18CS3166S – MACHINE LEARNING

PROJECT BASED REPORT

ON

PREDICTION OF PARKINSON'S DISEASE

submitted in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

G. LIKHITHA 180030305

(Batch - 4)

Under the Esteemed Guidance of

Dr. Rudra Kalyan Nayak

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K L E F

Green Fields, Vaddeswaram,

Guntur District – 522 502.

(2020-2021)

K L E F

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is certify that the project based report entitled "**PREDICTION OF PARKINSON'S DISEASE USING GAIT DATASET**" is a bonafide work done and submitted by SAI SOWRI VIKAS(180030271), G.LIKHITHA(180030305), P.V.SAI LATHA(180030350) in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in Department of Computer Science Engineering, KLEF Guntur District during the academic year 2020-2021.

FACULTY INCHARGE

Dr. RUDRA KALYAN NAYAK

HEAD OF THE DEPARTMENT

Dr. HARI KIRAN VEGE

ACKNOWLEDGEMENT

The success in this project would not have been possible but for the timely help and guidance rendered by many people. Our sincere thanks to all those who has assisted us in one way or the other for the completion of my project.

Our greatest appreciation to my guide **Dr. RUDRA KALYAN NAYAK**, Associate Professor, Department of Computer Science and Engineering which cannot be expressed in words for his tremendous support, encouragement and guidance for this project.

We express our gratitude to **Dr. Hari Kiran Vege,** Head of the Department for Computer Science and Engineering for providing us with adequate facilities, ways and means by which we are able to complete this project.

We thank all the members of teaching and non-teaching staff members, and also who have assisted me directly or indirectly for successful completion of this project.

Finally, I sincerely thank my friends and classmates for their kind help and co- operation during our work.

By

G. LIKHITHA - 180030305

INDEX

INTRODUCTION

What is Parkinson’s disease?

Parkinson's disease is a brain disorder that leads to shaking, stiffness, and difficulty with walking, balance, and coordination.

Parkinson's symptoms usually begin gradually and get worse over time. As the disease progresses, people may have difficulty walking and talking. They may also have mental and behavioral changes, sleep problems, depression, memory difficulties, and fatigue.Both men and women can have Parkinson’s disease. However, the disease affects about 50 percent more men than women.

One clear risk factor for Parkinson's is age. Although most people with Parkinson’s first develop the disease at about age 60, about 5 to 10 percent of people with Parkinson's have "early-onset" disease, which begins before the age of 50. Early-onset forms of Parkinson's are often, but not always, inherited, and some forms have been linked to specific gene mutations.

What causes Parkinson’s disease?

Parkinson's disease occurs when nerve cells, or neurons, in an area of the brain that controls movement become impaired and/or die. Normally, these neurons produce an important brain chemical known as dopamine. When the neurons die or become impaired, they produce less dopamine, which causes the movement problems of Parkinson's.

Symptoms of Parkinson’s Disease

Parkinson's disease has four main symptoms:

- Tremor (trembling) in hands, arms, legs, jaw, or head
- Stiffness of the limbs
- Slowness of movement
- Impaired balance and coordination, sometimes leading to falls.

Other symptoms may include depression and other emotional changes; difficulty swallowing, chewing, and speaking; urinary problems or constipation; skin problems; and sleep disruptions.

Diagnosis of Parkinson’s Disease

There are currently no blood or laboratory tests to diagnose nongenetic cases of Parkinson's disease. Diagnosis is based on a person's medical history and a neurological examination.

Medicines for Parkinson's Disease
Medicines prescribed for Parkinson's include:

- Drugs that increase the level of dopamine in the brain
- Drugs that affect other brain chemicals in the body
- Drugs that help control non motor symptoms

Gait Dataset

Gait recognition is a popular pattern recognition problem for which attracts a lot of researchers from different communities such as computer vision, machine learning, biomedical, forensic studying and robotics.This problem also has great potential in industries such as visual surveillance.

