PIE INFOCOMM PRIVATE LIMITED

PTE INFOCOMM

RESEARCH, TRAINING AND DEVELOPMENT

A Report on

PARKINSON DISEASE PREDICTION

Under the Guidance(Mentor) of

Aishwarya Saxena

By – Yash Gaikwad

Machine Learning - Python

College - Pad. Dr DY Patil Institute of Engineering, Management & Research

Summer Internship - September 2020

INDEX

SR NO.	TOPIC	PAGE NO.
1	INTRODUCTION	3
2	OBJECTIVE	3
3	BACKGROUND	4
4	HARDWARE AND SOFTWARE REQUIREMENTS	7
5	CODING	8
6	OUTPUTS	10
7	FUTURE SCOPE	13
8	CONCLUSION	13
9	BIBLIOGRAPHY AND REFERENCE	13

PARKINSON DISEASE PREDICTION

INTRODUCTION :

Parkinson's disease is a progressive disorder of the central nervous system affecting movement and inducing tremors and stiffness. It has 5 stages to it and affects more than 1 million individuals every year in India. This is chronic and has no cure yet. It is a neurodegenerative disorder affecting dopamine-producing neurons in the brain.

In a data warehouse, the user usually has a view of multiple data-sets collected from different data sources and understanding their relationships is an extremely important part of the KDD process. Data mining is a rapidly evolving advanced technology that used in bio-medical sciences and research in order to predict and analyze large volumes of medical data such as Parkinson's disease. Data mining in neuro- degenerative diseases like Parkinson disease is an emerging field with enormous importance for deeper understanding of mechanisms relevant to disease, providing prognosis and complete treatment.
A relatively high prediction performance has been achieved with classification accuracy. Classification algorithms predict accuracy for discrete variables that are relevant on the other attributes present in the data-set. The continual increase of the number of features can significantly contribute with clustering-based feature.

OBJECTIVE :

In this Python machine learning project, using the ***Python libraries*** scikit-learn, numpy, pandas, seaborn, Matplotlib, math & time we are predicting Parkinson diseased patient.
Decision support tools are gaining significant research interest due to their potential to improve health-care provision. Among many possible approaches, those that provide noninvasive monitoring and diagnosis of diseases are of increased interest to clinicians and biomedical engineers.
We aim to provide this diagnosis to people in remote areas where healthcare is not just lacking but extremely inadequate.

BACKGROUND :

Libraries used :
1. Numpy
2. Pandas

Typically, the symptoms of PD are attenuated by the use of dopaminergic medications such as levodopa. During data collection, patients were asked to give information regard- ing when, relative to taking medication, they provided their data. The options included: *Just after Parkinson medication (at your best)*, *Another time*, *Immediately before Parkinson medication*, *I don't take Parkinson medications*, and *no value*. These medication time points were interpreted to mean: time of best symptom control, on medication but not immediately before or after, time of worst symptoms, not on medications, and not applicable, respectively. This information, crossed with the clinical diagnosis responses from the demographics survey led to three groups of patients and data, as shown in Figure 1. Patients that had medication prior to the voice test were not used as participants in the analysis. The rationale for this parameter selection is that the voice of the patient will depict the most extreme effects of the PD without the effect of any medication. The assumption is that the voice features will be noticeably different from those of the controls. The control in this experiment is a participant who has not been professionally diagnosed with PD.

Each patient could contribute to multiple voice submissions, so the number of unique audio files exceeds the total number of patients surveyed. Based on the data extracted from these studies, a csv PyAudioAnalysis library in Python. was contained the data linked with the health unique to each. The voice was also pre-processed.

