are as follows. Is it possible to predict PD from vocals with machine learning? What algorithm works best to predict Parkinson’s Disease (PD) vs the healthy controls? How accurate can the results be?

Exploration

The first step in utilizing this data to help with a diagnosis is to explore it. The data is from those patients with 6 recordings from each patient. The following are the features that are shown in the data: Average vocal fundamental frequency, Maximum vocal fundamental frequency, Minimum vocal fundamental frequency, measures of variation in fundamental frequency (MDVP:Jitter(%), MDVP:Jitter(Abs), MDVP:RAP, MDVP:PPQ, Jitter:DDP), measures of variation in amplitude (MDVP:Shimmer, MDVP:Shimmer(dB), Shimmer:APQ3, Shimmer:APQ5, MDVP:APQ, Shimmer:DDA), two measures of ration of noise to tonal components in the voice, two nonlinear dynamical complexity measures, a signal fractal scaling exponent, nonlinear measures of fundamental frequency variation (spread1,spread2,PPE), and whether a patient has or does not have Parkinson’s [4]. To further explore the data, I brought it into Python and created some charts. First, I compared the age of the patients who have PD vs the age without PD as well as the same versus but looking at the sex of the patients.

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

For the above figures, 0 is male and 1 is female. After that I split the data into training and testing, which is a common format for machine learning, and checked that it split the PD and non-PD people equally which is seen below.

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

Analysis

To find the most accurate diagnosis of PD using Machine learning, multiple algorithms where tested. Random Forest (RF), Logistic regression, SVM, Naïve Bayes Classifier, Neural Network Classifier, K-Nearest neighbor (KNN), PCA with Random Forest, and AdaBoost were all viable algorithms to test. After running a script that tested the base of each algorithm, it was found that Random Forest and KNN preformed the best with RF pulling just above a 90% accuracy on average and KNN pulling around 93%.

To further refine these algorithms, I used Grid Search to find the best parameters to get the best accuracy. The ending results of this was RF improving to around a 91% and KNN improving to 95%. With these results, the questions posed at the beginning can be answered. A 95% accuracy is most definitely good enough help a neurologist’s confidence in their diagnosis. It can also be seen that KNN is best for predicting with this data and it can get up to around 95% accuracy.

There are of course some pitfalls to this including that the pool of people used is small and leans towards PD. There is also the fact that the recordings were done in a specific environment and it wouldn’t be easy to recreate that environment every time a doctor wants to diagnose a patient. That was fixed in a later study done by Max Little where there are more people involved and the audio recordings were done with an app on the person’s phone. That study used a total of 726 unique participants with 262 of them having PD. This study was able to produce a 75.3% accuracy using a gradient boosted decision tree model called XGBoost [5]. This is interesting considering our second-best results were seen using a decision tree model as well.

In another study done by Max Little, their results for classifying the same dataset we used can be found. They classified using a SVM and received around a 90% accuracy [6] while our SVM achieved an 85% accuracy. Overall, our tuned KNN had a 5% better accuracy than their ending results, so it would be interesting to run their newest data through a similar pipeline to see if better results can be found.

Conclusion

Parkinson’s Disease is a neurodegenerative disorder that affects many people. When those people go to get diagnosed, they want to know that the doctor is most confident as they can be. To help with that, this study was run to try and find a tool to improve the doctor’s confidence as well as the patients. The results of this study show that it is possible to create such a tool with a very good accuracy rate compared to past studies that have been done. It was also found that newer data has been collected that could make a tool like we found more accessible to the general population. Overall, a tool like what was created over the course of this project could help improve the lives of patients by helping ensure they receive a correct diagnosis.

**References**

1 <https://www.mayoclinic.org/diseases-conditions/parkinsons-disease/symptoms-causes/syc-20376055#:~:text=Parkinson's%20disease%20is%20a%20progressive,stiffness%20or%20slowing%20of%20movement>.

2 <https://www.mayoclinic.org/diseases-conditions/parkinsons-disease/diagnosis-treatment/drc-20376062#:~:text=No%20specific%20test%20exists%20to,a%20neurological%20and%20physical%20examination>.

3 <https://www.parkinson.org/pd-library/fact-sheets/Speech-Therapy>

4 <https://archive.ics.uci.edu/ml/datasets/parkinsons>

5 <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8564663/>

6 <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3051371/>

**The Code and How to Navigate it:**

There are a total of 4 code files contained within the GitHub repo.

Parkinsons.py contains the original efforts to find the best algorithm as well as some of the code used to create the graphs within this paper. It can be run to display the accuracies for each algorithm with the original data and can be modified slightly to be run on other datasets as well as a basic algorithm finding tool.

PD\_Highest\_Accuracy.py contains the untuned RF and KNN algorithms and their accuracies. PD\_Final\_Program.py showcases how the tuning process occurred. Both can be simply ran to see the results.

The last program is most similar to what a tool used by neurologists might be and is named PD\_Classification\_Pipeline.py. In it there is the normal dataset as well as a spot at the bottom of the code with a variable called filename, more specifically it is in the 56th line of code. This variable can be replaced by a csv containing data from patients in a similar format to the original data, but without the status column containing whether a patient does or doesn’t have PD. The specific columns required can be found at the 63rd line of code containing the column names. Once this filename variable is replaced with the file name in quotations (such as the following: “patients.csv”), the program can be run and will output the classifications for the patients in the csv that was added in as well as the accuracy of the algorithm in the console.