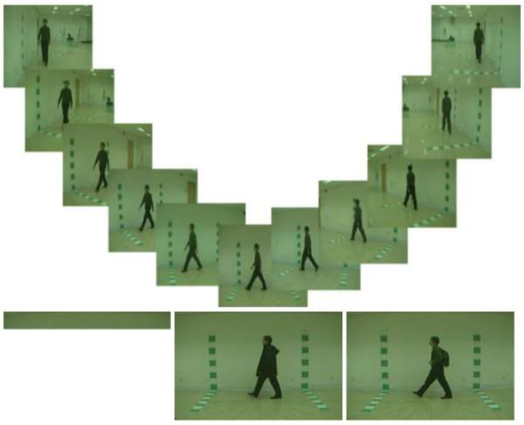
Hence, The Intelligent Recognition & Digital Security Group, which was formed in 1998 by Prof. Tieniu Tan at NLPR (National Laboratory of Pattern Recognition), developed this dataset since Dec. 10, 2001. This Gait Recognition Dataset currently contains 4 subsets: Dataset A(standard dataset)**,** Dataset B (multi view gait dataset) and Dataset C (infrared gait dataset), and Dataset D(gait and its corresponding footprint dataset).

Basically, this dataset in a whole contains the Gaits(style of walking or movement of body) which is helpful for predicting the parkinson's disease.

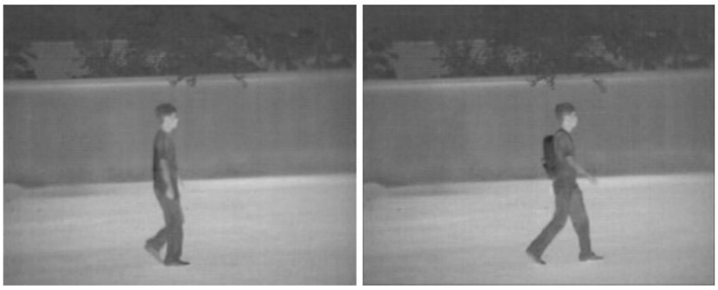
This dataset contains 4 parts.

Dataset A (former NLPR Gait Database) was created on Dec. 10, 2001, including 20 persons. Each person has 12 image sequences, 4 sequences for each of the three directions, i.e. parallel, 45 degrees and 90 degrees to the image plane. The length of each sequence is not identical for the variation of the walker's speed, but it must ranges from 37 to 127. The size of Dataset A is about 2.2GB and the database includes 19139 images.

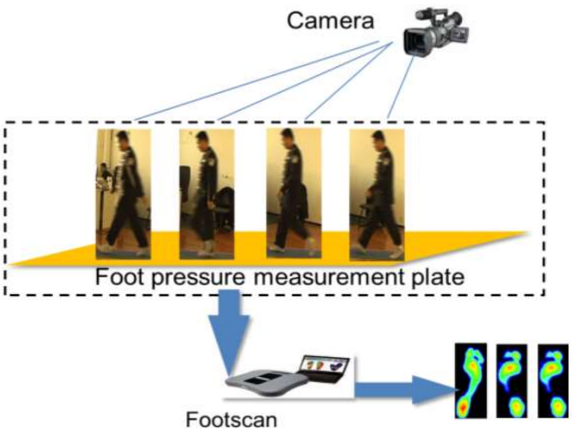
Dataset B is a large multiview gait database, which is created in January 2005. There are 124 subjects, and the gait data was captured from 11 views. Three variations, namely view angle, clothing and carrying condition changes, are separately considered.



Dataset C was collected by an infrared (thermal) camera in Jul.-Aug. 2005. It contains 153 subjects and takes into account four walking conditions: normal walking, slow walking, fast walking, and normal walking with a bag. The videos were all captured at night.



Dataset D was collected synchronously by camera and Rscan Footscan in Jul.-Aug. 2009. It contains 88 subjects and takes into account real surveillance scenes and wide age distribution.



The above mentioned details are about the first dataset created.

This is the information regarding the Parkinson's disease and the dataset (Gait dataset) that we are going to use in this project.

METHODOLOGY

Problem statement:

Prediction of Parkinson's Disease using Gait Dataset.

