Parkinson’s Disease prediction from Voice signals

Created By: Mrinal Bhan & Gautam Gupta

Parkinson's disease

- Abnormalities of the Parkinson's disease speech can be associated with several dimensions
- Symptoms include impairment in the normal production of vocal sounds (dysphonia), and problems with normal articulation of speech (dysarthria)
- Dysphonic symptoms typically include
  - reduced loudness,
  - breathiness,
  - roughness,
  - decreased energy in the higher parts of the harmonic spectrum, and
  - exaggerated vocal tremor

Speech measurement for PD voice disorder:

- Traditional methods:
  - Fundamental frequency
  - Absolute sound pressure level
  - Jitter
  - Shimmer
  - Noise-to-harmonics ratios
- Novel measurement methods (based on non-linear dynamic systems):
  - Recurrence period density entropy (RPDE), D2
  - Detrended fluctuation analysis (DFA)
  - Pitch period entropy (PPE)
  - Spread 1, Spread 2 - Nonlinear measures of fundamental frequency variation

References:

1. Little MA, McSharry PE, Hunter EJ, Spielman J, Ramig LO. Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. IEEE Trans Biomed Eng. 2009 Apr;56(4):1015. doi: 10.1109/TBME.2008.2005954. PMID: 21399744; PMCID: PMC3051371.
2. Rusz J, Cmejla R, Ruzickova H, Ruzicka E. Quantitative acoustic measurements for characterization of speech and voice disorders in early untreated Parkinson's disease. J Acoust Soc Am. 2011 Jan;129(1):350-67. doi: 10.1121/1.3514381. PMID: 21303016.

Objective:

- Find the key predictors, especially speech-related characteristics, of PD

