Early detection of Parkinson's using voice samples and Application of Machine Learning

Anju Mathew, Sangeetha Sreekumar, Kshitiz Kharel, Koustav Dutta

University Of South Florida

ABSTRACT:

Machine Learning is soon becoming an asset for prognosis, it's helping doctors diagnose patients more accurately, make predictions about patients' future health, and recommend better treatments. Parkinson's is a neurological disorder, which leads to the shaking of the body and stiffness with treatment being possible at the early onset of the disease. We are proposing to design and test the accuracy of this Parkinson classification using various machine learning algorithms like Support vector machines, Neural networks and XGBoost to detect the best algorithm for early Parkinson detection from voice samples of people.

INTRODUCTION:

Parkinson's is a degenerative disorder of the nervous system that results from the death of dopamine containing cells in the brain, reason of which is unknown. Doctors and medical professionals can take advantage of using machine learning algorithms to detect the onset of Parkinson's by detecting abnormalities through speech and voice recordings. Since voice is one of the earliest indicators of Parkinson's, we use acoustic features to test for sustained phonations, articulation pauses and energy to make our analysis. Our project will use several machine learning classification models to analyze voice samples and test the accuracy of predictions about Parkinson's.

Support vector Machine:

Support vector machines is a powerful model capable of performing both classification and regression tasks. It is more commonly used for classification problems and does that by dividing data into two classes using a hyperplane. There are many ways to separate two points but our aim involves finding a plane with maximum margin i.e the maximum distance between data points of both classes. Maximizing the margin distance provides reinforcement which allows us to classify future points more accurately.

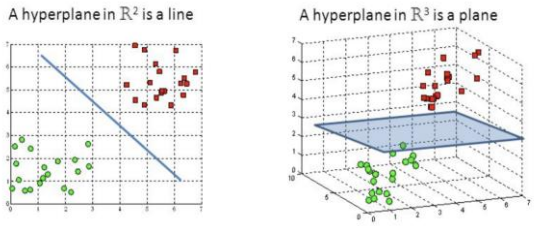


Fig1. Hyperplanes in 2D and 3D feature space

Hyperplanes boundaries help classify the data pointsand points on either side are attributed to different classes.The dimensions of the hyperplane depends on the number of features. We use SVM to classify patients into two categories based on the results of their voice samples and generate a plane to classify them as affected or healthy

Deep Neural Network:

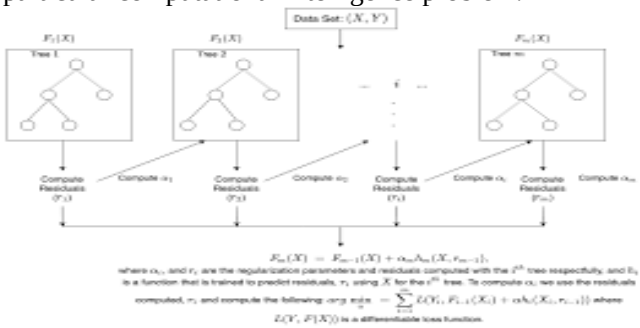
A Deep Neural Network is an Artificial Neural Network with multiple layers between input and output layers. For the binary classification, we have used DNN architecture with a threshold of 0.5.Artificial Neural Networks comprises of node layers, which has input layers, hidden layers and an output layer. Each neuron in the layers is connected to other neurons in other layers which have an associated weight and a threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. For our classification problem, If the output probability is greater than 0.5 than, it will be classified as 1, which refers that the patient has Parkinson's Disease.

Logistic Regression:

Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts P(Y=1) as a function of X.

XGBoost

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) ... A wide range of applications: Can be used to solve regression, classification, ranking, and user-defined prediction problems. Ensemble learning is the process by which multiple models, such as classifiers or experts, are strategically generated and combined to solve a particular computational intelligence problem.



We will try using XGBoost here and hyper tune the parameters to increase the accuracy. It is an implementation of gradient boosting machines created by Tianqi Chen, now with contributions from many developers. It belongs to a broader collection of tools under the umbrella of the Distributed Machine Learning Community or DMLC who are also the creators of the popular mxnet deep learning library.

