Current Trends in Biotechnology and Pharmacy
Vol. 17(2) 808 - 818, April 2023, ISSN 0973-8916 (Print), 2230-7303 (Online)
DOI: 10.5530/ctbp.2023.2.19

Parkinson's disease Detection Using Tree Based Machine Learning Algorithms

Venkata Srinivas Babu Oguri[1] , Sudhakar Poda[2] , A. Krishna Satya[*2] , NK Prasanna

[1]Department of Computer Science and Engineering, Mahindra University, Bahadurpally Jeedimetla, Hyderabad - 500043 - Telangana, INDIA
[2]Department of Biotechnology, Acharya Nagarjuna University, Nagarjuna nagar, Guntur-522510, Andhra Pradesh, India
[3]CSIR- National Institute of Science Communication & Policy Research, New Delhi-110 012, Delhi, India

Abstract

Parkinson's Disease (PD), also known as primary Parkinsonism is a persistent, idiopathic, degenerative nervous disorder which results from lack of dopaminergic neurons in the substantia nigra pars compacta, which is the source of nigrostriatal dopamine pathway within the midbrain. The clinical detection relies on motor symptoms recognition. Significant neurological damage is already done by the time motor symptom occur. Early detection is necessary to stalk the progression of the disease. The problem of detection of PD comes under classification. Several tree-based classification algorithms were applied to the dataset retrieved from UCI machine learning database. The dataset was first split into train and test data. Various models were created using four different algorithms. Correlation coefficients were calculated for each of the features in the dataset. The model was fitted with train data obtained after removing highly correlated features. Predictions were made and various parameters were considered for comparison. Accuracy, precision, recall, F1-Score, Youden Index, error rate and specificity were the parameters calculated. Out of the four algorithms (Decision Tree, Random Forest, XGBoost and LightGBM), LightGBM achieved the highest accuracy of 97.43%.

Keywords: Parkinson's Disease (PD), LightGBM, Pearson Correlation, Accuracy, Error Rate, Jupyter Notebook.

Introduction

Parkinson's Disease (PD) also known as primary Parkinsonism is a persistent, idiopathic, degenerative nervous disorder which results from lack of dopaminergic neurons in the substantia nigra pars compacta, which is responsible for nigrostriatal dopamine pathway within the midbrain (1). Some of the symptoms of PD include bradykinesia, rigidity, and rest tremor. Resting tremor (initially unilateral), rigidness, bradykinesia (slow movements), dissimilarities in gait, and unstable posture come under the motor symptoms while cognitive changes, behavioural and neuropsychiatric changes, autonomic nervous system failure, sensory and sleep disturbances come under non-motor symptoms (1). Motor and non-motor symptoms are used to diagnose PD. Non-motor issues of the disease can become more troublesome as the disease progresses (2). Also, voice and speech impairment typically occur in PD patients. The loss of ability to communicate properly is the main source of disability in patients. The multidimensional irregularities in the speech such as hoarse voice, reduced loudness, and restricted pitch variability (Mono pitch and Mono loudness), imprecise articulation and abnormalities of speech rate, and pause ratio can be attributed to the loss of dopaminergic neurons (3). Voice and speech performance will show further deterioration in the course of time which hints at nondopaminergic mechanisms of progression of dysarthrophonia. Early detection reduces the disease progression and limits the treatment expenses. Several machine learning algorithms can be used for the detection of PD in preliminary stages using voice data. Machine learning algorithms have made commendable progress in medical diagnosis in the recent times because of their ease in implementation. The current study aims to utilise the MLalgorithms to facilitate early detection of the disease. A total of four ML algorithms were used in this study. They are Decision Tree, Random Forest, XGBoost and LightGBM classifiers.

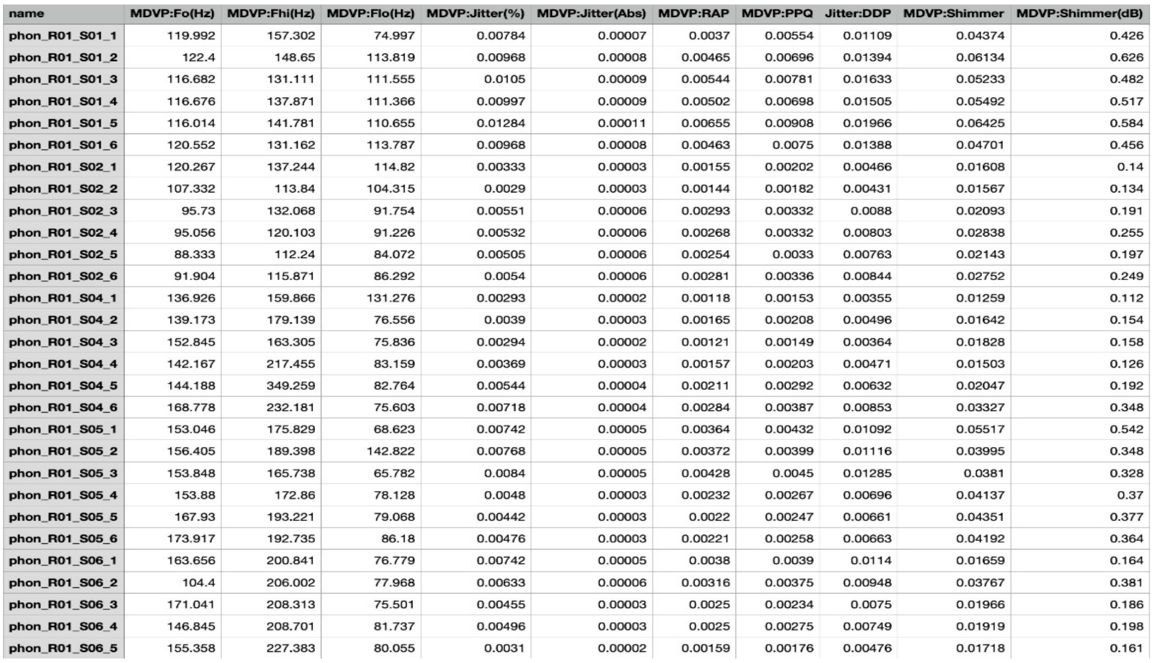
Review of Literature

10 million people (about half the population of New York) worldwide have PD from the information found in Parkinson's diseases foundation (2015). Death and disability due to PD is increasing faster than any other neurological disease according to WHO. One in every 500 people have PD in Britain and this number is expected to grow threefold by 2050 according to Parkinson's Disease Society website. This illness effects people from 50 -70 years old and becomes worse over time. Diagnosis of PD is heavily reliant on evaluation of motion which is difficult to detect by human sight. This method aims to overcome these difficulties and improve the assessment process by employing machine learning algorithms (6). One attempt was made by implementing Convolutional Neural Networks (CNNs). They were used to classify gait signals converted to spectrogram images by image classification on a big-scale and deep dense Artificial Neural Networks (ANNs) were employed to predict PD at an early stage. Voice recordings were used in this instance. A total of 54 studies in the category 'Diagnosis of PD' were examined. Out of them 33 studies used datasets from UCI machine learning repository, mPower and PhysioNet databases. In one of the study, data from public repositories was joined with local data bases (7). 14 studies performed diagnosis as well as differential diagnosis. Research articles not written in English were not considered. Most commonly voice data was used, while some studies also used MRI, movement, handwriting patterns and SPECT imagining data. The most common metric used for assessment of performance was accuracy. Most methods are based on speech data (8), gait patterns (9), cardiovascular oscillations (10), smell identification (11) and force tracking data (12). A one- dimensional neural network relying on signals of gait was introduced to detect PD in (13). It should be noted that accuracy is low when using gait analyses because of background noise in voice recordings, causing false positives. Detection of motor impairment based on mobile screen typing was introduced in (14). Four classifiers, namely Decision Tree, Regression, DMneural and Neural Networks (NN) are used and their performances are compared in (15), in which the best accuracy of 92.90% was achieved by NN algorithm. Early and accurate detection of PD is essential to stalk the progression of the disease.