- Try at least three different machine learning approaches to PD identification and Find the best approach.

EDA

- We have 195 rows and 24 variables.
- No null values or missing values in the data
- All of the independent variables are numeric

```
data.shape
```

```
(195, 24)
```

```
data.isna().sum().sort_values
```

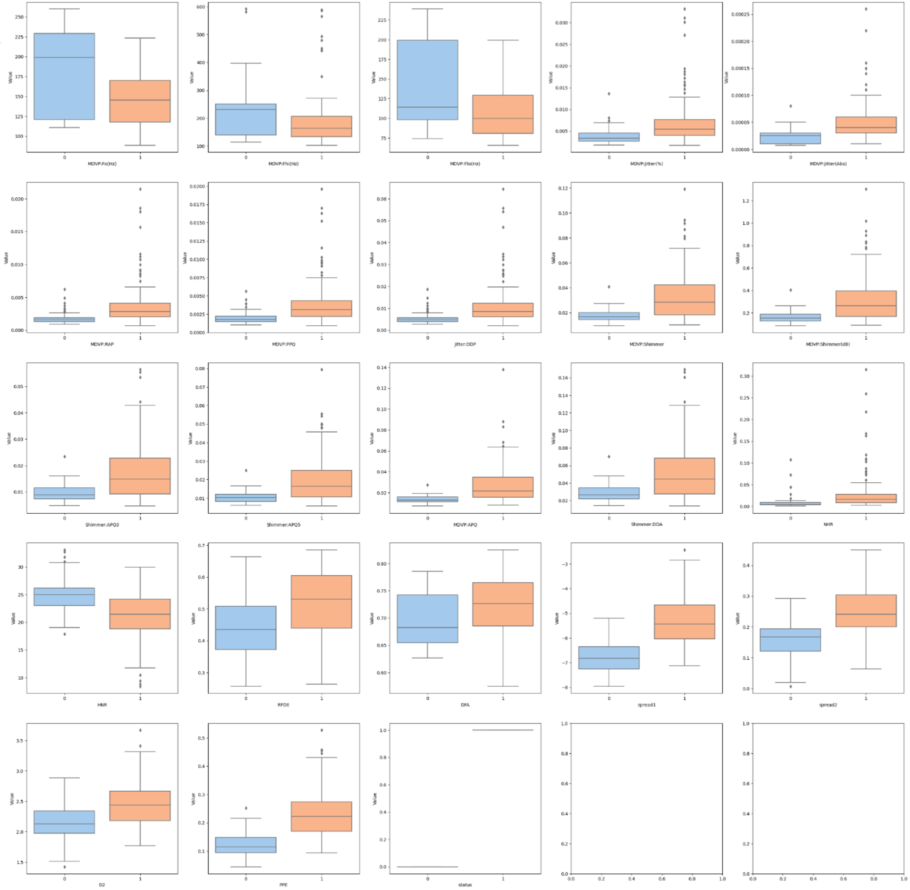
```
name                0
MDVP:Fo(Hz)         0
PPE                 0
D2                  0
spread2             0
spread1             0
DFA                 0
RPDE                0
HNR                 0
NHR                 0
Shimmer:DDA         0
MDVP:APQ            0
Shimmer:APQ5        0
Shimmer:APQ3        0
MDVP:Shimmer(dB)    0
MDVP:Shimmer        0
Jitter:DDP          0
MDVP:PPQ            0
MDVP:RAP            0
MDVP:Jitter(Abs)    0
MDVP:Jitter(%)      0
MDVP:Flo(Hz)        0
MDVP:Fhi(Hz)        0
status              0
dtype: int64
```

```
data.dtypes
```

```
name                 object
MDVP:Fo(Hz)         float64
MDVP:Fhi(Hz)        float64
MDVP:Flo(Hz)        float64
MDVP:Jitter(%)      float64
MDVP:Jitter(Abs)    float64
MDVP:RAP            float64
MDVP:PPQ            float64
Jitter:DDP          float64
MDVP:Shimmer        float64
MDVP:Shimmer(dB)    float64
Shimmer:APQ3        float64
Shimmer:APQ5        float64
MDVP:APQ            float64
Shimmer:DDA         float64
NHR                 float64
HNR                 float64
RPDE                float64
DFA                 float64
spread1             float64
spread2             float64
D2                  float64
PPE                 float64
status                int64
dtype: object
```