In this project, we should use Gait Dataset that represents the characteristics of a person i.e, movement of the body. Using this dataset and Machine learning algorithms we can predict if a person has the Parkinson's disease or not.

Here we used Random Forest Classifier algorithm to solve the problem.

Gait Dataset:

Gait recognition is a popular pattern recognition problem for which attracts a lot of researchers from different communities such as computer vision, machine learning, biomedical, forensic studying and robotics. This problem also has great potential in industries such as visual surveillance.

Gait Dataset is a collection of different images of different objects in many angles or directions. Here, in this project, we use humans as the objects. The dataset contains images of humans, each human in various angles or postures.

The original Gait dataset is too large to handle. So, in this project we took a small gait dataset which is easier to handle with. The details regarding Gait dataset are already discussed in the introduction section.

Pre-processing:

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm.
Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

Need of Data Pre-processing:

• For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format,
for example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set.

• Another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one data set, and best out of them is chosen.

Pre-processing involves mainly two steps

Handling Categorical values:
Categorical values are the values which are not numeric. If data contains this type of values the machine learning algorithm that we are using may not give proper results. So, to avoid the categorical values we use **LabelEncoder().**

LabelEncoder() can be used to normalize labels. It can also be used to transform non-numerical labels (as long as they are hashable and comparable) to numerical labels. Transform labels back to original encoding. Transform labels to normalized encoding.

Handling missing values:
Missing values are nothing but the null values or N/A values in the data. These can be identified using functions like isnull() and we use dropna() to remove the missing values in the data.

The **isnull()** function is used to identify the missing values in the data.

The **dropna()** function is used to remove missing values. Determine if rows or columns which contain missing values are removed. 0, or 'index' : Drop rows which contain missing values. 1, or 'columns' : Drop columns which contain missing value.

Here, in the dataset we used the data is already cleaned. So we do not use dropna() function. We just use isnull() to check whether there are any null values. Since there are no null values we don't use dropna().

Feature Description:

In our project the features are –
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), Shimmer:APQ3, Shimmer:APQ5, MDVP:APQ, Shimmer:DDA, NHR, HNR, status, RPDE, DFA, spread1, spread2, D2, PPE.

Some basic terms,

MDVP – Multi Dimensional Voice Program
Jitter - slight irregular movement, variation, or unsteadiness, especially in an electrical signal or electronic device.
Shimmer – A new tool for gait analysis.

Features:
Name – The name of the patient

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 in percentage

MDVP:Jitter(Abs) - the average absolute difference of differences between **jitter** cycles

MDVP:RAP - Five measures of variation in fundamental frequency

MDVP:PPQ - MDVP five-point period perturbation quotient

Jitter:DDP - This is the average absolute difference between consecutive differences between consecutive periods, divided by the average period.

MDVP:Shimmer - MDVP local shimmer

MDVP:Shimmer(dB) - MDVP local shimmer in dB

Shimmer:APQ3 - Three-point amplitude perturbation quotient

Shimmer:APQ5 - Five-point amplitude perturbation quotient

MDVP:APQ11 - MDVP 11-point amplitude perturbation quotient

Shimmer:DDA - Average absolute differences between the amplitudes of consecutive periods

NHR - Noise-to-harmonics ratio

HNR - Harmonics-to-noise ratio

RPDE - Recurrence period density entropy measure

D2 - Correlation dimension

DFA - Signal fractal scaling exponent of detrended fluctuation analysis

Spread1 - Two nonlinear measures of fundamental

Spread2 - Frequency variation

PPE - Pitch period entropy

Status – The output of the data i.e., If the patient has the Parkinson's disease or not.

Algorithm:

In this project, we used Random Forest Classifier Algorithm to predict the parkinson's disease.

Random Forest Classifier:

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest.
Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

For optimization, we used Recursive Feature Elimination With Cross-Validation. This indicates the features which are important with importance ranking. This enables us to build the model with optimal dimensions.

