1. Objective

The overall objective is to predict if a patient has Parkinson's disease or not.

The model is to be built using data obtained from UCI at: https://archive.ics.uci.edu/ml/machine-learning-databases/00470/pd_speech_features.rar.

2. Exploration and Data Analysis

2.1. Data Description

The data has 755 columns which is made up 745 attributes and a class. The features are made up of various kinds of speech signal processing algorithms such as Time Frequency Features, Mel Frequency Cepstral Coefficients (MFCCs), Wavelet Transform based Features, Vocal Fold Features and tunable Q-factor wavelet transform (TQWT) which were applied t speech recordings of Parkinson's Disease patients so as to extract useful clinical information.

The Columns are given in Table 1. A detailed attribute information can be found at "https://archive.ics.uci.edu/ml/datasets/Parkinson%27s+Disease+Classification#."

From the paper where the data was obtained, some of the features are described as shown in Table 1. However, it is noted that features such as identity (ID), gender, tunable Q-factor wavelet transform (TQWT) and class are not included in Table 1.

Table 1: Overview of Features

Feature Set	Measure	Explanation	No of Features
Baseline Features	Jitter variants	Jitter variants are employed to capture the instabilities occurred in the oscillating pattern of the vocal folds and this feature sub-set quantifies the cycle-to-cycle changes in the fundamental frequency.	5
	Shimmer variants	Shimmer variants are also employed to capture instabilities of the oscillating pattern of the vocal folds, but this time this feature sub-set	6

		quantifies the cycle-to-cycle changes in the amplitude	
	Fundamental frequency parameters	The frequency of vocal fold vibration. Mean, median, standard deviation, minimum and maximum values were used.	5
	Harmonicity parameters	Due to incomplete vocal fold closure, increased noise components occur in speech pathologies. Harmonics to Noise Ratio and Noise to Harmonics Ratio parameters, which quantify the ratio of signal information over noise, were used as features.	2
	Recurrence Period Density Entropy (RPDE)	RPDE gives information about the ability of the vocal folds to sustain stable vocal fold oscillations and it quantifies the deviations from F_0.	1
	Detrended Fluctuation Analysis (DFA)	DFA quantifies the stochastic self-similarity of the turbulent noise.	1
	Pitch Period Entropy (PPE)	PPE measures the impaired control of fundamental frequency F0 by using logarithmic scale	1
Time Frequency Features	Intensity Parameters	Intensity is related with the power of speech signal in dB. Mean, minimum and maximum	3

		intensity values were used.	
	Formant Frequencies	Frequencies amplified by the vocal tract; the first four formants were used as features.	4
	Bandwidth	The frequency range between the formant frequencies, the first four bandwidths were employed as features	4
Mel Frequency Cepstral Coefficients (MFCCs)	MFCCs	MFCCs are employed to catch the PD affects in vocal tract separately from the vocal folds	84
Wavelet Transform based Features	Wavelet transform (WT) features related with F0	WT features quantify the deviations in F0	182
Vocal Fold Features	Glottis Quotient (GQ)	GQ gives information about opening and closing durations of the glottis. It is a measure of periodicity in glottis movements.	3
	Glottal to Noise Excitation (GNE)	GNE quantifies the extent of turbulent noise, which caused by incomplete vocal fold closure, in the speech signal.	6
	Vocal Fold Excitation Ratio (VFER)	VFER quantifies the amount of noise produced due to the pathological vocal fold vibration by using nonlinear energy and entropy concepts.	7

	Empirical Mode Decomposition (EMD)	EMD decomposes a speech signal into elementary signal components by using adaptive basis functions and energy/entropy values obtained from these components are used to quantify noise	6

The class features as values of 1's and 0's to which indicate if a patient has the Parkinson's disease or not, respectively.

3. Feature Engineering

The data information can be shown in Table 2 and the top five rows of the data is displaced in Table 3. It can be seen that the columns are unnamed and the data types of data are 'object'. A view of the top 5 rows of the data after reindex is shown in Table 4 and resetting datatypes to float is displaced in table 5.