Materials and Methods

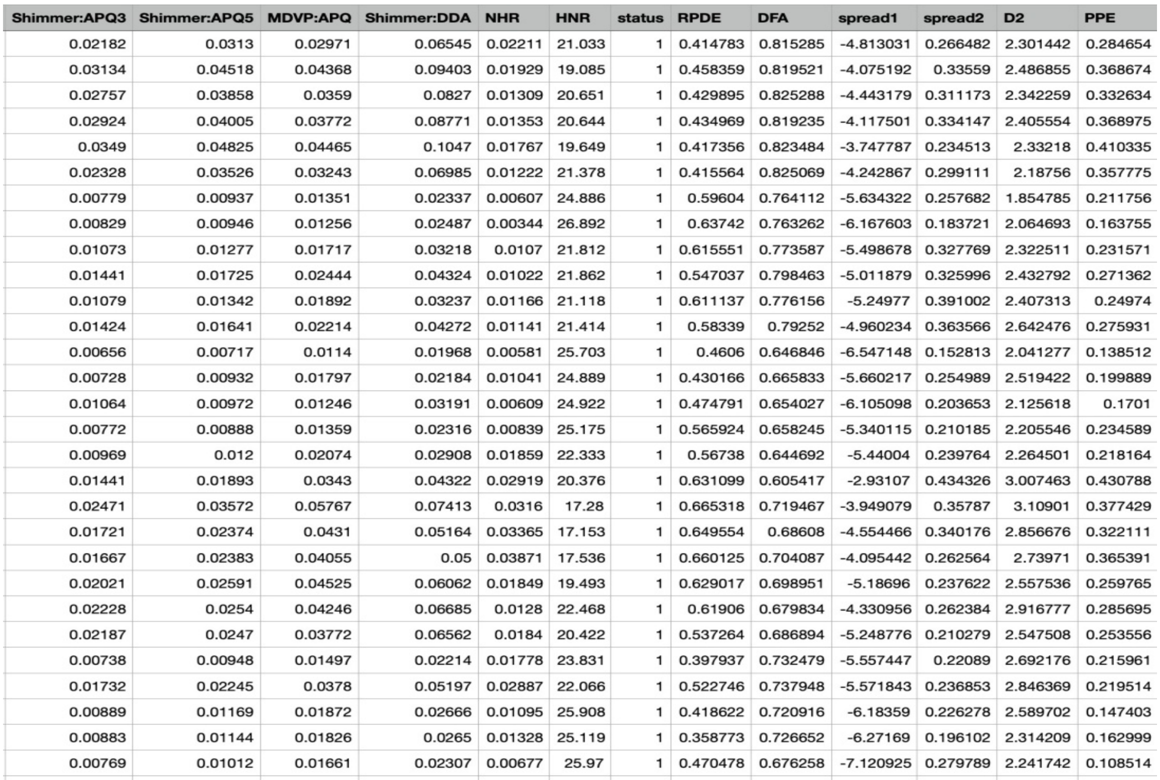
The dataset was retrieved from UCI machine learning repository. It was created by University of Oxford and National Centre for Voice and Speech, Denver, Colorado. The dataset contains voice measurements from 31 people, and 23 with PD. It has a total of 195voice recordings (4). The data aims to discriminate PD people from healthy people. The status column denotes '0' for healthy and '1' for PD affected persons. There are about 5-6recordings for each patient.

Fig 1 and Fig 2 show a section of the dataset. The problem of diagnosis of PD comesunder classification. Classification algorithms come under supervised learning. Severalclas-



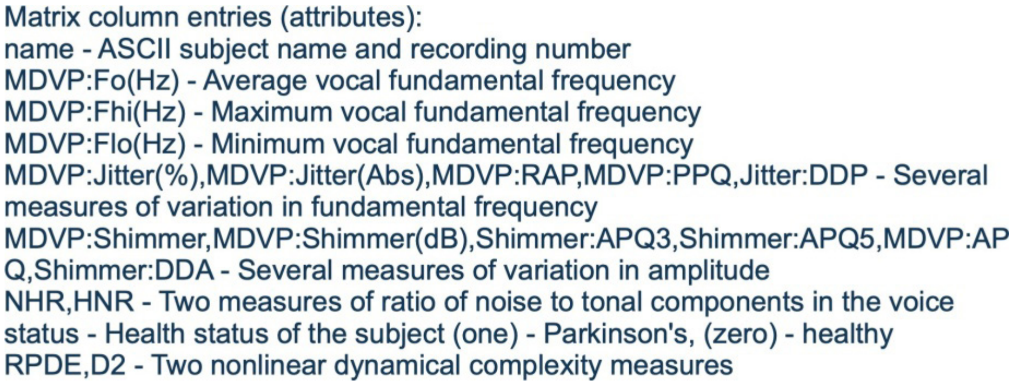
name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	MDVP:Shimmer(dB)
phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.0037	0.00554	0.01109	0.04374	0.426
phon_R01_S01_2	122.4	148.65	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	0.626
phon_R01_S01_3	116.682	131.111	111.555	0.0105	0.00009	0.00544	0.00781	0.01633	0.05233	0.482
phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	0.517
phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	0.584
phon_R01_S01_6	120.552	131.162	113.787	0.00968	0.00008	0.00463	0.0075	0.01388	0.04701	0.456
phon_R01_S02_1	120.267	137.244	114.82	0.00333	0.00003	0.00155	0.00202	0.00466	0.01608	0.14
phon_R01_S02_2	107.332	113.84	104.315	0.0029	0.00003	0.00144	0.00182	0.00431	0.01567	0.134
phon_R01_S02_3	95.73	132.068	91.754	0.00551	0.00006	0.00293	0.00332	0.0088	0.02093	0.191
phon_R01_S02_4	95.056	120.103	91.226	0.00532	0.00006	0.00268	0.00332	0.00803	0.02838	0.255
phon_R01_S02_5	88.333	112.24	84.072	0.00505	0.00006	0.00254	0.0033	0.00763	0.02143	0.197
phon_R01_S02_6	91.904	115.871	86.292	0.0054	0.00006	0.00281	0.00336	0.00844	0.02752	0.249
phon_R01_S04_1	136.926	159.866	131.276	0.00293	0.00002	0.00118	0.00153	0.00355	0.01259	0.112
phon_R01_S04_2	139.173	179.139	76.556	0.0039	0.00003	0.00165	0.00208	0.00496	0.01642	0.154
phon_R01_S04_3	152.845	163.305	75.836	0.00294	0.00002	0.00121	0.00149	0.00364	0.01828	0.158
phon_R01_S04_4	142.167	217.455	83.159	0.00369	0.00003	0.00157	0.00203	0.00471	0.01503	0.126
phon_R01_S04_5	144.188	349.259	82.764	0.00544	0.00004	0.00211	0.00292	0.00632	0.02047	0.192
phon_R01_S04_6	168.778	232.181	75.603	0.00718	0.00004	0.00284	0.00387	0.00853	0.03327	0.348
phon_R01_S05_1	153.046	175.829	68.623	0.00742	0.00005	0.00364	0.00432	0.01092	0.05517	0.542
phon_R01_S05_2	156.405	189.398	142.822	0.00768	0.00005	0.00372	0.00399	0.01116	0.03995	0.348
phon_R01_S05_3	153.848	165.738	65.782	0.0084	0.00005	0.00428	0.0045	0.01285	0.0381	0.328
phon_R01_S05_4	153.88	172.86	78.128	0.0048	0.00003	0.00232	0.00267	0.00696	0.04137	0.37
phon_R01_S05_5	167.93	193.221	79.068	0.00442	0.00003	0.0022	0.00247	0.00661	0.04351	0.377
phon_R01_S05_6	173.917	192.735	86.18	0.00476	0.00003	0.00221	0.00258	0.00663	0.04192	0.364
phon_R01_S06_1	163.656	200.841	76.779	0.00742	0.00005	0.0038	0.0039	0.0114	0.01659	0.164
phon_R01_S06_2	104.4	206.002	77.968	0.00633	0.00006	0.00316	0.00375	0.00948	0.03767	0.381
phon_R01_S06_3	171.041	208.313	75.501	0.00455	0.00003	0.0025	0.00234	0.0075	0.01966	0.186
phon_R01_S06_4	146.845	208.701	81.737	0.00496	0.00003	0.0025	0.00275	0.00749	0.01919	0.198
phon_R01_S06_5	155.358	227.383	80.055	0.0031	0.00002	0.00159	0.00176	0.00476	0.01718	0.161