Figure 1

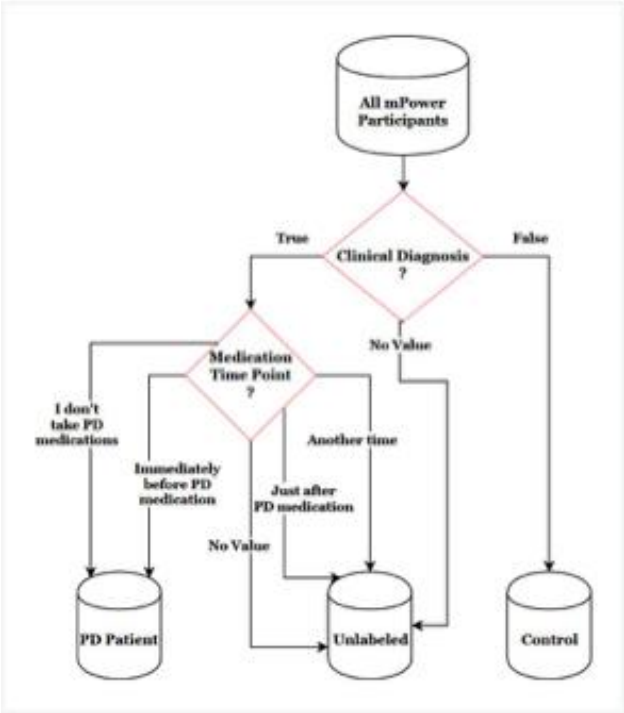
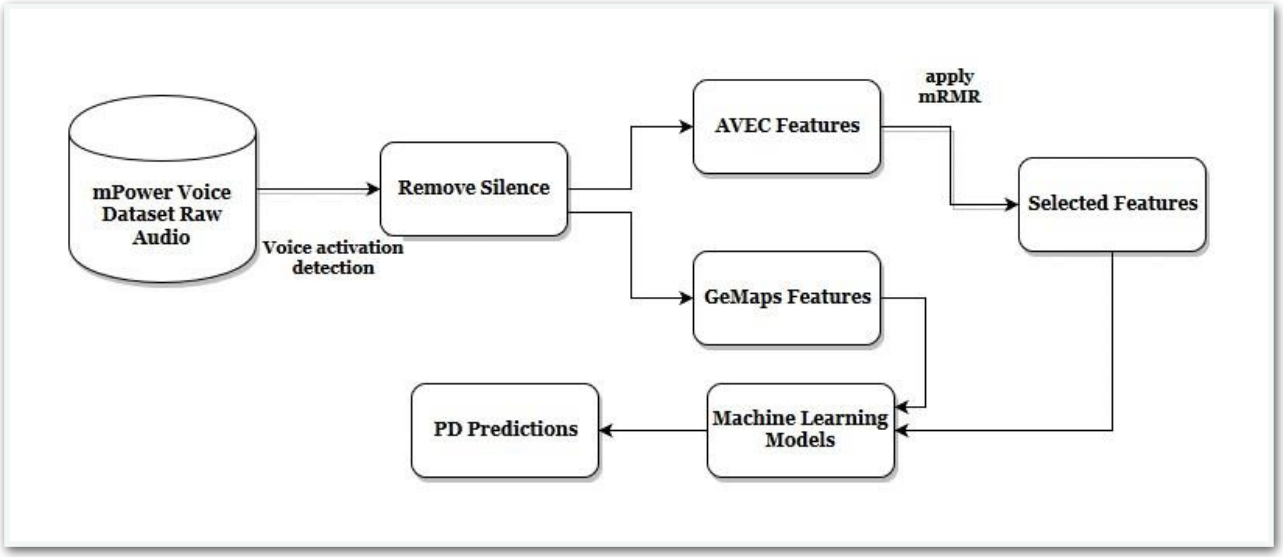