Table 2: Information of Data Before Feature Engineering

```
<class 'pandas.core.frame.DataFrame'>
  RangeIndex: 756 entries, 0 to 755
  Columns: 755 entries, id to class
         dtypes: object(755)
       memory usage: 4.4+ MB
```

Table 3: A view of the top 5 rows of the Data before Feature Engineering

	Unnamed: 0	Unnamed: 1	Baseline Features	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnai
0	id	gender	PPE	DFA	RPDE	numPulses	numPeriodsPulses	meanPeriod
1	0	1	0.85247	0.71826	0.57227	240	239	0.008
2	0	1	0.76686	0.69481	0.53966	234	233	0.0082
3	0	1	0.85083	0.67604	0.58982	232	231	0.008
4	1	0	0.41121	0.79672	0.59257	178	177	0.0108

5 rows × 755 columns

Table 4: A view of the top 5 rows of the Data after Feature Engineering

	id	gender	PPE	DFA	RPDE	numPulses	numPeriodsPulses	meanPeriodPulses	stdDev
0	0	1	0.85247	0.71826	0.57227	240	239	0.00806353	
1	0	1	0.76686	0.69481	0.53966	234	233	0.008258256	
2	0	1	0.85083	0.67604	0.58982	232	231	0.00833959	
3	1	0	0.41121	0.79672	0.59257	178	177	0.010857733	
4	1	0	0.3279	0.79782	0.53028	236	235	0.008161574	

5 rows × 755 columns

Table 5: Information of Data After Feature Engineering

```
<class 'pandas.core.frame.DataFrame'>
  RangeIndex: 756 entries, 0 to 755
  Columns: 755 entries, id to class
        dtypes: float64(755)
        memory usage: 4.4 MB
```

4. Pre-Modelling

The class (y) is separated from the predicting features (X). Furthermore, stratified shuffle split is used to split into training (X_train;y_test) and testing set (X_test,y_test) so as to maintain equal ratio of predictors as shown in Table 6.

Table 6: Number of rows in Training and Test Sets

Train		Test	
1	0.74	1	0.74
0	0.25	0	0.25

Furthermore, the X (train and test) data is scaled to values in between 0 and 1 using the MInMAxScaler.

5. Modelling

The aim is to predict using a deep learning model.

The architecture of the first model is built using an input layer of 754 neurons (with ReLu activation), two hidden layers of 754 neurons each (with ReLu activations), and an output layer of one neuron (with sigmoid activation). Also, all layers are fully connected. Furthermore, it is compiled (using the SGD optimizer, binary-crossentropy loss and accuracy metrics) and fitted with a batch_size of 75 with 100 epochs.

The validation loss and accuracy of the model are 2.7377 and 0.0441 respectively and displaced in Figure 1.

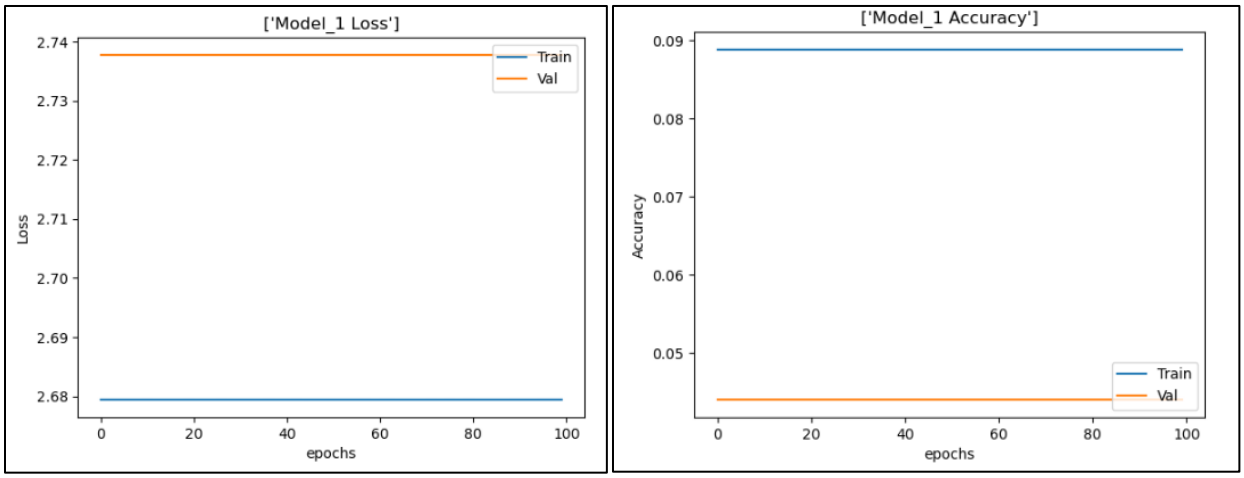

Figure 1: Loss and Accuracy of Model 1

The first model has a very poor loss and accuracy. Hence other models are built.