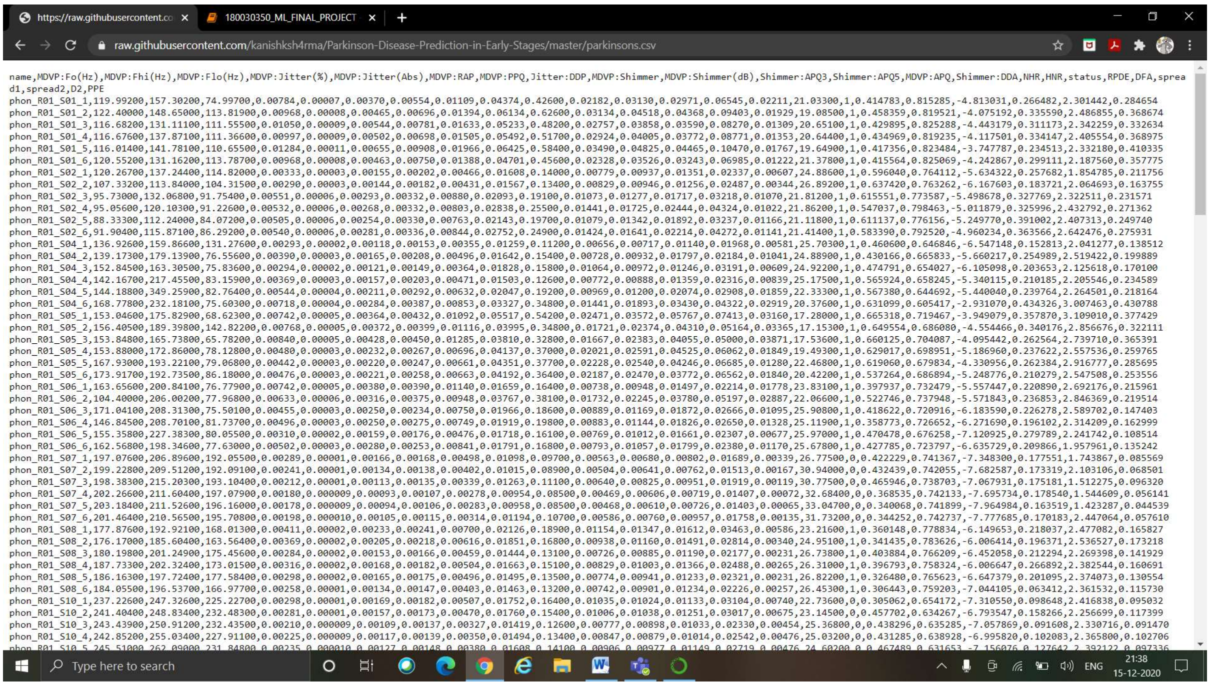
Steps to predict Parkinson's disease using Random forest classifier:

- Importing Libraries
- Loading Dataset
- Pre-processing of data
- Dividing the dataset
- Visualization of data
- Splitting the dataset into training and testing
- Initializing models and fitting the models with data
- Plotting with number of features and selected features
- Predicting the data
- Accuracy and Confusion matrix