Fig1: Dataset Part 1

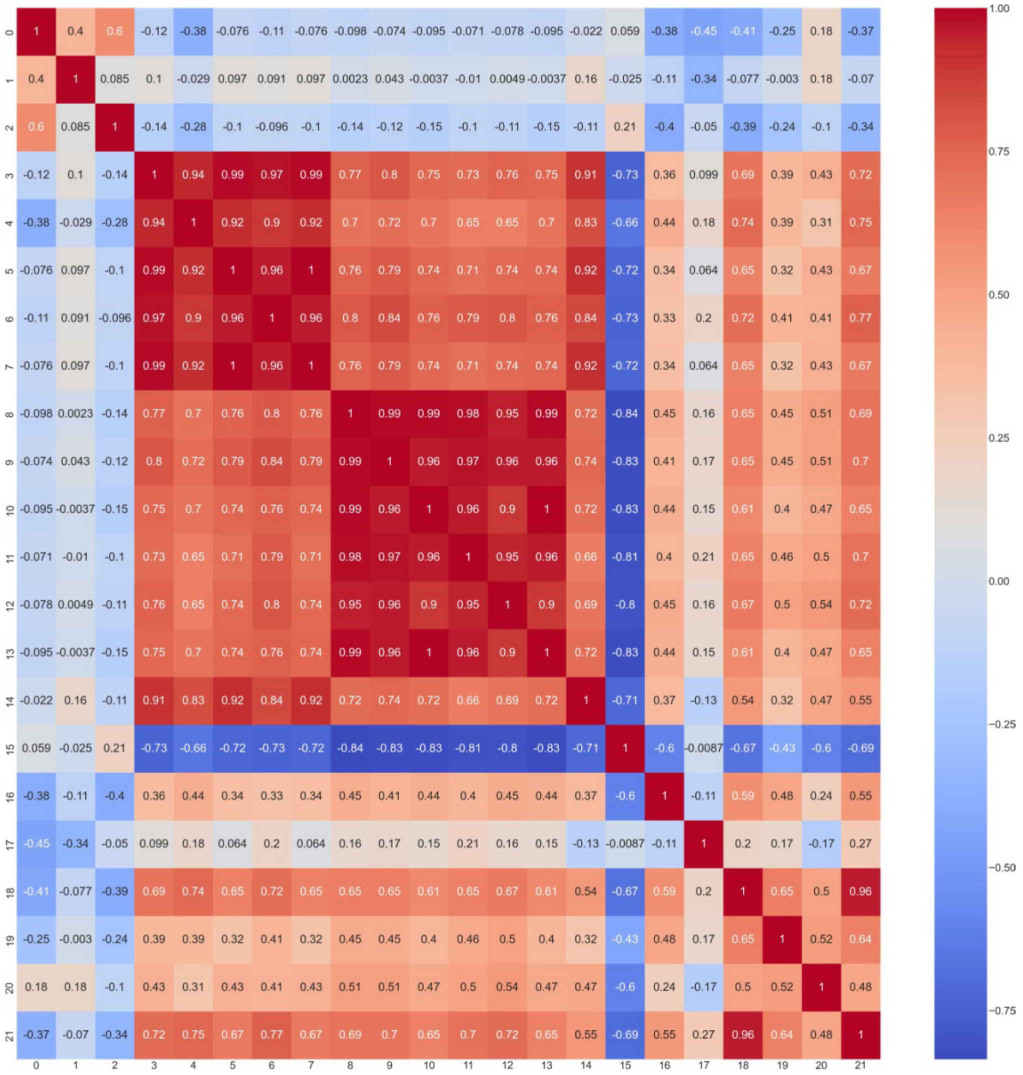


Shimmer:APQ3	Shimmer:APQ5	MDVP:APQ	Shimmer:DDA	NHR	HNR	status	RPDE	DFA	spread1	spread2	D2	PPE
0.02182	0.0313	0.02971	0.06545	0.02211	21.033	1	0.414783	0.815285	-4.813031	0.266482	2.301442	0.284654
0.03134	0.04518	0.04368	0.09403	0.01929	19.085	1	0.458359	0.819521	-4.075192	0.33559	2.486855	0.368674
0.02757	0.03858	0.0359	0.0827	0.01309	20.651	1	0.429895	0.825288	-4.443179	0.311173	2.342259	0.332634
0.02924	0.04005	0.03772	0.08771	0.01353	20.644	1	0.434969	0.819235	-4.117501	0.334147	2.405554	0.368975
0.0349	0.04825	0.04465	0.1047	0.01767	19.649	1	0.417356	0.823484	-3.747787	0.234513	2.33218	0.410335
0.02328	0.03526	0.03243	0.06985	0.01222	21.378	1	0.415564	0.825069	-4.242867	0.299111	2.18756	0.357775
0.00779	0.00937	0.01351	0.02337	0.00607	24.886	1	0.59604	0.764112	-5.634322	0.257682	1.854785	0.211756
0.00829	0.00946	0.01256	0.02487	0.00344	26.892	1	0.63742	0.763262	-6.167603	0.183721	2.064693	0.163755
0.01073	0.01277	0.01717	0.03218	0.0107	21.812	1	0.615551	0.773587	-5.498678	0.327769	2.322511	0.231571
0.01441	0.01725	0.02444	0.04324	0.01022	21.862	1	0.547037	0.798463	-5.011879	0.325996	2.432792	0.271362
0.01079	0.01342	0.01892	0.03237	0.01166	21.118	1	0.611137	0.776156	-5.24977	0.391002	2.407313	0.24974
0.01424	0.01641	0.02214	0.04272	0.01141	21.414	1	0.58339	0.79252	-4.960234	0.363566	2.642476	0.275931
0.00656	0.00717	0.0114	0.01968	0.00581	25.703	1	0.4606	0.646846	-6.547148	0.152813	2.041277	0.138512
0.00728	0.00932	0.01797	0.02184	0.01041	24.889	1	0.430166	0.665833	-5.660217	0.254989	2.519422	0.199889
0.01064	0.00972	0.01246	0.03191	0.00609	24.922	1	0.474791	0.654027	-6.105098	0.203653	2.125618	0.1701
0.00772	0.00888	0.01359	0.02316	0.00839	25.175	1	0.565924	0.658245	-5.340115	0.210185	2.205546	0.234589
0.00969	0.012	0.02074	0.02908	0.01859	22.333	1	0.56738	0.644692	-5.44004	0.239764	2.264501	0.218164
0.01441	0.01893	0.0343	0.04322	0.02919	20.376	1	0.631099	0.605417	-2.93107	0.434326	3.007463	0.430788
0.02471	0.03572	0.05767	0.07413	0.0316	17.28	1	0.665318	0.719467	-3.949079	0.35787	3.10901	0.377429
0.01721	0.02374	0.0431	0.05164	0.03365	17.153	1	0.649554	0.68608	-4.554466	0.340176	2.856676	0.322111
0.01667	0.02383	0.04055	0.05	0.03871	17.536	1	0.660125	0.704087	-4.095442	0.262564	2.73971	0.365391
0.02021	0.02591	0.04525	0.06062	0.01849	19.493	1	0.629017	0.698951	-5.18696	0.237622	2.557536	0.259765
0.02228	0.0254	0.04246	0.06685	0.0128	22.468	1	0.61906	0.679834	-4.330956	0.262384	2.916777	0.285695
0.02187	0.0247	0.03772	0.06562	0.0184	20.422	1	0.537264	0.686894	-5.248776	0.210279	2.547508	0.253556
0.00738	0.00948	0.01497	0.02214	0.01778	23.831	1	0.397937	0.732479	-5.557447	0.22089	2.692176	0.215961
0.01732	0.02245	0.0378	0.05197	0.02887	22.066	1	0.522746	0.737948	-5.571843	0.236853	2.846369	0.219514
0.00889	0.01169	0.01872	0.02666	0.01095	25.908	1	0.418622	0.720916	-6.18359	0.226278	2.589702	0.147403
0.00883	0.01144	0.01826	0.0265	0.01328	25.119	1	0.358773	0.726652	-6.27169	0.196102	2.314209	0.162999