The second model is built similar the model 1, however with 2000 neurons in each layers, Adam optimizer and a batch size of 50. The resulting validation loss and accuracy of model 2 are 0.0900 and 0.9912 respectively and the corresponding visualization at each epoch is shown in Figure 2.

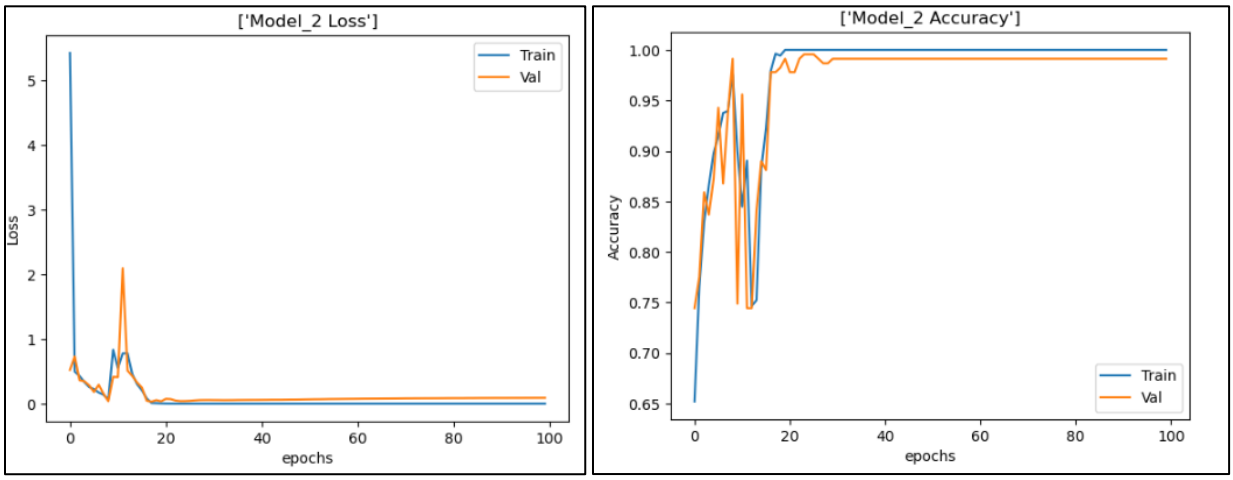

Figure 2: Loss and Accuracy of Model 2

This is an improved model with a very high accuracy of 99.12%. However, it would be interesting to see what will happen if the number of neurons is increased with the addition of dropout and L2(Ridge) regularization to the model.

Thus, the third model is built with the number of neurons increased to 3000, L2 accuracy and Dropout of 0.3 (i.e., 30% of the neurons in each layer is dropped) in each layer.

The validation loss and validation accuracy of the third model are 0.1570 and 1.0000 respectively with the visualization shown in Figure 3.