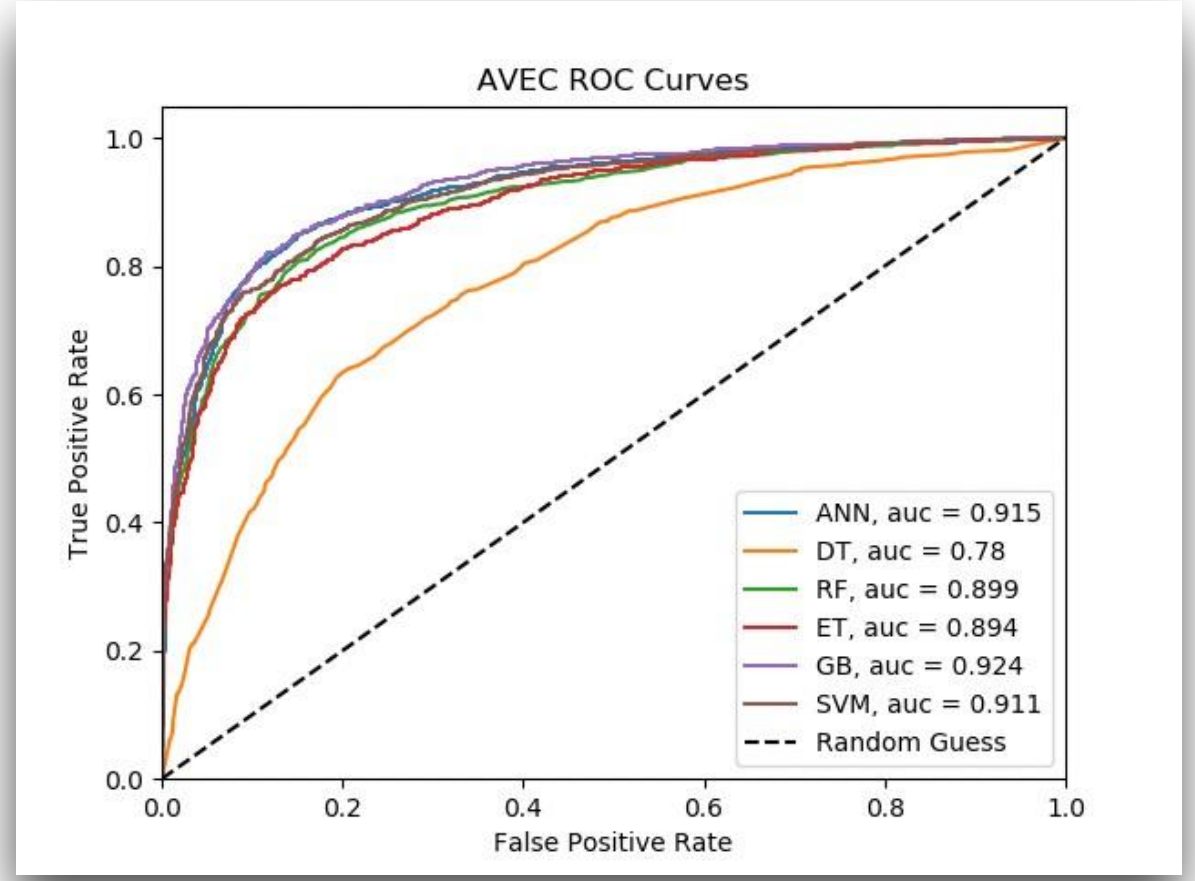
Figure 2

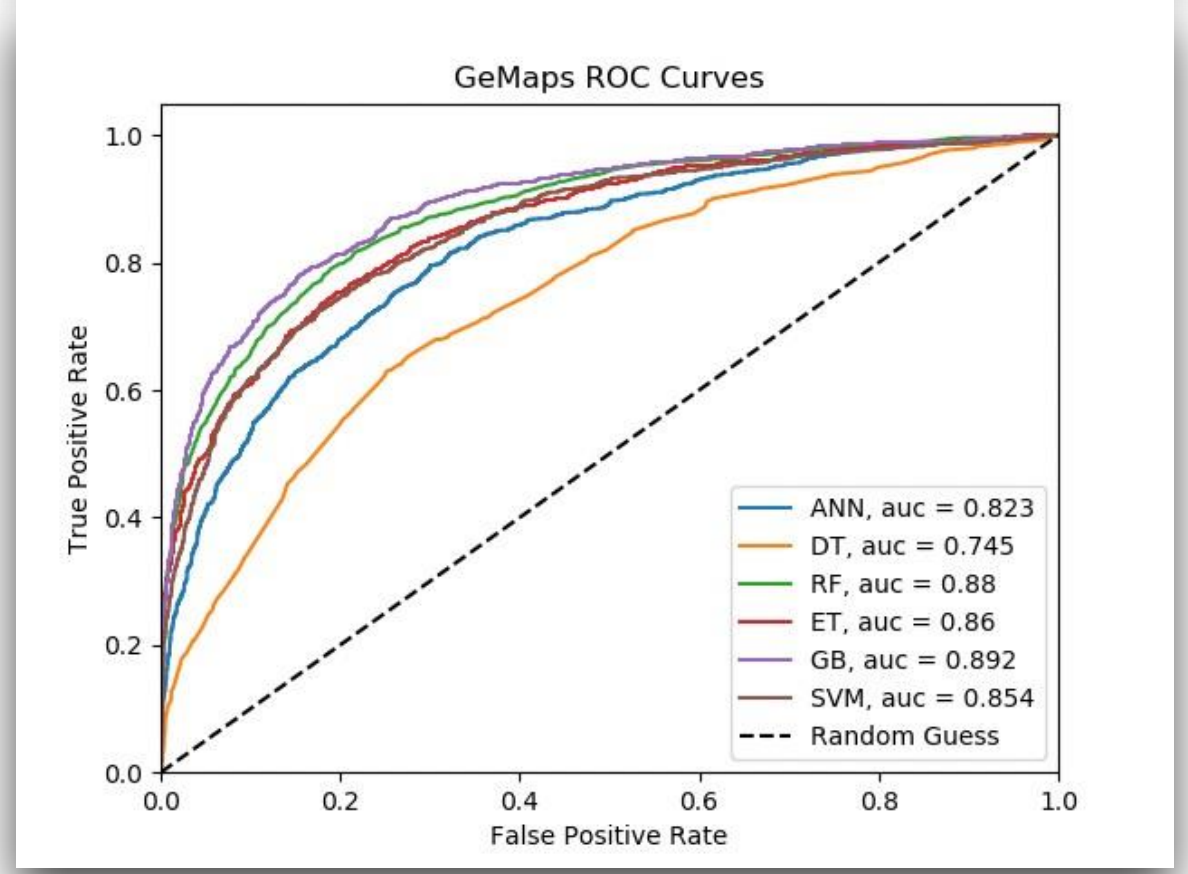


A series of decision tree classifiers were used to classify the dataset including standard decision trees, random forest, gradient boosted decision trees and extra tree classifiers. A decision tree operates by creating binary decision boundaries about features to separate the data homogeneously between the two classes by using a metric that minimizes information entropy. In aggregate, these separations create a classification accuracy over the training set that is the applied to the testing set to assess generalization. Random forests are an extension of decision trees that use arbitrary mixing of the data to create different subsets of the training data which are then run through decision tree models. These models are then tested for accuracy for samples not used in the sub trees and parameters are tuned to maximize the expected accuracy of the model over the training set.

$\gamma = \frac{1}{N_{features}}$ (1)

Figure 3


AVEC ROC Curves
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ANN, auc = 0.915
DT, auc = 0.78
RF, auc = 0.899
ET, auc = 0.894
GB, auc = 0.924
SVM, auc = 0.911
Random Guess


GeMaps ROC Curves
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ANN, auc = 0.823
DT, auc = 0.745
RF, auc = 0.88
ET, auc = 0.86
GB, auc = 0.892
SVM, auc = 0.854
Random Guess

The algorithm's performance here is limited to that of the clinician. We are confident that with more patient driven data, the accuracy of these models using speech as a biomarker for disease can be improved, especially when validated against currently available biomarkers such as the DaT scan. Ultimately, PD diagnosis primarily relies on clinically observed ratings and biomarker confirmation and is not sought in the majority of cases because of the clear response most patients show to treatment. The goal of a digital biomarker then shifts more toward not only accurately capturing the state of PD in a patient, but also learning the individual patient's symptoms and providing enhanced care by assisting with treatment management and assessing severity progression.

The models trained on the AVEC features often outperformed the models trained on GeMaps features based on metrics of accuracy, precision, recall and F-1. A possible reason for this trend is that there is more information encoded within the feature vectors for AVEC that can correlate to PD diagnosis. The AVEC features contain 1200 unique dimensions of information drawn from the audio recording while the GeMaps features only contain 62 dimensions. This validates the concept that as more information can be drawn from the patient regarding their health, better diagnostic accuracy can be acquired using automated machine learning models.