0.00769	0.01012	0.01661	0.02307	0.00677	25.97	1	0.470478	0.676258	-7.120925	0.279789	2.241742	0.108514

Fig 2: Dataset Part 2



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 ratio of noise to tonal components in the voice
status - Health status of the subject (one) - Parkinson's, (zero) - healthy
RPDE,D2 - Two nonlinear dynamical complexity measures



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	1	0.4	0.6	-0.12	-0.38	-0.076	-0.11	-0.076	-0.098	-0.074	-0.095	-0.071	-0.078	-0.095	-0.022	0.059	-0.38	-0.45	-0.41	-0.25	0.18	-0.37
1	0.4	1	0.085	0.1	-0.029	0.097	0.091	0.097	0.0023	0.043	-0.0037	-0.01	0.0049	-0.0037	0.16	-0.025	-0.11	-0.34	-0.077	-0.003	0.18	-0.07
2	0.6	0.085	1	-0.14	-0.28	-0.1	-0.096	-0.1	-0.14	-0.12	-0.15	-0.1	-0.11	-0.15	-0.11	0.21	-0.4	-0.05	-0.39	-0.24	-0.1	-0.34
3	-0.12	0.1	-0.14	1	0.94	0.99	0.97	0.99	0.77	0.8	0.75	0.73	0.76	0.75	0.91	-0.73	0.36	0.099	0.69	0.39	0.43	0.72
4	-0.38	-0.029	-0.28	0.94	1	0.92	0.9	0.92	0.7	0.72	0.7	0.65	0.65	0.7	0.83	-0.66	0.44	0.18	0.74	0.39	0.31	0.75
5	-0.076	0.097	-0.1	0.99	0.92	1	0.96	1	0.76	0.79	0.74	0.71	0.74	0.74	0.92	-0.72	0.34	0.064	0.65	0.32	0.43	0.67
6	-0.11	0.091	-0.096	0.97	0.9	0.96	1	0.96	0.8	0.84	0.76	0.79	0.8	0.76	0.84	-0.73	0.33	0.2	0.72	0.41	0.41	0.77
7	-0.076	0.097	-0.1	0.99	0.92	1	0.96	1	0.76	0.79	0.74	0.71	0.74	0.74	0.92	-0.72	0.34	0.064	0.65	0.32	0.43	0.67
8	-0.098	0.0023	-0.14	0.77	0.7	0.76	0.8	0.76	1	0.99	0.99	0.98	0.95	0.99	0.72	-0.84	0.45	0.16	0.65	0.45	0.51	0.69
9	-0.074	0.043	-0.12	0.8	0.72	0.79	0.84	0.79	0.99	1	0.96	0.97	0.96	0.96	0.74	-0.83	0.41	0.17	0.65	0.45	0.51	0.7
10	-0.095	-0.0037	-0.15	0.75	0.7	0.74	0.76	0.74	0.99	0.96	1	0.96	0.9	1	0.72	-0.83	0.44	0.15	0.61	0.4	0.47	0.65
11	-0.071	-0.01	-0.1	0.73	0.65	0.71	0.79	0.71	0.98	0.97	0.96	1	0.95	0.96	0.66	-0.81	0.4	0.21	0.65	0.46	0.5	0.7
12	-0.078	0.0049	-0.11	0.76	0.65	0.74	0.8	0.74	0.95	0.96	0.9	0.95	1	0.9	0.69	-0.8	0.45	0.16	0.67	0.5	0.54	0.72
13	-0.095	-0.0037	-0.15	0.75	0.7	0.74	0.76	0.74	0.99	0.96	1	0.96	0.9	1	0.72	-0.83	0.44	0.15	0.61	0.4	0.47	0.65
14	-0.022	0.16	-0.11	0.91	0.83	0.92	0.84	0.92	0.72	0.74	0.72	0.66	0.69	0.72	1	-0.71	0.37	-0.13	0.54	0.32	0.47	0.55
15	0.059	-0.025	0.21	-0.73	-0.66	-0.72	-0.73	-0.72	-0.84	-0.83	-0.83	-0.81	-0.8	-0.83	-0.71	1	-0.6	-0.0087	-0.67	-0.43	-0.6	-0.69
16	-0.38	-0.11	-0.4	0.36	0.44	0.34	0.33	0.34	0.45	0.41	0.44	0.4	0.45	0.44	0.37	-0.6	1	-0.11	0.59	0.48	0.24	0.55
17	-0.45	-0.34	-0.05	0.099	0.18	0.064	0.2	0.064	0.16	0.17	0.15	0.21	0.16	0.15	-0.13	-0.0087	-0.11	1	0.2	0.17	-0.17	0.27
18	-0.41	-0.077	-0.39	0.69	0.74	0.65	0.72	0.65	0.65	0.65	0.61	0.65	0.67	0.61	0.54	-0.67	0.59	0.2	1	0.65	0.5	0.96
19	-0.25	-0.003	-0.24	0.39	0.39	0.32	0.41	0.32	0.45	0.45	0.4	0.46	0.5	0.4	0.32	-0.43	0.48	0.17	0.65	1	0.52	0.64
20	0.18	0.18	-0.1	0.43	0.31	0.43	0.41	0.43	0.51	0.51	0.47	0.5	0.54	0.47	0.47	-0.6	0.24	-0.17	0.5	0.52	1	0.48
21	-0.37	-0.07	-0.34	0.72	0.75	0.67	0.77	0.67	0.69	0.7	0.65	0.7	0.72	0.65	0.55	-0.69	0.55	0.27	0.96	0.64	0.48	1