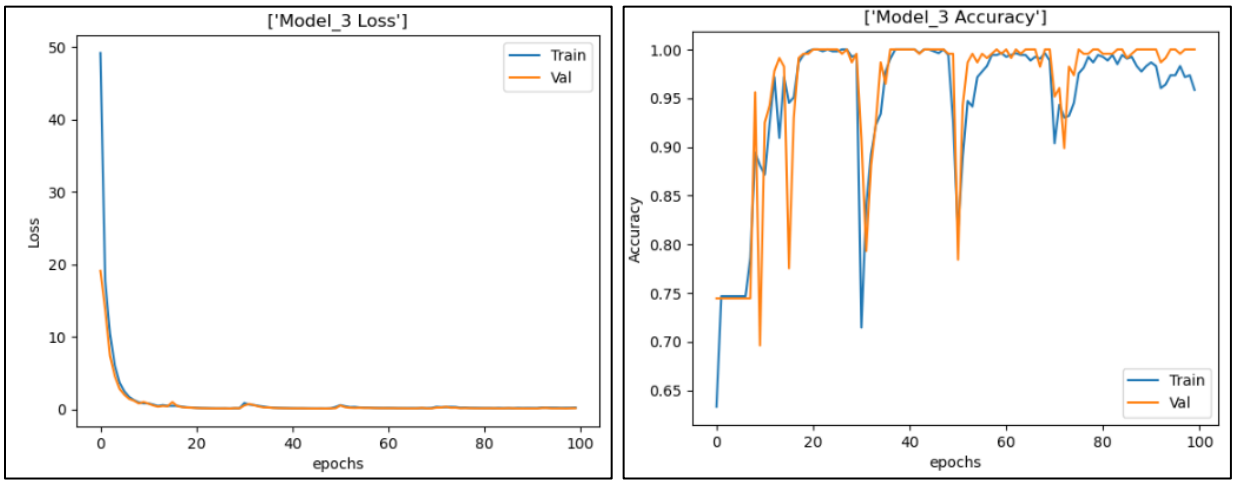

Figure 3: Accuracy and Loss of Model 3

The model 3 performed even better.

Lastly, the number of neurons is reduced to 2000 with 50 epochs.

This resulting validation loss and accuracy are 0.1637 and 0.9868 respectively and the values for each epoch is plotted in Figure 4.

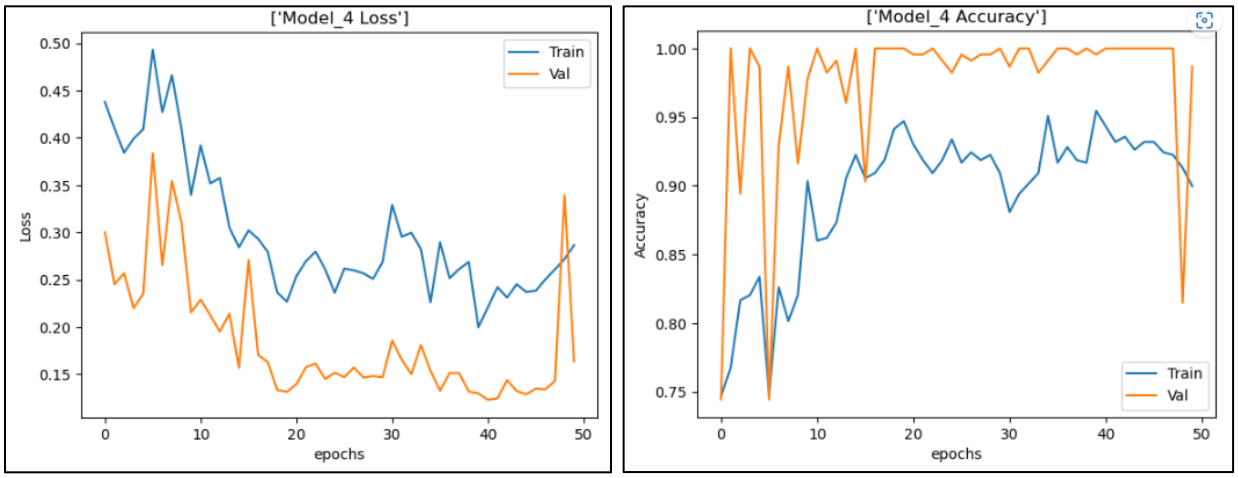

Figure 4: Accuracy and Loss of Model 4

It can be seen that the loss of the validation set is lower while and accuracy of the validation set is higher than that of the training set.

Observations/Suggestions

Table 7 gives a summary of the accuracies and loss of the model on the validation sets.

Table 6: Performance Metrics of Models Used for Prediction

Performance Metrics	Model			
	1	2	3	4
Loss	2.7377	0.0900	0.1570	0.1637
Accuracy	0.0441	0.9912	1.0000	0.9868

Model 1 is very bad at predicting. Increasing the number of neurons and regularizing with Adam optimizer increased models 2, 3 and 4 accuracies of predicting if a patient had Parkinson's disease or not with reduced losses.

Furthermore, regularization with L2 (Ridge regularization) and dropout further increased the accuracy to 1.

A look through the plots of the validation loss and accuracy shows that model 4 had a lower validation loss and a higher accuracy than the training set. This is unusual as the training loss is usually lower than those of the testing set. However, this can be put to test with a larger data in the validation set.

Another suggestion is to divide the validation set into two – validation set and test set. Then the final model chosen would be checked with the test set.

Thus, if a model is to be picked, it would be model 2 or 3 until when enough data can be used on model 4.

Succinctly, all model except model 1 achieved the objective of predicting if a patient had Parkinson's disease or not.