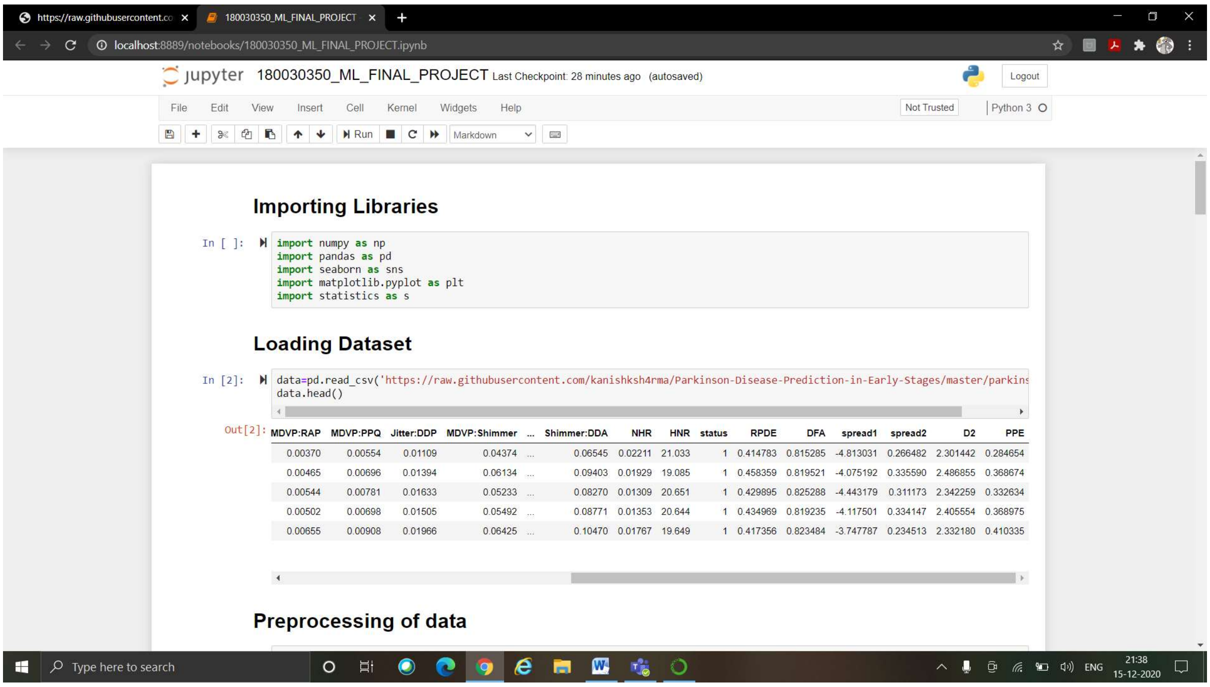
By following these steps we can predict the Parkinson's disease using Gait dataset using the Random Forest Classifier algorithm.

RESULTS AND DISCUSSION

Dataset:



Outputs:



jupyter 180030350_ML_FINAL_PROJECT Last Checkpoint: 28 minutes ago (autosaved)

Preprocessing of data

In [3]:
```
data.columns
```

Out[3]:
```
Index(['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)', 'Shimmer:APQ3', 'Shimmer:APQ5',
       'MDVP:APQ', 'Shimmer:DDA', 'NHR', 'HNR', 'status', 'RPDE', 'DFA',
       'spread1', 'spread2', 'D2', 'PPE'],
      dtype='object')
```

In [4]:
```
data.isnull().sum()
```

Out[4]:
```
name                0
MDVP:Fo(Hz)         0
MDVP:Fhi(Hz)        0
MDVP:Flo(Hz)        0
MDVP:Jitter(%)      0
MDVP:Jitter(Abs)    0
MDVP:RAP            0
MDVP:PPQ            0
Jitter:DDP          0
MDVP:Shimmer        0
MDVP:Shimmer(dB)    0
Shimmer:APQ3        0
Shimmer:APQ5        0
MDVP:APQ            0
Shimmer:DDA         0
NHR                 0
HNR                 0
status              0
RPDE                0
DFA                 0
spread1             0
spread2             0
D2                  0
```

jupyter 180030350_ML_FINAL_PROJECT Last Checkpoint: 28 minutes ago (autosaved)

```
spread1             0
spread2             0
D2                  0
PPE                 0
dtype: int64
```