HARDWARE AND SOFTWARE REQUIREMENTS :

HARDWARE TOOLS	MINIMUM REQUIREMENTS
Processor	i5 or above
Hard Disk	10GB
RAM	8GB
Monitor	17" Coloured
Mouse	Optical
Keyboard	122 Keys

SOFTWARE TOOLS	MINIMUM REQUIREMENTS
Platform	Windows, Linux or MacOS
Operating System	Windows, Linux or MacOS
Technology	Machine Learning-Python
Scripting Language	Python
IDE	Spyder

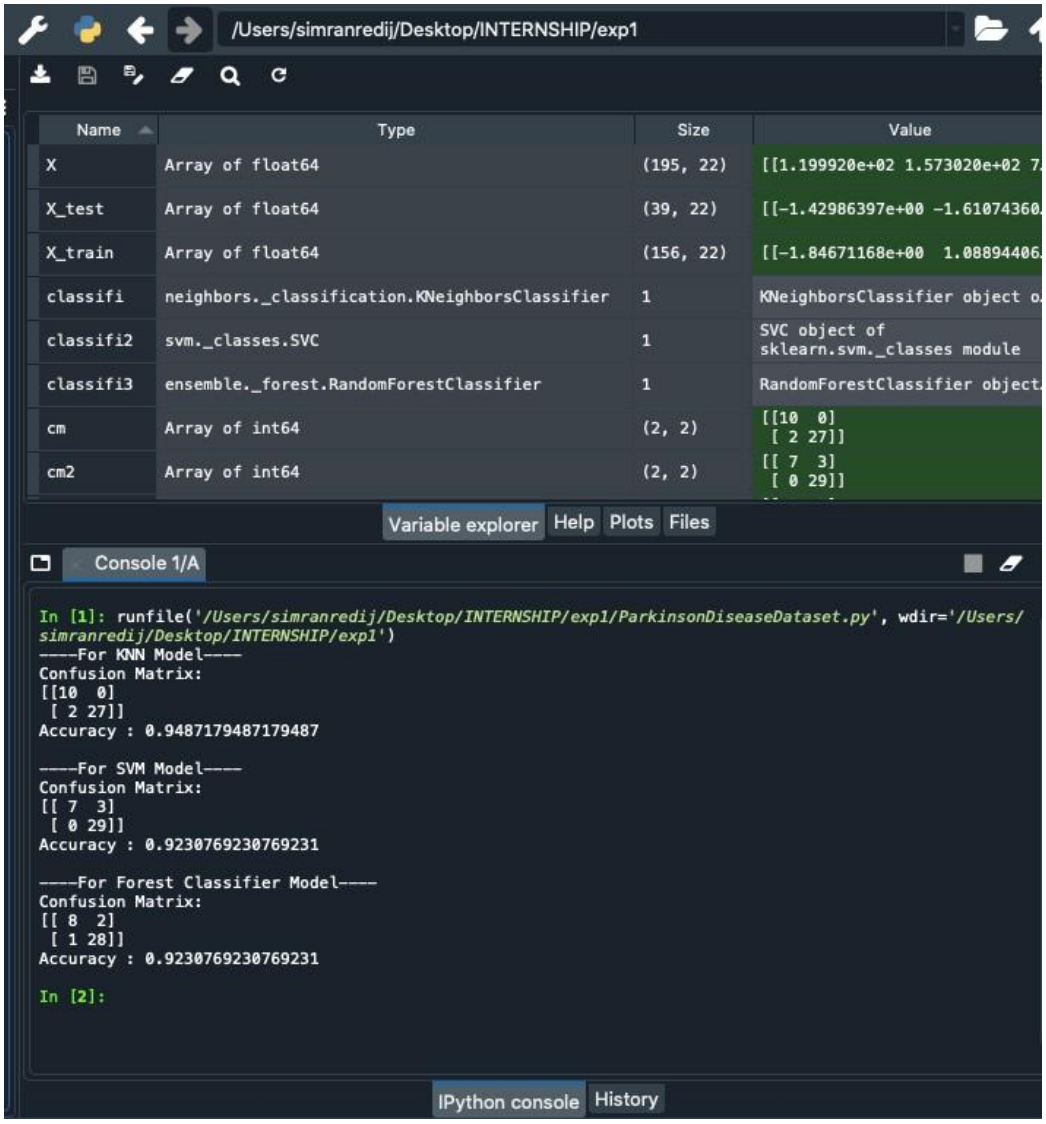
CODING :

```
#importing libraries
import numpy as np
import pandas as pd

#importing dataset
dataset = pd.read_csv("parkinsons.csv")
X = dataset.iloc[:,[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,18,19,20,21,22,23]].values
y = dataset.iloc[:,17].values

#splitting the dataset into train and test dataset
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y, test_size = 0.2, random_state =0)

# feature scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)

#applying PCA

from sklearn.decomposition import PCA
pca = PCA(n_components = None)
X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)
variance = pca.explained_variance_ratio_

#fitting into knn model

from sklearn.neighbors import KNeighborsClassifier
classifi = KNeighborsClassifier(n_neighbors = 8,p=2,metric ='minkowski')
classifi.fit(X_train,y_train)

#predicting reults
y1_pred = classifi.predict(X_test)

#fitting the model in SVM
from sklearn.svm import SVC
classifi2 = SVC()
classifi2.fit(X_train,y_train)

#predicting reults
y2_pred = classifi2.predict(X_test)
```