Fig3: Pearson Correlation Heat-map

sification algorithms were applied to the dataset. All of them were tree based. DecisionTree, Random Forest algorithms later followed by LightGBM and XGBoost were used toachieve highest accuracy. The reasons behind using tree-based classification algorithmswere that they mimic human thinking ability and they can be easily understood [5]. Thedataset underwent preprocessing initially. All the work was done in Jupyter Notebook, aninteractive python notebook software. Name column(attribute) was removed since it isirrelevant and decreases the accuracy. Next, correlation matrix was generated. Correlationmatrix is square and symmetric [16]. It measures the linear dependency between twoelements. We were specifically using Pearson's correlation coefficient. The coefficient canbe equal to any number between -1 and 1. perfectly negatively linear correlated variableshave coefficient equal to -1, highly correlated variables have a correlation of value 1 and 0indicates no linear correlation between the two variables. After the removal of the Nameattributefromthedataset,correlationmatrixisgeneratedfortheremaining22features. The matrix is shown in Fig 3. The matrix was color coded as 'cool warm' to easilyunderstand the strength of relationship. The stronger relations have warmer(red) colorgrading while the weak ones have cool(blue) color grading. All the diagonal elements willbe red in color and have correlation coefficient value of 1(since each attribute is mappedto itself). The threshold value for removal is set to 0.9. Out of 22 features, 11 wereremoved.Newdatasetwascreatedafterremovalofhighlycorrelatedfeatures.Thisdataset was split into training and testing datasets. 30% was randomly set aside for testingwhile the remaining was used to train the model. Two Tree based algorithms and twoBoosting Algorithms were used. Brief descriptions of the algorithms used in this work aregivenbelow.

Decision tree classifier

Decision Tree algorithm breaks a complex problem into a set of decisions which arerelatively simpler. Every Decision Tree contains a Root Node, Leaf Nodes and InternalNodes. DecisionTreeusesEntropy,InformationGainandGiniIndexascriteriaforevaluating attributes [17]. It comes under Supervised Learning Algorithms. Fig 4 shows theDecisionTree generated on this dataset.

Random forest classifier

Random Forest classification comes under ensemble learning i.e it's an ensemble ofDecisionTrees.Itisabagging-basedalgorithm. ThefundamentalconceptusedbyRandom Forest is that a large number of uncorrelated Decision Trees operating as onegroupwill outperformeach ofthe individual constituenttree [18].

XGBoost classifier

XGBoost is Gradient-Boosting algorithm that makes use of Ensemble Learning and is tree- based. Each Decision Tree corrects the errors committed by its predecessor. This method is called Gradient Boosting. XGBoost makes use of Gradient Boosted Decision Trees.Each of these trees then ensemble to give a more accurate model. XGBoost uses Regularization to penalise complex trees and Cross validation to avoid overfitting of themodel. performs well because of its handling of data types, distributions and the variety of hyper parameters that can be tuned [19].

Fig4:DecisionTreeGenerated

Light BGM classifier

LightGBM is also a Gradient-Boosting algorithm that makes use of Ensemble Learning andistree-based.Thisalgorithmsharescommonfeaturessuchassparseoptimisation,parallel training, multiple loss functions and bagging with XGBoost. But, LightGBM growstrees leaf-wise instead of level-wise[20]. Out of the boosting four types of algorithmsavailable,thedefaultoptioni.eGBDT(gradientboostingdecisiontree) wasusedtoimplementthis model.

Flow chart of the proposed work is depicted in Fig 5

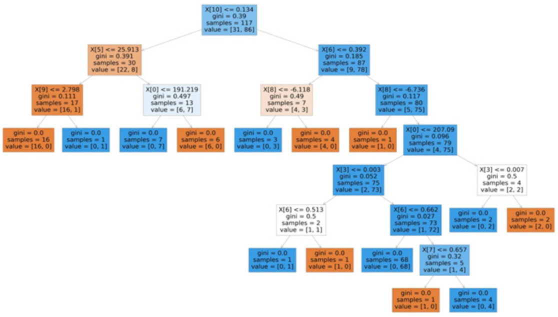


Fig 6: Scatter Plot

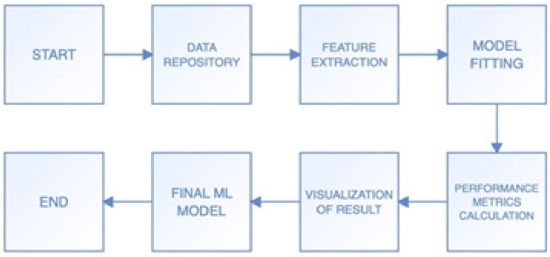
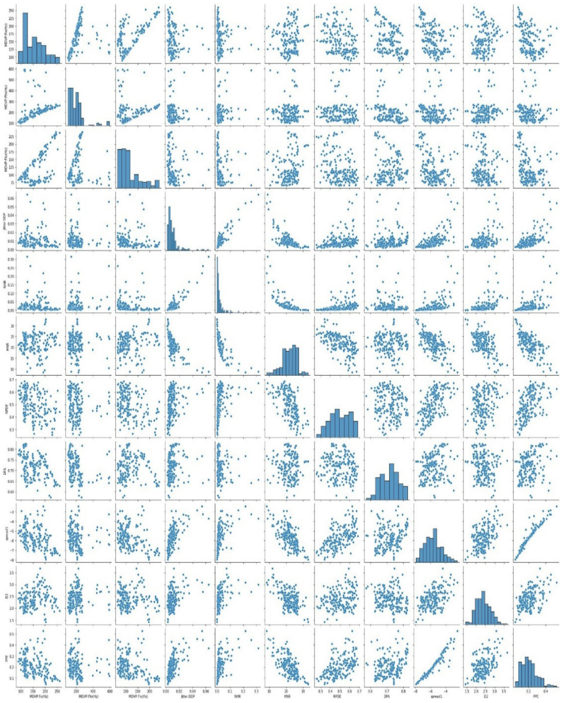


Fig6 shows the scatter plot generated on this dataset; itcontains11features.



