Detecting Parkinson's Disease

October 01, 2021

A. Introduction

A1. Background

Parkinson's Disease (PD) is a progressive nervous system disorder that affects movement. It affects more than 1% of the population above 60 years old with both motor and non-motor symptoms of escalating severity as it progresses. Initial motor symptoms are usually very subtle and, as a result, patients seek medical assistance only when their condition has substantially deteriorated. Thus, missing the opportunity for an improved clinical outcome.

This highlights the need for tools than can detect Parkinson's Disease symptoms so that individuals can act accordingly. In this project, we would take a data-driven approach for the early detection of Parkinson's Disease by developing a machine learning model that is based on biomedical voice measurements. The aim is to use this model to accurately recognize if a patient has Parkinson's Disease or not.

A2. Problem

For this project, the question that can be asked: "What Machine Learning Model can be used that can accurately detect Parkinson's Disease using data from voice recordings?"

A3. Target

The intended audience for this project is those who are interested in using biomedical voice measurements to be able to detect PD. As there are many ways to detect PD using different resources, this would be beneficial for those who are only able to use voice recordings to give accurate diagnosis.

B. Data Acquisition and Cleaning

B1. Data Source

For this project, we need the following data:

1. Dataset that contains biomedical voice measurements:

Data Source = Kaggle: "https://www.kaggle.com/vikasukani/parkinsons-disease-data-set/version/1"

Description: The Kaggle Dataset consists of 195 voice recordings and 24 attributes.

Matrix column entries (attributes):

- name - ASCII subject name and recording number
- MDVP:Fo(Hz) - Average vocal fundamental frequency
- MDVP:Fhi(Hz) - Maximum vocal fundamental frequency
- MDVP:Flo(Hz) - Minimum vocal fundamental frequency
- MDVP:Jitter(%), MDVP:Jitter(Abs), MDVP:RAP, MDVP:PPQ, Jitter:DDP - Several measures of variation in fundamental frequency
- MDVP:Shimmer, MDVP:Shimmer(dB), Shimmer:APQ3, Shimmer:APQ5, MDVP:APQ, Shimmer:DDA - Several measures of variation in amplitude
- NHR, HNR - Two measures of the ratio of noise to tonal components in the voice
- status - The health status of the subject (one) - Parkinson's, (zero) – healthy
- RPDE, D2 - Two nonlinear dynamical complexity measures
- DFA - Signal fractal scaling exponent
- spread1, spread2, PPE - Three nonlinear measures of fundamental frequency variation

B2. Data Cleaning

The Kaggle Dataset initially have to be converted into a CSV file which was then downloaded in the data assets column in Watson Studio. The file has 195 entries and 24 attributes.

I initially took the information about the dataset. This is to see what type each attribute is and to see if there are attributes that have missing values. When ensured that there is none, I started the pre-processing stage.

My first step is to separate the features and target of the dataset. The features included all attributes excluding “name” and “status” and this was put in a data frame called X. The Target only included the “status” attribute, in which the data frame is called Y. For this project, since the status of each patient is already one of the attributes, it can be used as a target variable for our model.

After we separated the dataset, we split the data into training data and test data. To do this, we used the train_test_split function from sklearn.model_selection. Using this function, we can split both X and Y data frames with the test size being 20% and the training data size consisting of 80% of the data. From splitting the data, we have 156 voice recordings in the training data and 39 voice recordings in the test data.

The next step was to standardize the data as the values in the attributes are not in the same range. To do this, the StandardScaler() function is used. Before we can transform the data though, we have to use the fit method for the X_train. The fit method calculates the mean and variance of each of the features present in our data. Then the transform method is used for both

X_train and X_test data. The transform method transforms all the features using the respective mean and variance. We want scaling to be applied to our test data and at the same time do not want to be biased with our model. We want our test data to be completely new for our model; the transform method helps in this case.

Lastly, looking at the distribution of the target variable, there are 147 patients whose "status" is Parkinson's positive and 48 who are considered healthy.

C. Exploratory Data Analysis

For this project, the aim is to use Support Vector Machine model to accurately detect Parkinson's Disease in a patient. The objective of this algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies data points. There are many possible hyperplanes that could be chosen to separate the two classes of data points. The objective of SVM model is to find a plane that the maximum distance between data points of both classes. This provides some reinforcement so that future data points can be classified with more confidence.

To use this algorithm, we imported the svm library from sklearn. Then we used the SVC (Support Vector Classifier) as we specifically want to classify patients to either healthy (0) or Parkinson Positive (1). The objective of SVC is to fit the data and returning a "best fit" hyperplane that divides, or categorizes, the data. After getting the hyperplane, we can then feed some features to the classifier to see what the "predicted" class is.

For our model, we specifically used a "linear" kernel as the data can be separated using a single line. Then the training data is used to train the SVM model. Next, we used the model to predict either the patient is 1 or 0 using the features training data. To test the accuracy score of this prediction, we used the accuracy_score function imported from sklearn.metrics. We obtained the prediction and compared it to the actual values – target training data – and printed the accuracy score. For this we got an accuracy score of 88%. We also did the same process for the test data where we used the features test data as the input to predict a patient's classification using the SVM model. We then tested the accuracy score by comparing the prediction and actual target test data where we got an accuracy score of 87%. Since the accuracy score between the two datasets are very similar, the model is considered good. There is no big difference that indicates a problem of overfitting or underfitting.

C1. Mean of Attributes based on Target Variable

Before model analysis, we grouped the data based on the target variable and took the means of each attribute. From this we can compare the mean of each status on each attribute and obtain insights. From this analysis, we conclude that a healthy person a higher mean in all attributes excluding Average, Maximum, Minimum Vocal Fundamental Frequency, and HNR.

C2. Building a Predictive System

Now, we can use the SVM model created to test several voice recording data to classify whether a patient has Parkinson's or not. To do this, we took a row from the original dataset and deleted the "name" and "status" values. This would the input data. We initially changed the input data to a numpy array and then reshaped it to a one shape dimension. We did this as the SVM model requires several input data. After that, we standardized the data by using the transform() function and then used it as input to predict the classification. Since we already know the "status" of the patient, it easier to check if the model predicted it right.

D. Conclusion

Detecting the presence of Parkinson's Disease in a patient using biomedical voice recordings has been made easier with SVM models. With an accuracy score of 0.88 for the training data and 0.87 for the test data, the SVM model generated is considered very good in predicting the patient's classification. From this model, we can build a predictive system that can be used by medical institutions as one of its guides to diagnose a patient. Hopefully, this can help improved diagnosis such that people can get the treatment they need as early as possible.