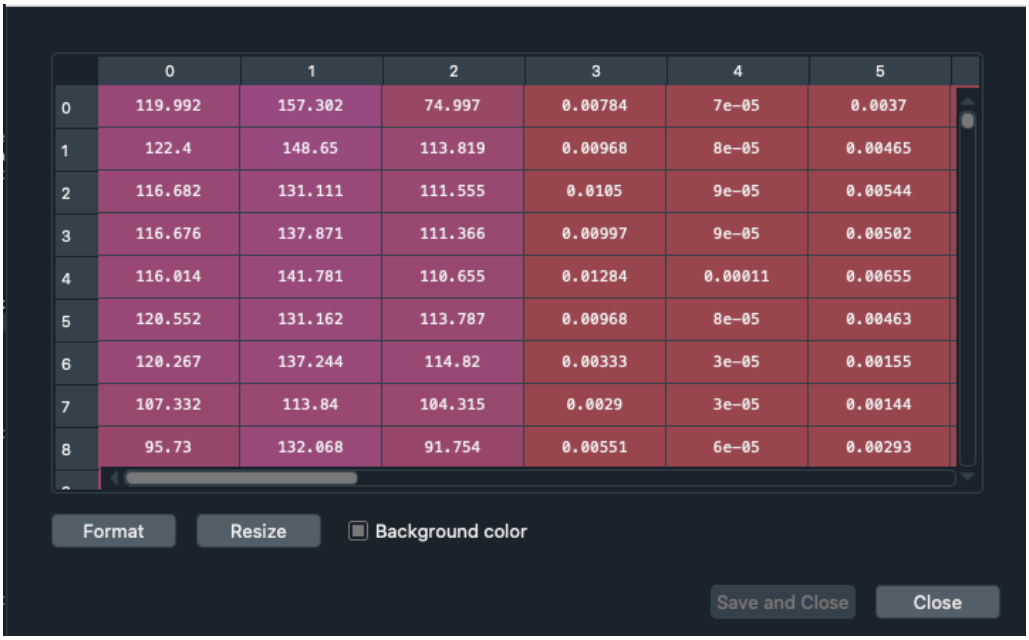
```
#fitting the data in random forest classifier
from sklearn.ensemble import RandomForestClassifier
classifi3 = RandomForestClassifier(n_estimators=16,criterion = "entropy",random_state=0)
classifi3.fit(X_train,y_train)

#predicting reults
y3_pred = classifi3.predict(X_test)

#Analyzing
from sklearn.metrics import confusion_matrix,accuracy_score

#KNN model
print("----For KNN Model --- ")
cm=confusion_matrix(y_test,y1_pred)
print("Confusion Matrix: ")
print(cm)
print("Accuracy : " + str(accuracy_score(y_test,y1_pred)))

print()

#SVM model
print("----For SVM Model --- ")
cm2=confusion_matrix(y_test,y2_pred)
print("Confusion Matrix: ")
print(cm2)
print("Accuracy : " + str(accuracy_score(y_test,y2_pred)))

print()

#Random Forest Classifier Model
print("----For Forest Classifier Model--- ")
cm3=confusion_matrix(y_test,y3_pred)
print("Confusion Matrix: ")
print(cm3)
print("Accuracy : " + str(accuracy_score(y_test,y3_pred)))
```

OUTPUTS :

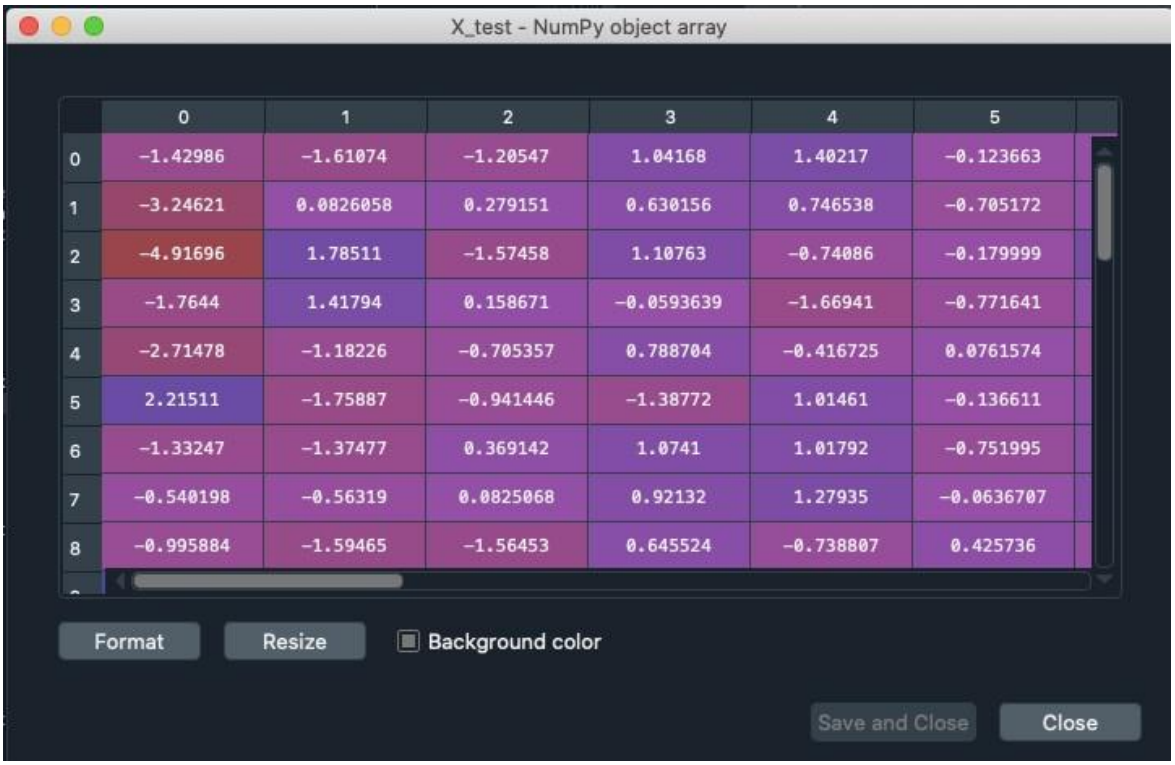
1)