Decision Tree Classification is the least performer of all, scoring 88.46%, while Random Forest Classification and XGBoost scored 93.58% and 96.15% respectively. Light GBM achieved highest accuracy of 97.43%. Fig 7-10 shows the confusion matrices of all the algorithms used. Confusion Matrix shows the number of True positive (TP), Truenegative (TN), False positive (FP) and False negative (FN) instances. Decision Tree Classifier has shown 57 True positive, 12 true negative, 5 False positive and 4 False negative instances. Hence a total of 69 instance have been correct out of 78. Therefore, the accuracy is 88.46%.

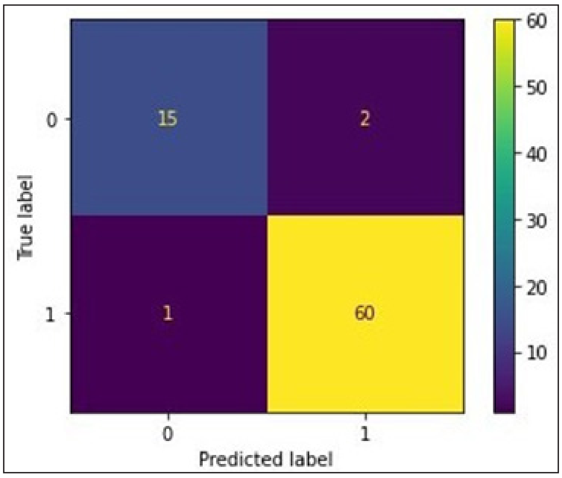


Fig 6DecisionTree

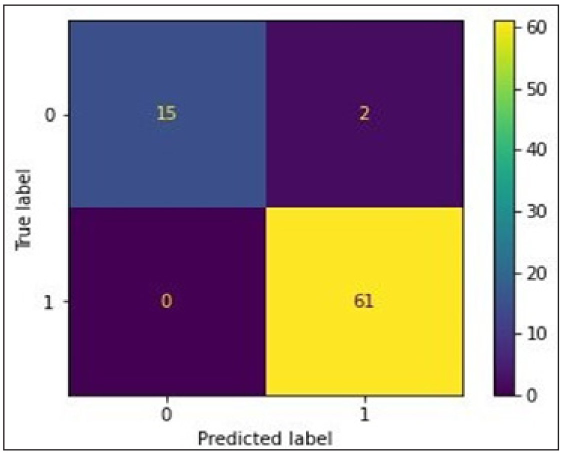


Fig 7RandomForest

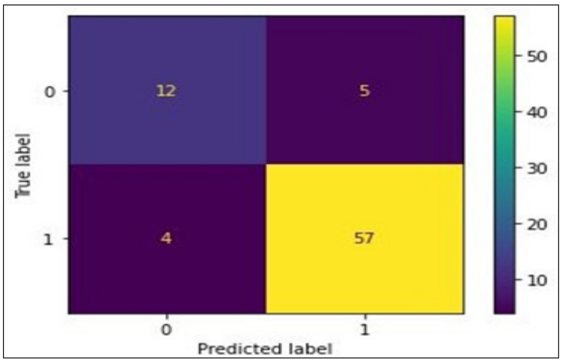


Fig 8 XGBoost

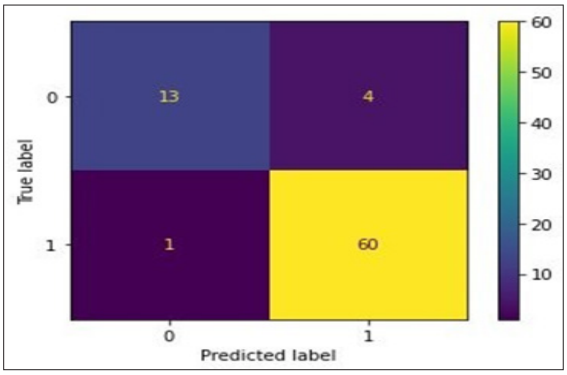


Fig 9 LightGBM

Variousotherparametersareobtainedaswellinordertocomparetheperformances.Theirformulas are listed below.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 - Score = \frac{2 \times Precision \times recall}{Precision + recall}$$

$$YI = recall + specificity - 1$$

$$Specificity = \frac{TN}{TN + FP}$$

$$error\_rate = \frac{FP + FN}{TP + TN + FN + FP}$$

The cemetrics were evaluated and listed in the table below.

Algorithm	Accuracy	Precision	Recall	F1_score	YoudenIndex	Specificity	Error-Rate
Decision Tree	0.8846	0.9193	0.9344	0.9267	0.6402	0.7058	0.1153
Random Forest	0.9358	0.9375	0.9836	0.9599	0.7483	0.7647	0.0641
XGBoost	0.9615	0.9677	0.9836	0.9755	0.8659	0.8823	0.0384
LightGBM	0.9743	0.9682	1.0000	0.9838	0.8823	0.8823	0.0256

