Pynapples

Detecting Parkinson's Using ML

Aditi S Kulkarni, Namita Achyuthan, Soumya LS

Introduction

Parkinson’s is a disease that affects the production of dopaminergic neurons, which can in turn have drastic affects on one’s lifestyle- be it their independence, thinking, or just quality of life in general.

We have built a model to identify the early stages of Parkinson's.

Problem Statement

Identifying Parkinson’s at the early stages by discerning tremors in speech

Dataset

name	MDVP:Fo(	MDVP:Fhi(
phon_R01	119.992	157.302
phon_R01	122.4	148.65
phon_R01	116.682	131.111
phon_R01	116.676	137.871
phon_R01	116.014	141.781

What We Chose

Parkinson's Disease (PD)

Citations: 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

Why We Chose This

Since MRIs cannot discern neuronal loss, we have chosen to work with a dataset which analyzes auditory inputs to look for tremors, a common sign of Parkinson’s.

Scope of the Project

The modern approach to healthcare is to prevent the disease with early intervention rather than go for treatment after diagnosis. These ML models save thousands of lives by allowing patients to seek medical attention at the earliest.

Such models can also be used to diagnose other diseases, thus advancing healthcare greatly.

Architecture

Back to Agenda Page

```
import numpy as np
import pandas as pd
from sklearn import svm #used for classification a
from sklearn.metrics import accuracy_score #accura
from sklearn.preprocessing import StandardScaler #
from sklearn.model_selection import train_test_spl
```

- Numpy for math
- Pandas for datasets
- SVM to build our model
- accuracy_score to predict accuracy
- StandardScaler for standardization
- train_test_split for supervised learning

```
data = pd.read_csv("Parkinsson disease.
data
```

	name	MDVP:Fo(Hz)	MDVP
0	phon_R01_S01_1	119.992	
1	phon_R01_S01_2	122.400	

- Reading our CSV and assigning a variable name

Understanding and Cleaning Data

Back to Agenda Page

```
data.info()
data.describe()
data.shape
```

```
(195, 24)
```

- data.info(): gives us info about the dataset
- data.describe(): gives us important statistical values
- data.shape: rows and columns

```
data.isnull().sum()
```

```
data['status'].value_counts()
#1 implies affected, 0 implies non affected
```

- we do this to find the number of null values
- our target column/variable will be status because it tells us whether someone is affected or not

Getting Ready for Training

Back to Agenda Page

```
x = data.drop(columns = ['name', 'status'],
x
```

```
y = data['status']
y
```

- we're dropping name and status columns because they aren't necessary data for prediction
- Status is our dependent variable

```
x_train, x_test, y_train, y_test =

train_test_split(x, y, test_size= 0.2)
```

- we're doing this to split our data into training and testing

Training Our Model

Back to Agenda Page

```
stanscal = StandardScaler()
x_train = stanscal.transform(x_train)
x_test = stanscal.transform(x_test)
```

- Standardizing all our x test and x train

```
model = svm.SVC(kernel = 'linear')
model.fit(x_train, y_train)
```

- we train our model using svm

Model Evaluation

Back to Agenda Page

```
x_train_pred = model.predict(x_train)
train_data_accu = accuracy_score(y_train, x_train_pred)
train_data_accu
```

```
x_test_pred = model.predict(x_test)
test_data_accu = accuracy_score(y_test, x_test_pred)
test_data_accu
```

- tests the accuracy of our train wrt dependent y training variables
- tests the accuracy of our test wrt dependent y test variables

```
input_data = (120.552,131.162,113.787,0.00968,
input_data_np = np.asarray(input_data)
input_data_re = input_data_np.reshape(1, -1)
s_data = stanscal.transform(input_data_re)
pred = model.predict(s_data)
print(pred)
if(pred[0]==0):
  print("Negative, no Parkinson's")
else:
  print("Positive, you have Parkinson's")
```

- Final testing