/Users/simranredij/Desktop/INTERNSHIP/exp1
Name
Type
Size
Value
X
Array of float64
(195, 22)
X_test
Array of float64
(39, 22)
X_train
Array of float64
(156, 22)
classifi
neighbors._classification.KNeighborsClassifier
1
classifi2
svm._classes.SVC
1
SVC object of sklearn.svm._classes module
classifi3
ensemble._forest.RandomForestClassifier
1
cm
Array of int64
(2, 2)
[[10 0]
[2 27]]
cm2
Array of int64
(2, 2)
[[7 3]
[0 29]]
Variable explorer
Help
Plots
Files
Console 1/A
In [1]: runfile('/Users/simranredij/Desktop/INTERNSHIP/exp1/ParkinsonDiseaseDataset.py', wdir='/Users/simranredij/Desktop/INTERNSHIP/exp1')
----For KNN Model----
Confusion Matrix:
[[10 0]
[2 27]]
Accuracy : 0.9487179487179487
----For SVM Model----
Confusion Matrix:
[[7 3]
[0 29]]
Accuracy : 0.9230769230769231
----For Forest Classifier Model----
Confusion Matrix:
[[8 2]
[1 28]]
Accuracy : 0.9230769230769231
In [2]:
IPython console
History

2)

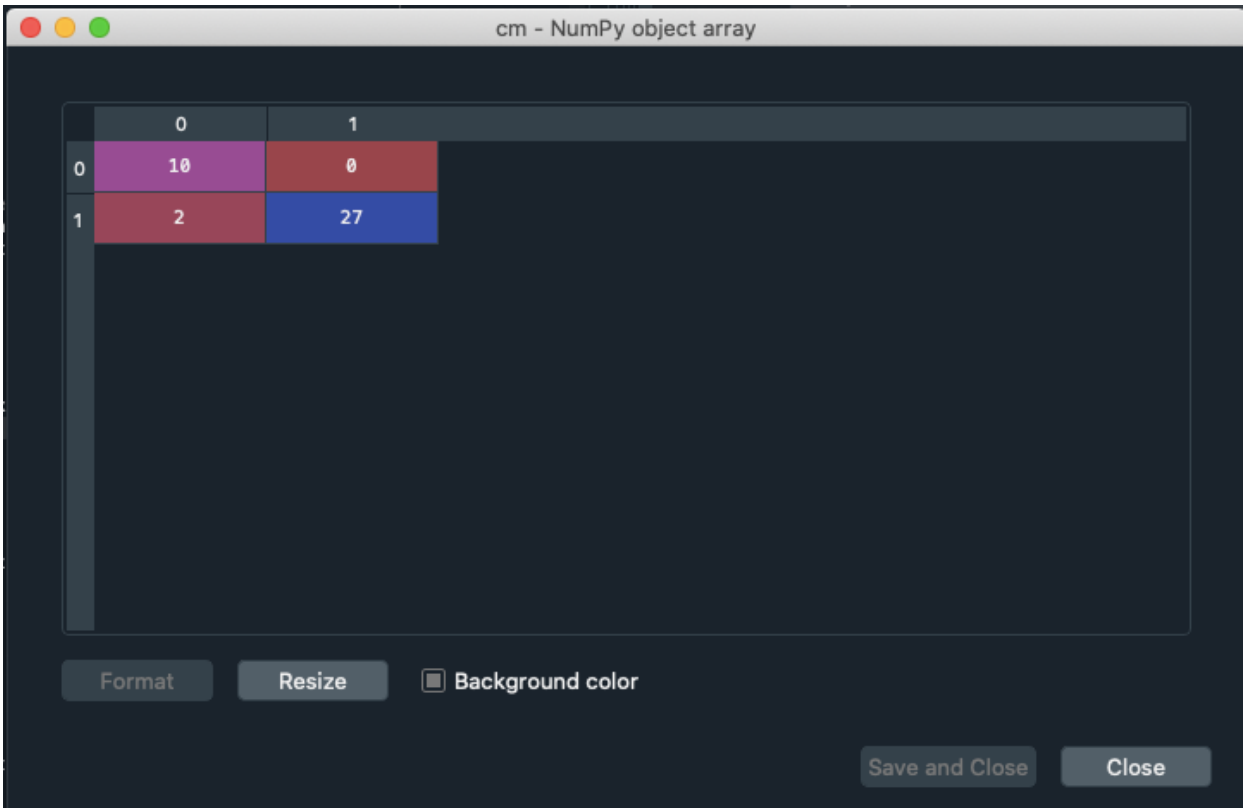

0 1 2 3 4 5
0 119.992 157.302 74.997 0.00784 7e-05 0.0037
1 122.4 148.65 113.819 0.00968 8e-05 0.00465
2 116.682 131.111 111.555 0.0105 9e-05 0.00544
3 116.676 137.871 111.366 0.00997 9e-05 0.00502
4 116.014 141.781 110.655 0.01284 0.00011 0.00655
5 120.552 131.162 113.787 0.00968 8e-05 0.00463
6 120.267 137.244 114.82 0.00333 3e-05 0.00155
7 107.332 113.84 104.315 0.0029 3e-05 0.00144
8 95.73 132.068 91.754 0.00551 6e-05 0.00293
Format
Resize
Background color
Save and Close
Close