EDA

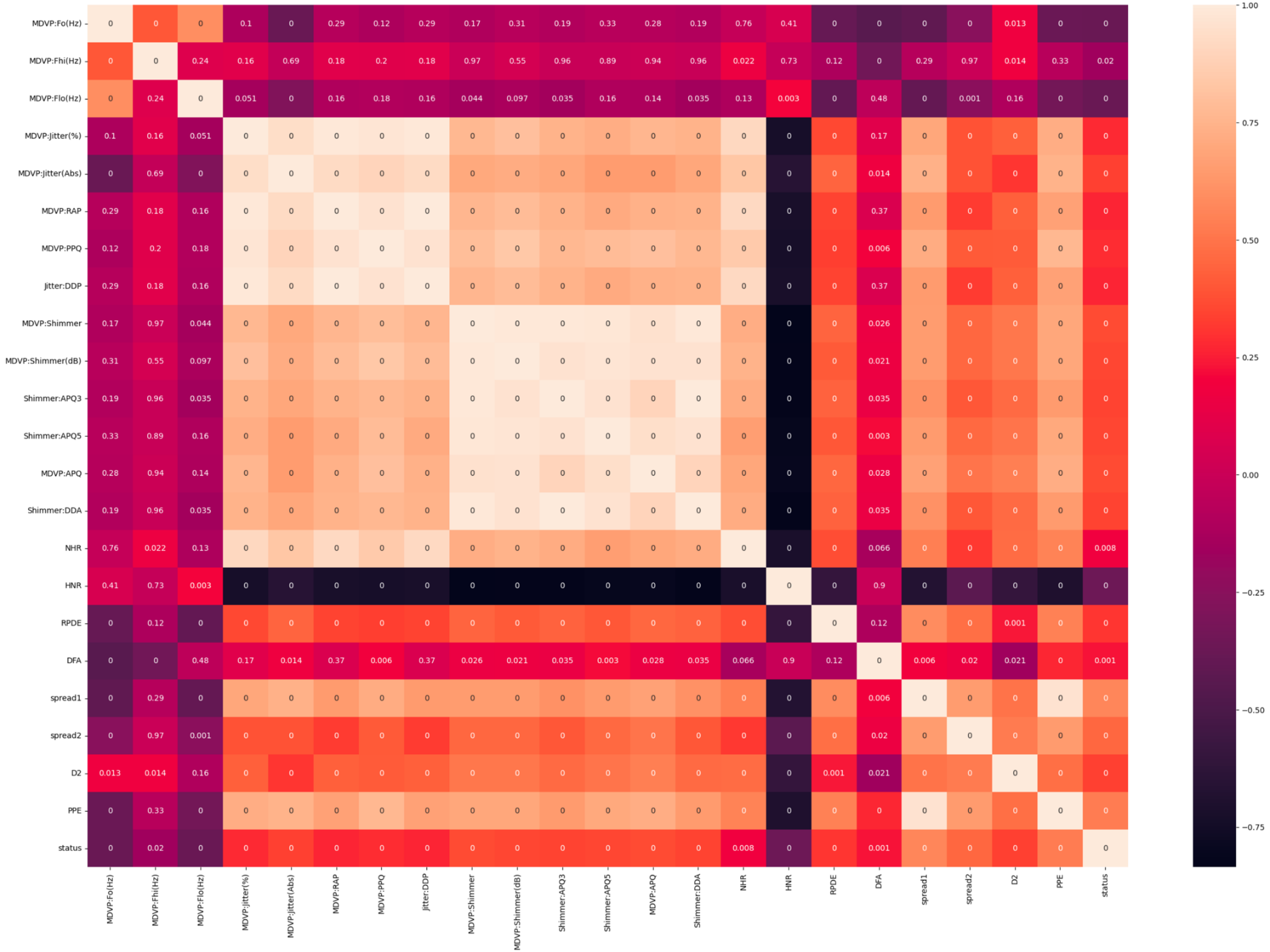
```
Number of outliers in MDVP:Fo(Hz): 0
Number of outliers in MDVP:Fhi(Hz): 11
Number of outliers in MDVP:Flo(Hz): 9
Number of outliers in MDVP:Jitter(%): 14
Number of outliers in MDVP:Jitter(Abs): 7
Number of outliers in MDVP:RAP: 14
Number of outliers in MDVP:PPQ: 15
Number of outliers in Jitter:DDP: 14
Number of outliers in MDVP:Shimmer: 8
Number of outliers in MDVP:Shimmer(dB): 10
Number of outliers in Shimmer:APQ3: 6
Number of outliers in Shimmer:APQ5: 13
Number of outliers in MDVP:APQ: 12
Number of outliers in Shimmer:DDA: 6
Number of outliers in NHR: 19
Number of outliers in HNR: 3
Number of outliers in RPDE: 0
Number of outliers in DFA: 0
Number of outliers in spread1: 4
Number of outliers in spread2: 2
Number of outliers in D2: 1
Number of outliers in PPE: 5
```



- We see that there are outliers in many features.
- But all of them are in the possible biological ranges for those variables.
- Decided to let the outliers be as is in the data,