Dividing the dataset

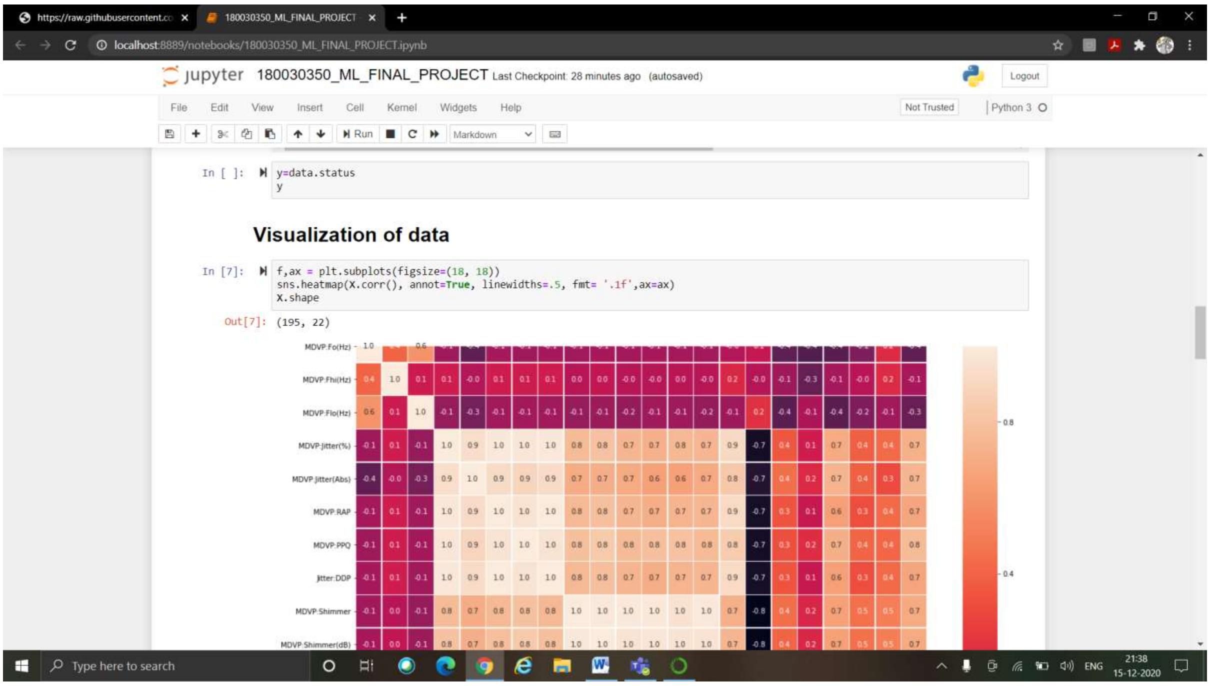
In [6]:
```
list=['name','status']
X=data.drop(list,axis = 1)
X
```

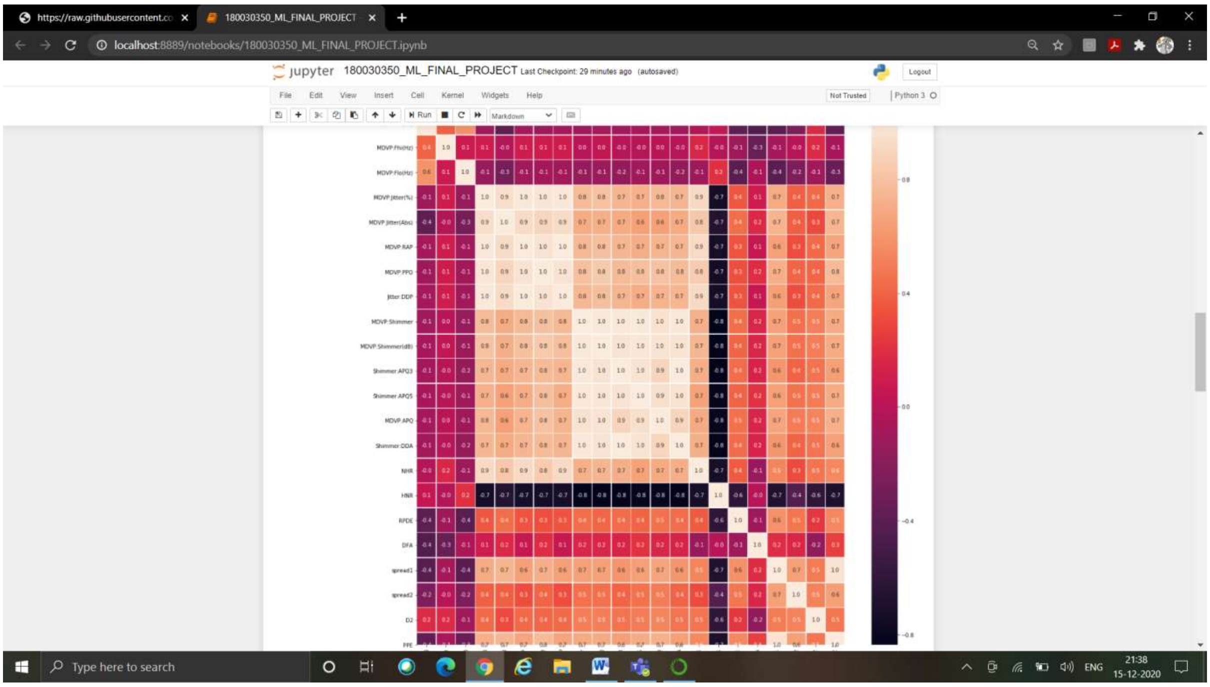
Out[6]:

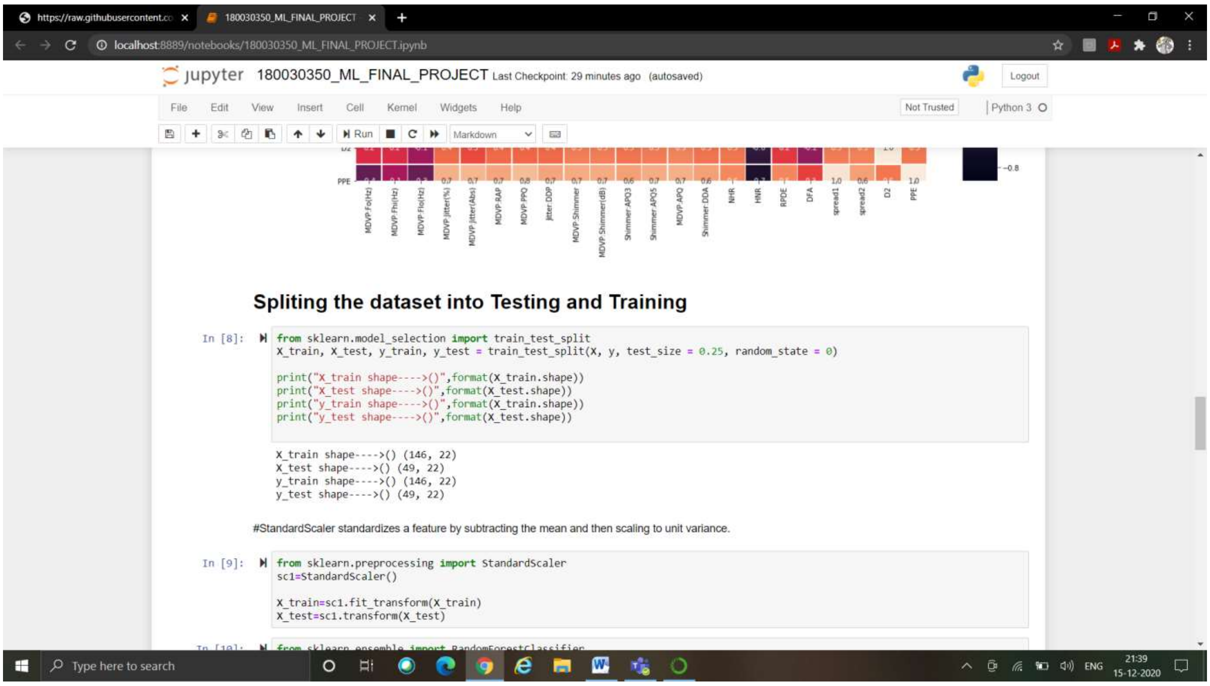
	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	MDVP:Shimmer(dB
0	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	0.426
1	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	0.626
2	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	0.482
3	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	0.517
4	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	0.584
...	...	...	...	...	...	...	...	...	...	...
190	174.188	230.978	94.261	0.00459	0.00003	0.00263	0.00259	0.00790	0.04087	0.405
191	209.516	253.017	89.488	0.00564	0.00003	0.00331	0.00292	0.00994	0.02751	0.263
192	174.688	240.005	74.287	0.01360	0.00008	0.00624	0.00564	0.01873	0.02308	0.256
193	198.764	396.961	74.904	0.00740	0.00004	0.00370	0.00390	0.01109	0.02296	0.241
194	214.289	260.277	77.973	0.00567	0.00003	0.00295	0.00317	0.00885	0.01884	0.190