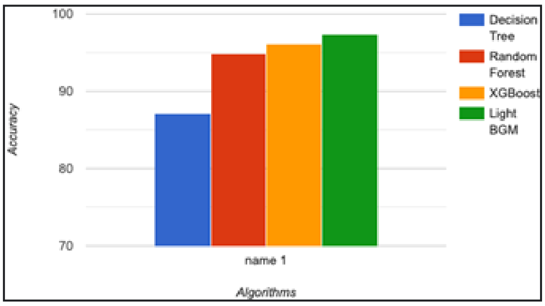


Fig 10: Accuracy

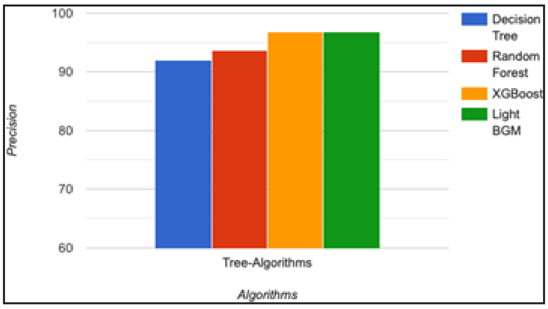


Fig 11: Precision

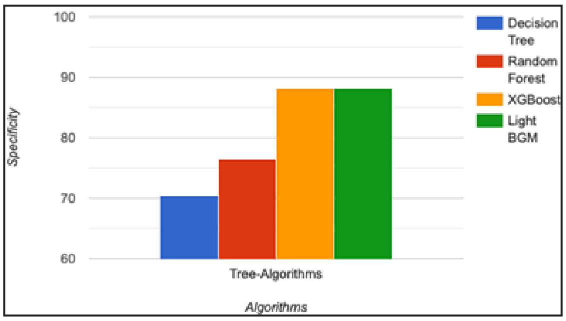


Fig 12: Specificity

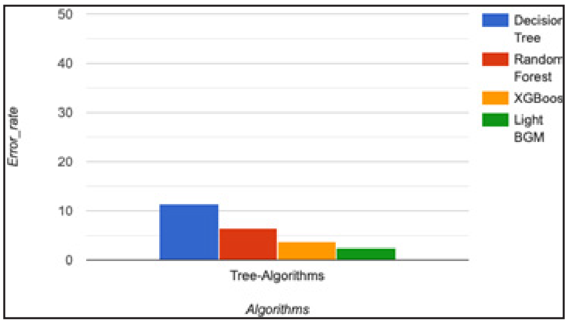


Fig 13: Error rate

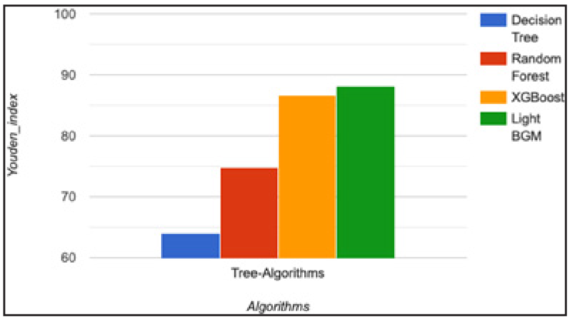


Fig 14: Youden-Index

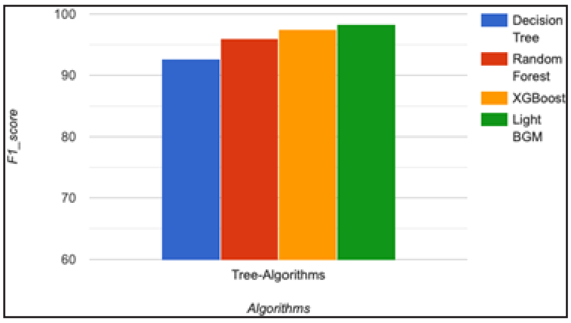


Fig 15: F1_score

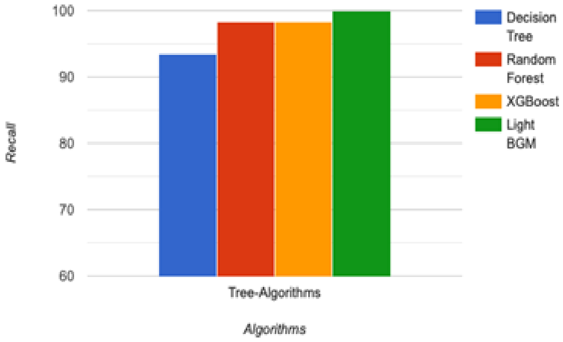


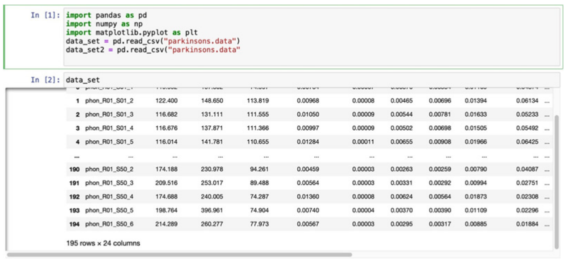
Fig 16: Recall

Fig 17-19 shows the code snippets of this work. They are screenshots of python notebook (.ipynb) file in jupyter notebook workspace.

Comparing with the work done by other researchers, this method achieved highest accuracy of 97.43% using LightGBM. Kuresanet.al [21] got 95.16% percent accuracy using HMM, while the work of other researchers is portrayed in the table below.

The workflow ofthe proposed model is summarised below:

Author	AlgorithmUsed	Accuracy
Kuresanetal, 2019[21]	HMM	95.16%
Hakan Gunduz[22]	CNN SVM	83.3% 86.9%
Marar et al, 2018[23]	ANN	94.87%
Goyalet.al[24]	XGBoost	91.40%
Mathuretal[25]	KNN+Adaboost KNN+MLP	91.28% 91.28%
KarapinarSen-turk,2020[26]	SVM	93.84%
ProposedWork	LightGBM	97.43%



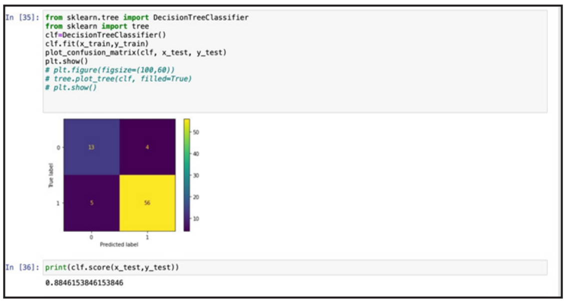
```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        data_set = pd.read_csv("parkinsons.data")
        data_set2 = pd.read_csv("parkinsons.data"

In [2]: data_set
```

```
  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 ...
...             ...      ...      ...      ...      ...      ...      ...      ...      ...      ... ...
190  phon_R01_S50_2  174.188  230.978   94.261  0.00459  0.00003  0.00263  0.00259  0.00790  0.04087 ...
191  phon_R01_S50_3  209.516  253.017   89.488  0.00564  0.00003  0.00331  0.00292  0.00994  0.02751 ...
192  phon_R01_S50_4  174.688  240.005   74.287  0.01360  0.00008  0.00624  0.00564  0.01873  0.02308 ...
193  phon_R01_S50_5  198.764  396.961   74.904  0.00740  0.00004  0.00370  0.00390  0.01109  0.02296 ...
194  phon_R01_S50_6  214.289  260.277   77.973  0.00567  0.00003  0.00295  0.00317  0.00885  0.01884 ...

195 rows × 24 columns
```

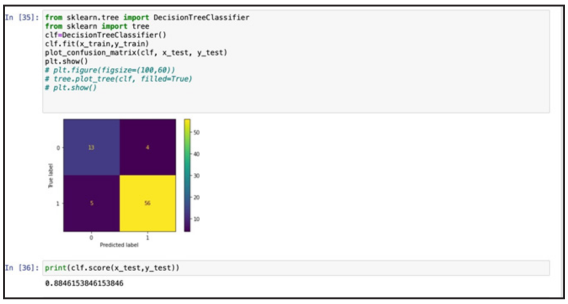
Fig17: Importing libraries and loading the dataset



```
In [35]: from sklearn.tree import DecisionTreeClassifier
         from sklearn import tree
         clf=DecisionTreeClassifier()
         clf.fit(x_train,y_train)
         plot_confusion_matrix(clf, x_test, y_test)
         plt.show()
         # plt.figure(figsize=(100,60))
         # tree.plot_tree(clf, filled=True)
         # plt.show()
```

```
In [36]: print(clf.score(x_test,y_test))
         0.8846153846153846
```

Fig18: Fitting the model with training data



```
In [35]: from sklearn.tree import DecisionTreeClassifier
         from sklearn import tree
         clf=DecisionTreeClassifier()
         clf.fit(x_train,y_train)
         plot_confusion_matrix(clf, x_test, y_test)
         plt.show()
         # plt.figure(figsize=(100,60))
         # tree.plot_tree(clf, filled=True)
         # plt.show()
```

```
In [36]: print(clf.score(x_test,y_test))
         0.8846153846153846
```

Fig19: Generating the correlation matrix and printing the accuracy

(1)Retrieve the dataset. (2)Calculate Pearson correlation coefficient tand generate cool warm colour graded matrix. (3)Remove one of the highly correlated features(coefficient>0.9). (4)Split the dataset into training and testing data. (5)BuildML model using DecisionTree, Random Forest, XGBoost, LightGBM algorithms. (6)Fit the models with training data. (7)Obtain the values of Accuracy, Precision, Recall, F1_score, Specificity and Error Rate. (8) Generate confusion matrices of the four models. (9)Compare the models using the obtained parameters.