Dataset Description:

The dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD) containing 6 recordings for each patient and 7th recordings for 3 patients with PD. The dataset contains 195 voice recordings of the people where 147 is labelled as 1 which refers to the people who had Parkinson's, and the rest 48 is labelled as 0 which refers to the people who didn't have Parkinson's.

Various speech signal processing algorithms including Time Frequency Features, Mel Frequency Cepstral Coefficients (MFCCs), Wavelet Transform based Features, Vocal Fold Features and TWQT features have been applied to the speech recordings of Parkinson's Disease (PD) patients to extract clinically useful information for PD assessment.

```
data.columns
```

```
Index(['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', 'status', 'RPDE', 'DFA',
       'spread1', 'spread2', 'D2', 'PPE'],
      dtype='object')
```

Name: ASCII, Subject name, and the recording number
MVDP: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 ratio of noise to tonal components in the voice
RPDE,D2 - Two nonlinear dynamical complexity measures
DFA - Signal fractal scaling exponent
spread1,spread2,PPE - Three nonlinear measures of fundamental frequency variation
status - Health status of the subject (one) - Parkinson's, (zero) – healthy

Data preparation:

Reading the dataset:

We are using preprocessed voice samples from a UCI Machine language repository and use Pandas to read our data into our file for analysis. These recording are already broken down into its various components like jitter, shimmer, PPQ and various frequencies.

```
# Loading the data from csv file to a Pandas DataFrame
parkinsons_data = pd.read_csv('parkinsons.csv')
# printing the first 5 rows of the dataframe
parkinsons_data.head()
```

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	...
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	...
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	...
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	...
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	...
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	...

Grouping:

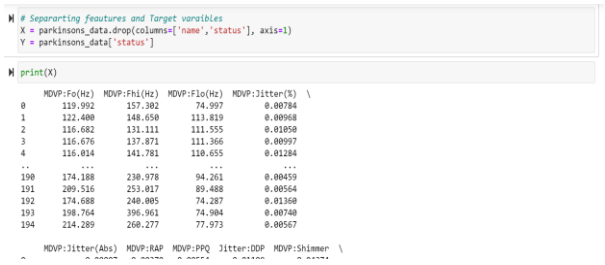
We are grouping the data based on our target variable by using a calculation of mean values.

```
# grouping the data bas3ed on the target variable
parkinsons_data.groupby('status').mean()
```

status	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	MDVP:Shimmer(d
0	181.937771	223.636750	145.207292	0.003866	0.000023	0.001925	0.002056	0.005776	0.017615	0.162
1	145.180762	188.441463	106.893558	0.006989	0.000051	0.003757	0.003900	0.011273	0.033658	0.321

Separating target and Feature variables:

We separate our data into X and Y with Target and feature variables by keeping only relevant data in each to run our models.



```
# Separarting feautures and Target varaibles
X = parkinsons_data.drop(columns=['name','status'], axis=1)
Y = parkinsons_data['status']
```

```
print(X)
```

```
     MDVP:Fo(Hz)  MDVP:Fhi(Hz)  MDVP:Flo(Hz)  MDVP:Jitter(%)  \
0        119.992       157.302        74.997         0.00784
1        122.400       148.650       113.819         0.00968
2        116.682       131.111       111.555         0.01050
3        116.676       137.871       111.366         0.00997
4        116.014       141.781       110.655         0.01284
..           ...           ...           ...             ...
190      174.188       230.978        94.261         0.00459
191      209.516       253.017        89.488         0.00564
192      174.688       240.005        74.287         0.01360
193      198.764       396.961        74.904         0.00740
194      214.289       260.277        77.973         0.00567

     MDVP:Jitter(Abs)  MDVP:RAP  MDVP:PPQ  Jitter:DDP  MDVP:Shimmer  \
```

Training:

After preprocessing, the next step is to train the model. We built 4 different models, Support Vector Machines, Deep Neural Network, XGBoost and Logistic Regression for our binary classification problem. We have split the data as Training data 80% and Test data 20%
Since, there is imbalanced amount of data in our target variable. For train-test split, we have used stratification.