3)



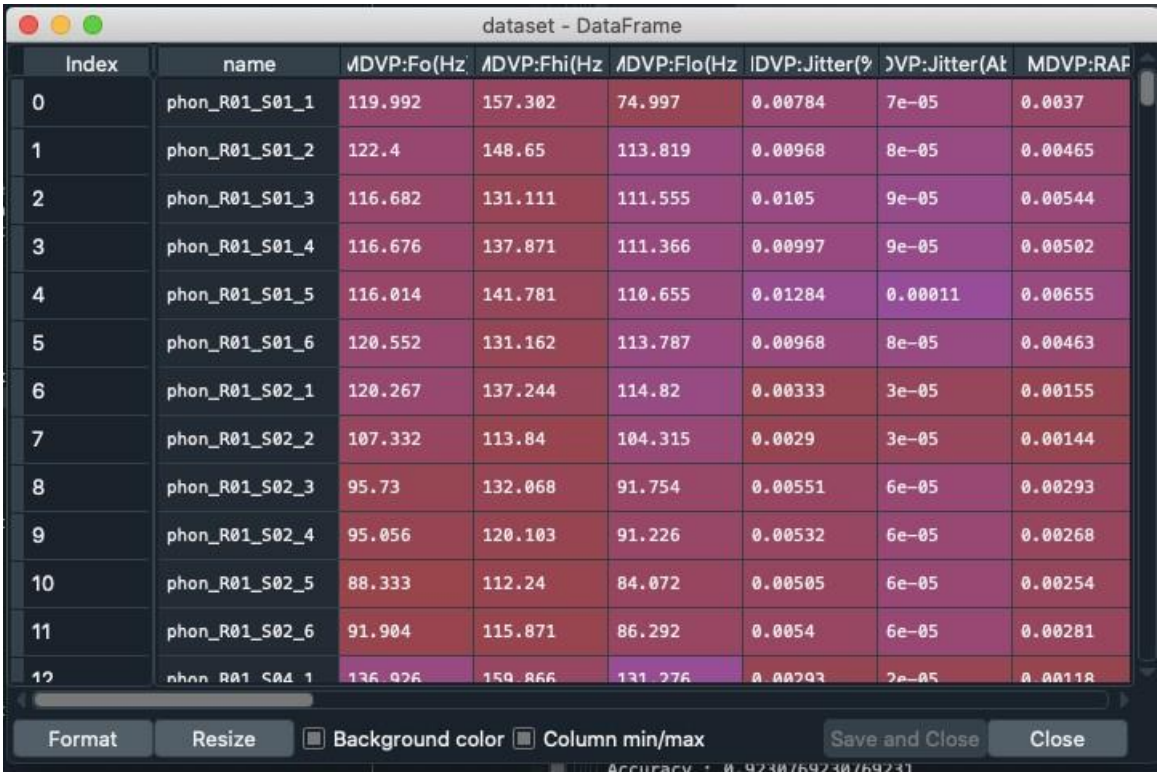
	0	1	2	3	4	5
0	-1.42986	-1.61074	-1.20547	1.04168	1.40217	-0.123663
1	-3.24621	0.0826058	0.279151	0.630156	0.746538	-0.705172
2	-4.91696	1.78511	-1.57458	1.10763	-0.74086	-0.179999
3	-1.7644	1.41794	0.158671	-0.0593639	-1.66941	-0.771641
4	-2.71478	-1.18226	-0.705357	0.788704	-0.416725	0.0761574
5	2.21511	-1.75887	-0.941446	-1.38772	1.01461	-0.136611
6	-1.33247	-1.37477	0.369142	1.0741	1.01792	-0.751995
7	-0.540198	-0.56319	0.0825068	0.92132	1.27935	-0.0636707
8	-0.995884	-1.59465	-1.56453	0.645524	-0.738807	0.425736

4)