Conclusion

Early detection of PD is essential to initiate appropriate treatment and to better understand the disease. Voice data is extremely important for this study. Machine Learning algorithms continue to prove useful in the area of medical diagnosis. The present method performs diagnosis of PD by making use of tree-based machine learning algorithms. LightGBM achieved the highest accuracy of 97.43%. The results show that boosting tree algorithms achieved better accuracy than regular tree-based algorithms as XGBoost and LightGBM performed superiorly. This method provides an automated diagnosis of PD and achieves clinical level accuracy. Application of this work will have great impact on health care system by improving the diagnosis of PD and thereby reduce its severity.

Future Scope

XGBoost and LightGBM can further be hyper parameter tuned in order to produce desired results. Performance of the ML model can further be improved by tuning. This however can vary depending on the dataset taken and the features selected. This Model can further be evaluated on larger datasets and accuracy can be tested.

References

1. Sherer TB, S Chowdhury, K Peabody, D Brooks: Overcoming obstacles in Parkinson’s Disease. Movement Disorders 27(13), 1606-1611 (2012) .

2. National Institute for Health and Clinical Excellence (2006) Parkinson’s Disease: Diagnosis and Management in Primary and Secondary Care. London: NICE (http:// guidance.nice.org. uk/CG35).

3. Progression of Voice and Speech Impairment in the Course of Parkinson’s Disease: A Longitudinal Study S. Skodda, W. Grönheit, N. Mancinelli, and U. Schlegel

4. Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection’, Little MA, McSharry PE, Roberts SJ, Costello DAE, Moroz IM. BioMedical Engineering OnLine 2007, 6:23 (26 June 2007)

5. Anyanwu MN, Shiva SG. Comparative analysis of serial decision tree classification algorithms. International Journal

of Computer Science and Security. 2009 Jun;3(3):230-40

6. “Parkinson Disease Detection Using Deep Neural Networks”, Shivangi, Anubhav Johri and Ashish Tripathi, Department of Computer Science Jaypee Institute of Information Technology Noida, India.

7. Agarwal, A., Chandrayan, S., and Sahu, S. S. (2016). “Prediction of Parkinson’s disease using speech signal with Extreme Learning Machine,” in 2016 International Conference on Electrical, Electronics, and Optimisation Techniques (ICEEOT) (Chennai), 3776–3779. doi: 10.1109/ ICEEOT.2016.77 55419

8. H.Gunduz,’’Deep learning-based Parkinson’s disease classification using vocal feature sets,’’ IEEE Access, vol. 7, pp. 115540–115551, 2019.

9. A. Zhao, L. Qi, J. Li, J. Dong, and H. Yu, ‘‘A hybrid spatio-temporal model for detection and severity rating of Parkinson’s disease from gait data,’’ Neurocomputing, vol. 315, pp. 1–8, Nov. 2018.

10. G. Valenza, S. Orsolini, S. Diciotti, L. Citi, E. P. Scilingo, M. Guerrisi, S. Danti, C. Lucetti, C. Tessa, R. Barbieri, and N. Toschi, ‘‘Assessment of spontaneous cardiovascular oscillations in Parkinson’s disease,’’ Biomed. Signal Process. Control, vol. 26, pp. 80–89, Apr. 2016.

11. L. Silveira-Moriyama, A. Petrie, D. R. Williams, A. Evans, R. Katzenschlager, E. R. Barbosa, and A. J. Lees, ‘‘The use of a colour coded probability scale to interpret smell tests in suspected Parkinsonism,’’ Movement Disorders, vol. 24, no. 8, pp. 1144–1153, Jun. 2009.

12. S. Bilgin, ‘‘The impact of feature extraction for the classification of amyotrophic lateral sclerosis among neurodegenerative diseases and healthy subjects,’’ Biomed. Signal Process. Control, vol. 31, pp. 288–294, Jan. 2017.

13. I. El Maachi, G.-A. Bilodeau, and W. Bouachir, ‘‘Deep 1D-convnet for accurate Parkinson disease detection and severity prediction from gait,’’ Expert Syst. Appl., vol. 143, Apr. 2020, Art. no. 113075.

14. T. Arroyo-Gallego, M. J. Ledesma-Carbayo, A. Sanchez-Ferro, I.Butterworth, C. S. Mendoza, M. Matarazzo, P. Montero, R. Lopez-Blanco, V. Puertas-Martin, R. Trincado, andL. Giancardo, ‘‘Detection of motor impairment in Parkinson’s disease via mobile touchscreen typing,’’ IEEE Trans. Biomed. Eng., vol. 64, no. 9, pp. 1994–2002, Sep. 2017.

15. R. Das, ‘‘A comparison of multiple classification methods for diagnosis of Parkinson disease,’’ Expert Syst. Appl., vol. 37, no. 2, pp. 1568–1572, Mar. 2010.

16. Cudeck, R. (1989). Analysis of correlation matrices using covariance structure models. Psychological Bulletin, 105(2), 317–327.

17. Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modelling. Journal of Chemometrics, 18 (6), 275–285.

18. Janitza, S., Kruppa, J., König, I.R., &Boulesteix, A.-L. (2012). Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. WIREs Data Mining and Knowledge Discovery, 2 (6), 493–507.

19. Chen T and Guestrin C 2016 XGBoost: A Scalable Tree Boosting System[J].

20. K. Guolin, M. Qi, F. Thomas, W. Taifeng,

C. Wei, M. Weidong, Y. Qiwei, L. Tie-Yan, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," Advances in Neural Information Processing Systems vol. 30, pp. 3149-3157, 2017.

21. Kuresan, H., Samiappan, D., and Masunda, S. (2019). Fusion of WPT and MFCC feature extraction in Parkinson's disease diagnosis. Technol. Health Care 27, 363–372. doi: 10.3233/THC-181306

22. Gunduz, Hakan. "Deep learning-based Parkinson's disease classification using vocal feature sets." IEEE Access 7 (2019): 115540-115551.

23. Marar, S., Swain, D., Hiwarkar, V., Motwani, N., and Awari, A. (2018). "Predicting the occurrence of Parkinson's disease using various classification models," in 2018 International Conference on Advanced Computation and Telecommunication (ICACAT) (Bhopal), 1–5. doi: 10.1109/ICACAT.2018.8933579

24. Goyal, Jinee, Padmavati Khandnor, and Trilok Chand Aseri. "A Comparative Analysis of Machine Learning classifiers for Dysphonia-based classification of Parkinson's Disease." International Journal of Data Science and Analytics 11, no. 1 (2021): 69-83.

25. Mathur, Richa, Vibhakar Pathak, and Devesh Bandil. "Parkinson disease prediction using machine learning algorithm." In Emerging Trends in Expert Applications and Security, pp. 357-363. Springer, Singapore, 2019.

26. KarapinarSenturk, Z. (2020). Early diagnosis of Parkinson's disease using machine learning algorithms. Med. Hypoth. 138:109603. doi: 10.1016/j.mehy.2020.109603