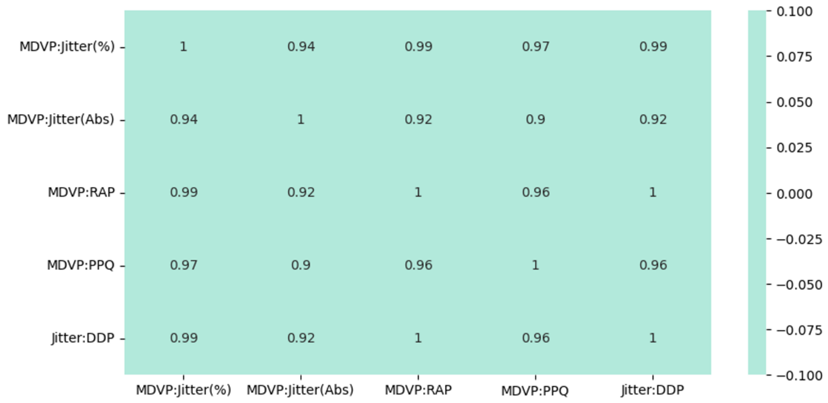
Correlation

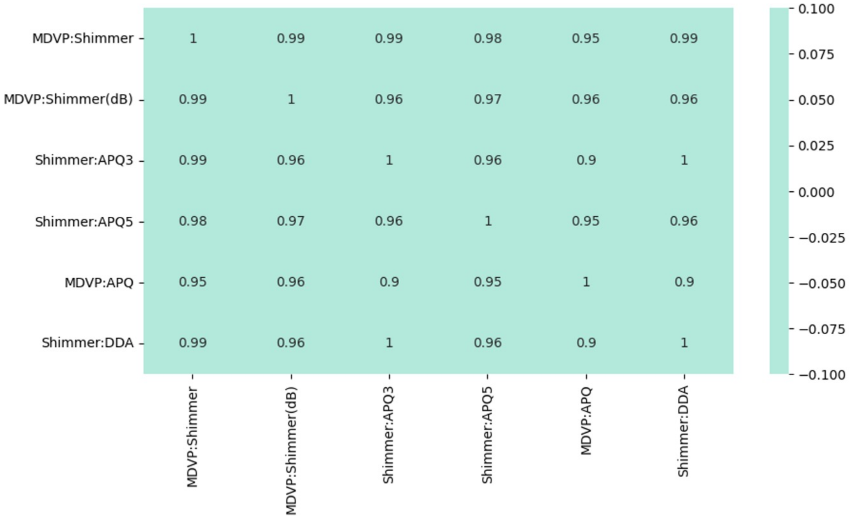
- We can see clusters of strong correlation between the variables
- For ex, Measurements of Jitter are strongly correlated with each other and a decent correlation is seen between Shimmer and Jitter variables.
- Also, there is a strong negative correlation HNR and the rest of the variables.
- The novel measurements like RPDE, DFA, PPE etc are not correlated strongly as they are non-linear measures.



Checking correlation within clusters:

- As we saw earlier, There is a strong correlation between the different measures of Jitter and similarly in the Shimmer variables.
- The information from using all the variables is limited.
- Therefore, we can drop the variables or combine them using Principal component analysis.
- We tried both approaches





Variables:

- We decided to consider the range of the Fundamental frequency instead of the Max and Min values of the Fundamental frequency based on our research.

```
data['MDVP:F_Spd'] = data['MDVP:Fhi(Hz)'] - data['MDVP:Flo(Hz)']
```

- Rest of the variables were left as is
- There 12 independent variables left in the data we can use for modelling
- The remaining independent variables in the data are:

```
df_scaled.columns
```

```
Index(['MDVP:Fo(Hz)', 'MDVP:Jitter(Abs)', 'MDVP:Shimmer', 'NHR', 'HNR', 'RPDE',
       'DFA', 'spread1', 'spread2', 'D2', 'PPE', 'MDVP:F_Spd'],
      dtype='object')
```

- First, we went with dropping the correlated variables. We also did PCA in coming up slides

Models (1):

- Normalised the data using the standard scaler

- We have built 4 different models
  a. Decision Tree
  b. Random Forest
  c. SVM
  d. k-Neural network

- The performance of the models on the test data:

	Metric	DT	RF	SVM	KNN
0	Accuracy	0.820513	0.948718	0.846154	0.897436
1	F1-Score	0.881356	0.962963	0.892857	0.928571
2	Recall	0.962963	0.962963	0.925926	0.962963
3	Precision	0.812500	0.962963	0.862069	0.896552
4	R2-Score	0.157407	0.759259	0.277778	0.518519

Principal Component Analysis:

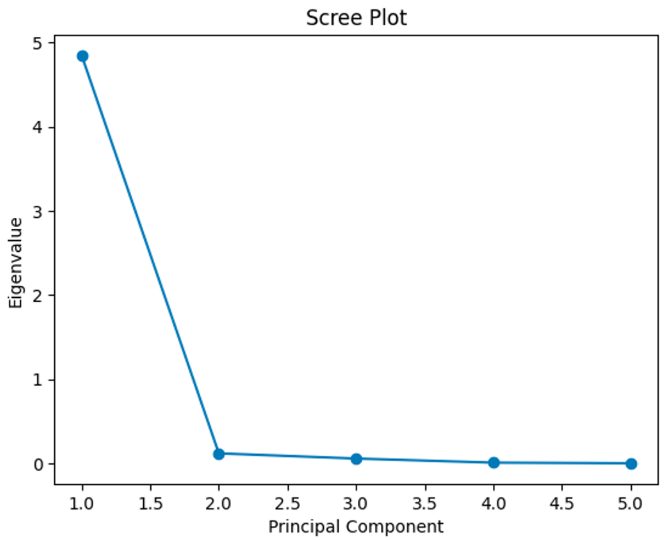
- As mentioned earlier, we have also applied PCA instead of dropping the correlated variables.
- We applied PCA on two different subsets of variables

```
variation_freq = ['MDVP:Jitter(%)', 'MDVP:Jitter(Abs)', 'MDVP:RAP', 'MDVP:PPQ', 'Jitter:DDP']
variation_amp = ['MDVP:Shimmer', 'MDVP:Shimmer(dB)','Shimmer:APQ3', 'Shimmer:APQ5', 'MDVP:APQ', 'Shimmer:DDA']
```

- Firstly, On the Frequency measures:

```
pca = PCA()
pca.fit(df_pc1)
explained_variance_ratio = pca.explained_variance_ratio_
print(explained_variance_ratio)
```

```
[9.64162746e-01 2.33046731e-02 1.11199310e-02 1.41256948e-03
 7.99702317e-08]
```



- As we can see from the scree plot, We can use one principal component to explain the five measures of Jitter.

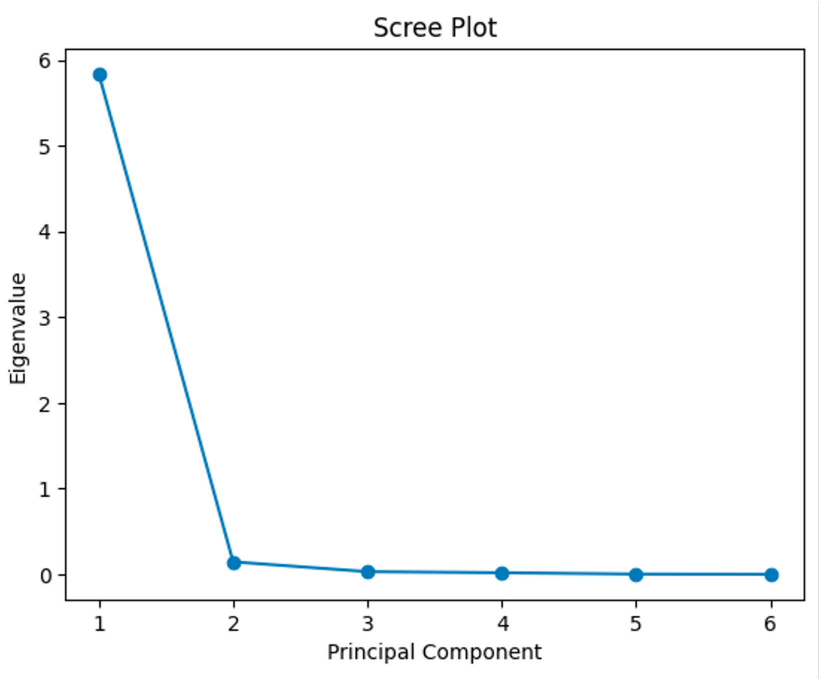
PCA (2)

- Repeating the process for the measures of Shimmer.
- We arrive at the same conclusion
- One principal component to explain the various measures of Shimmer

```
pca2 = PCA()
df_pc2 = df_pc[variation_amp]
df_pc2 = scaler.fit_transform(df_pc2)

pca2.fit(df_pc2)
explained_variance_ratio = pca2.explained_variance_ratio_
print(explained_variance_ratio)
```

```
[9.67891293e-01 2.39668230e-02 4.97378202e-03 2.88262892e-03
 2.85467090e-04 6.01341259e-09]
```



PCA Final:

- Add these two new principal component columns to the original dataset:

	MDVP:Fo(Hz)	NHR	HNR	RPDE	DFA	spread1	spread2	D2	PPE	status	MDVP:F_Spd	Jitter_pc	Shimmer_pc
0	119.992	0.02211	21.033	0.414783	0.815285	-4.813031	0.266482	2.301442	0.284654	1	82.305	0.934818	1.700278
1	122.400	0.01929	19.085	0.458359	0.819521	-4.075192	0.335590	2.486855	0.368674	1	34.831	1.751692	4.081638
2	116.682	0.01309	20.651	0.429895	0.825288	-4.443179	0.311173	2.342259	0.332634	1	19.556	2.333009	2.865896
3	116.676	0.01353	20.644	0.434969	0.819235	-4.117501	0.334147	2.405554	0.368975	1	26.505	2.020036	3.224675
4	116.014	0.01767	19.649	0.417356	0.823484	-3.747787	0.234513	2.332180	0.410335	1	31.126	3.346368	4.471558
5	120.552	0.01222	21.378	0.415564	0.825069	-4.242867	0.299111	2.187560	0.357775	1	17.375	1.832739	2.152991
6	120.267	0.00607	24.886	0.596040	0.764112	-5.634322	0.257682	1.854785	0.211756	1	22.424	-1.211720	-1.775359

Models (2):

- The same 4 models are created on the new data.
- The performance of the models in the new test data:

	Metric	DT	RF	SVM	KNN
0	Accuracy	0.820513	0.846154	0.820513	0.897436
1	F1-Score	0.877193	0.888889	0.877193	0.925926
2	Recall	0.925926	0.888889	0.925926	0.925926
3	Precision	0.833333	0.888889	0.833333	0.925926
4	R2-Score	0.157407	0.277778	0.157407	0.518519

Comparisons:

Model without PCA

	Metric	DT	RF	SVM	KNN
0	Accuracy	0.820513	0.948718	0.846154	0.897436
1	F1-Score	0.881356	0.962963	0.892857	0.928571
2	Recall	0.962963	0.962963	0.925926	0.962963
3	Precision	0.812500	0.962963	0.862069	0.896552
4	R2-Score	0.157407	0.759259	0.277778	0.518519

Model with PCA

	Metric	DT	RF	SVM	KNN
0	Accuracy	0.820513	0.846154	0.820513	0.897436
1	F1-Score	0.877193	0.888889	0.877193	0.925926
2	Recall	0.925926	0.888889	0.925926	0.925926
3	Precision	0.833333	0.888889	0.833333	0.925926
4	R2-Score	0.157407	0.277778	0.157407	0.518519

Feature importance:

- Feature importance from the Random forest model on the non-pca data.
- **PPE** is the most important followed by **spread1** and **MDVP:Fo(Hz)**

```
# Get feature importances
importances = rfc1.feature_importances_

# Print feature importances
for feature, importance in zip(X.columns, importances):
    print(f"{feature}: {importance}")
```

```
MDVP:Fo(Hz): 0.1320668098294375
MDVP:Jitter(Abs): 0.04023009691589492
MDVP:Shimmer: 0.1249343949515893l
NHR: 0.06507400866224698
HNR: 0.05291217867055756
RPDE: 0.05459951046943420
DFA: 0.04843645923393191
spread1: 0.13853025374107053
spread2: 0.08964864860571996
D2: 0.05462080274261521
PPE: 0.1443170982033454
MDVP:F_Spd: 0.0546297379741566
```

- Feature importance from the Random forest model on the pca data.
- We see the same variables showing up as important.

```
# Get feature importances
importances = rfc1.feature_importances_

# Print feature importances
for feature, importance in zip(X.columns, importances):
    print(f"{feature}: {importance}")
```

```
MDVP:Fo(Hz): 0.14001260488596276
NHR: 0.04838287411867583
HNR: 0.04510029096131943
RPDE: 0.04544629165275420
DFA: 0.04897296662768114
spread1: 0.172414809901646
spread2: 0.08367074620695798
D2: 0.05929651458774371
PPE: 0.1851486311305981
MDVP:F_Spd: 0.05345051076479391
Jitter_pc: 0.04580686727192774
Shimmer_pc: 0.07229689188993882
```