Deep Neural Network Architecture:

We have created a 5 layer dense deep neural network model with 3 hidden layers in the model with 100, 75 and 25 nodes respectively.
We have added a dropout layer to help the model with the regularization. We played around with different dropouts and found the optimum level of dropout when 40% dropout was used between first two hidden layers and 50% in second two hidden layers.

We have used Rectified Linear Unit (ReLu) activation function on all deep layers.A layer with ReLu as an activation function outputs the input directly if it is positive, otherwise it outputs a zero. This is the preferred activation function for our network since it accounts for backpropagation errors to help improve the performance. The final layer is a fully connected layer with 1 neuron which defines the classification of the model. Since it is the binary classification, we have used sigmoid activation function.

Hyper Parameters:

Optimizer: We have used Adam which is as the optimization algorithm that updates network weights iteratively based on training data with a learning rate of 0.005

Number of epochs: We have set the number of epochs to be 50. The audio inputs are sent into the model in batches of 25.

Early-Stopping: Early stopping is set in the model. The model converged better when early stopping was placed in the model with the patience level of 5.

XGBoost Architecture : XGBoost has given best accuracy.So we tried hypertuning the parameters of XGBoost to get better results,but the accuracy was not going up. Increasing the gamma rate actually brought the accuracy down. We think it is due to the fact that the data set is not that huge. Rest other tuning that has been tried out in the code didn't increase the accuracy.

Predictions and Results:

Confusion Matrix:

SVMs:

```
array([[ 5,  3],
       [ 2, 29]], dtype=int64)
```

Deep Neural Network

```
array([[ 7,  3],
       [ 0, 29]], dtype=int64)
```

Logistic Regression

```
[[5  5]
[ 0 29]]
```

We made four models to test, compare and contrast their accuracy.

Model	Accuracy	Precision	Recall	F1 Score
SVM	87.12%	90.06%	93.50%	92.07%
DNN	92.31%	90.06%	100.00%	95.08%
XGB	94.87%	96.9%	96.9%	96.88%
Log.R	87.18%	92.06%	100.00%	92.06%

Here, we can see through the accuracy score, that the XGboost has performed better than other models. XGBoost is an ensemble learning algorithm. It has been found that XGBoost performs best when structured and tabular datasets are integrated into the model for classification problems. Although, DNN has come close to the accuracy to the XGBoost, XGBoost still has performed better because of its ensemble technique.

FUTURE SCOPE:

- Currently, the data set is small but in future the data set can be extended.
- Audio processing to extract voice components from the recordings to perform a more extensive analysis rather than using preprocessed data
- To use a bigger and more complex dataset to test model performance in real time for different samples.
- The model with XGBoost has high accuracy and could be optimized further adding more datasets to the models to have higher accuracy for use in medical fields
- We can actually find real life voice samples of people with Parkinsons and train our model with it and make the accuracy better. More data can help us in increasing the accuracy as a part of the hypertuning done.
- Also the audio preprocessing using python can also be done as a part of future scope.

REFERENCES:

Géron, Aurélien; 2019, '*Hands-On MachineLearning with Scikit-Learn, Keras, and TensorFlow*', vol.2

https://archive.ics.uci.edu/ml/datasets/parkinsons

https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

https://machinelearningmastery.com/what-is-deep-learning/

https://scikit-learn.org/stable/modules/neural_networks_supervised.html

Sakar, C.O., Serbes, G., Gunduz, A., Tunc, H.C., Nizam, H., Sakar, B.E., Tutuncu, M., Aydin, T., Isenkul, M.E. and Apaydin, H., 2018. A comparative analysis of speech signal processing algorithms for Parkinsonâ€™s disease classification and the use of the tunable Q-factor wavelet transform. Applied Soft Computing, DOI: [Web Link]