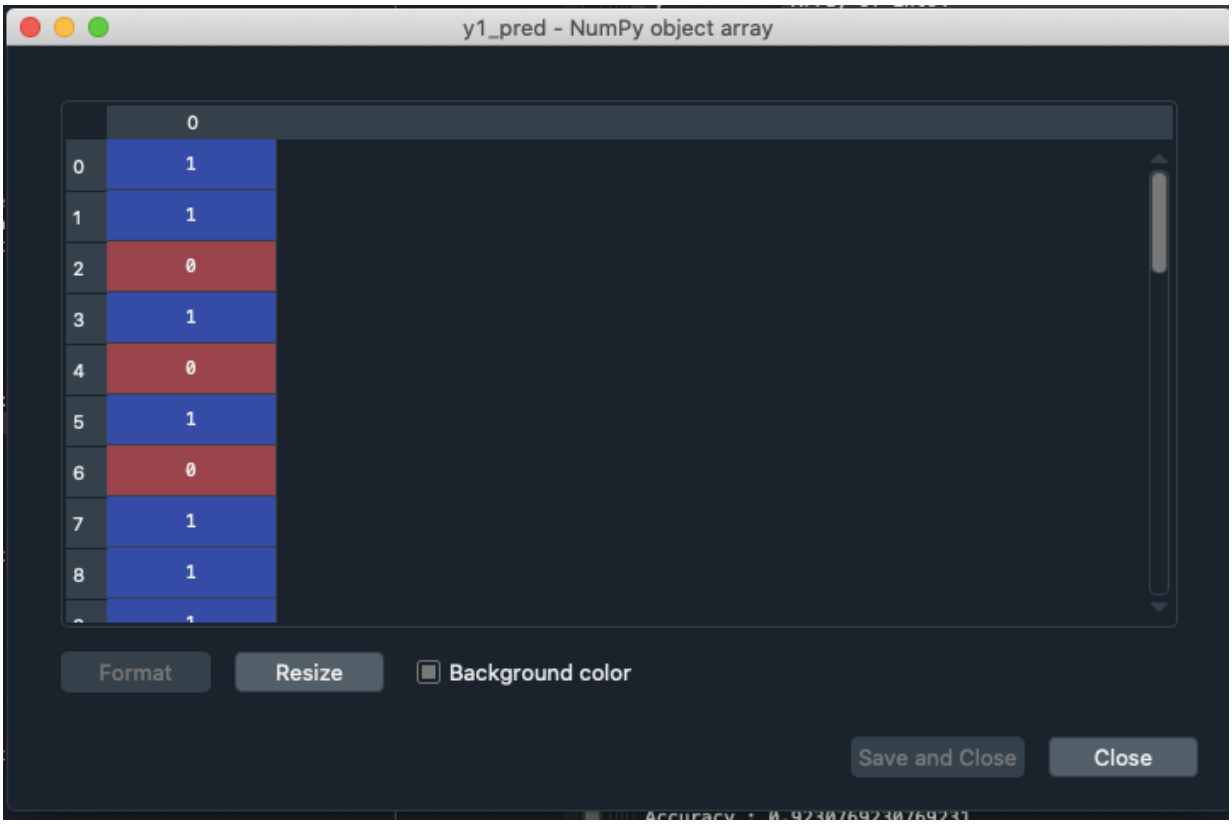


	0	1
0	10	0
1	2	27

5)


dataset - DataFrame
Index
name
MDVP:Fo(Hz
MDVP:Fhi(Hz
MDVP:Flo(Hz
MDVP:Jitter(%
MDVP:Jitter(Ab
MDVP:RAP
0 phon_R01_S01_1 119.992 157.302 74.997 0.00784 7e-05 0.0037
1 phon_R01_S01_2 122.4 148.65 113.819 0.00968 8e-05 0.00465
2 phon_R01_S01_3 116.682 131.111 111.555 0.0105 9e-05 0.00544
3 phon_R01_S01_4 116.676 137.871 111.366 0.00997 9e-05 0.00502
4 phon_R01_S01_5 116.014 141.781 110.655 0.01284 0.00011 0.00655
5 phon_R01_S01_6 120.552 131.162 113.787 0.00968 8e-05 0.00463
6 phon_R01_S02_1 120.267 137.244 114.82 0.00333 3e-05 0.00155
7 phon_R01_S02_2 107.332 113.84 104.315 0.0029 3e-05 0.00144
8 phon_R01_S02_3 95.73 132.068 91.754 0.00551 6e-05 0.00293
9 phon_R01_S02_4 95.056 120.103 91.226 0.00532 6e-05 0.00268
10 phon_R01_S02_5 88.333 112.24 84.072 0.00505 6e-05 0.00254
11 phon_R01_S02_6 91.904 115.871 86.292 0.0054 6e-05 0.00281
Format
Resize
Background color
Column min/max
Save and Close
Close

6)


y1_pred - NumPy object array
0
0 1
1 1
2 0
3 1
4 0
5 1
6 0
7 1
8 1
Format
Resize
Background color
Save and Close
Close

FUTURE SCOPE :

- FOR DOCTORS:

(1) Improving disease accuracy
(2) Anticipation of their patients' disease evolution
(3) Support to decision for choosing treatments for patients

- FOR PD PATIENTS:

(1) For people with risk of getting PD: early detection and monitoring
(2) For early-stage/late-stage PD people: monitoring better their disease by knowing the expected evolution
(3) Finding the best treatment that will improve the quality of life of PD people

- FOR HOSPITALS:

(1) cost reductions by improved patient management & treatment optimization
(2) operations improvement by predicting patients future visits, med supplies, etc.

CONCLUSION :

Disease diagnosis and prediction is possible through automated machine learning architectures using only non- invasive voice biomarkers as features. Our analysis provides a comparison of the effectiveness of various machine learning classifiers in disease diagnosis with noisy and high dimensional data. After thorough feature selection, clinical level accuracy is possible.

These results are promising because they may introduce novel means to assess patient health and neurological diseases using voice data. Due to the high accuracy performed by the models with these short audio clips there is reason to believe denser feature sets with spoken word, video, or other modalities would aid in disease prediction and clinical validation of diagnosis in the future.

BIBLIOGRAPHY AND REFERENCE :

1] Dataset (https://archive.ics.uci.edu/ml/datasets/Parkinson+Disease+Spiral+Drawings+Using+Digitized+Graphics+Tablet)

2] Algorithms Reference (www.semanticscholar.org),(https://www.ijitee.org)

3] Differential diagnosis of Parkinsons Disease (https://www.sciencedirect.com/science/article/pii/S0933365716000063)