195 rows × 22 columns

In [ ]:
```
y=data.status
```


https://raw.githubusercontent.co
180030350_ML_FINAL_PROJECT
localhost:8889/notebooks/180030350_ML_FINAL_PROJECT.ipynb
jupyter 180030350_ML_FINAL_PROJECT Last Checkpoint: 28 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
In [ ]:
y=data.status
y
Visualization of data
In [7]:
f,ax = plt.subplots(figsize=(18, 18))
sns.heatmap(X.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
X.shape
Out[7]: (195, 22)
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)
Type here to search
ENG
21:38
15-12-2020


https://raw.githubusercontent.co
180030350_ML_FINAL_PROJECT
localhost:8889/notebooks/180030350_ML_FINAL_PROJECT.ipynb
jupyter 180030350_ML_FINAL_PROJECT Last Checkpoint: 29 minutes ago (autosaved)
Logout
Type here to search
ENG
21:38
15-12-2020



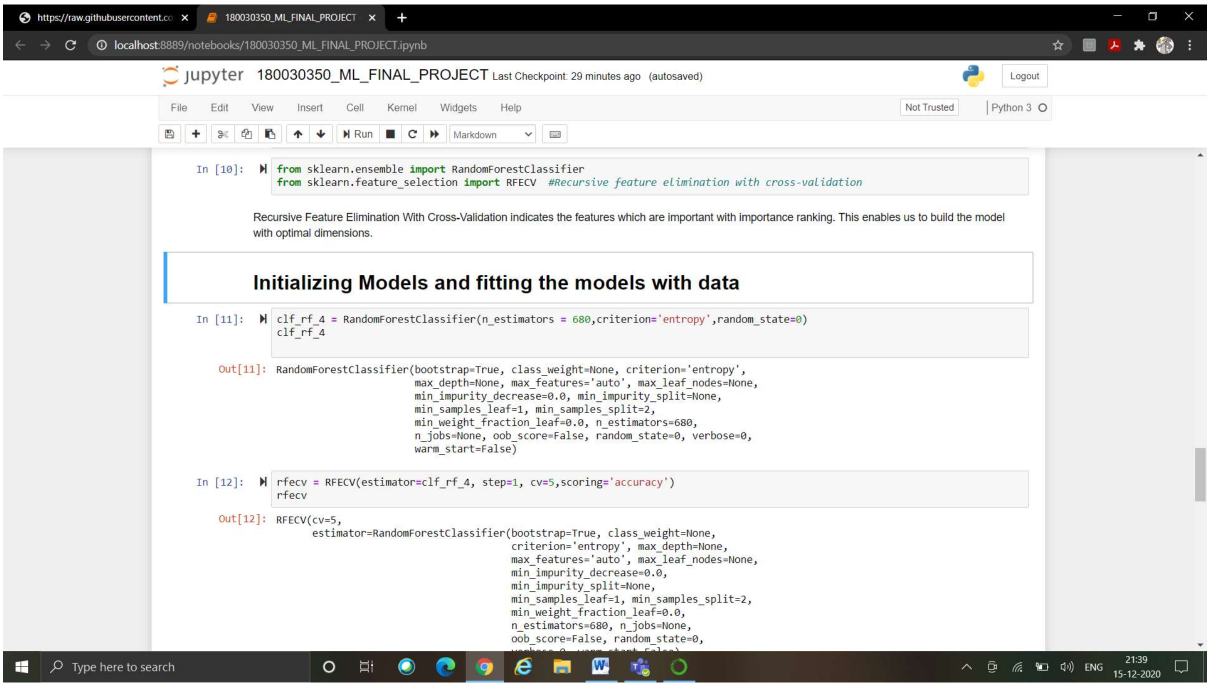
Spliting the dataset into Testing and Training

In [8]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)

print("X_train shape---->()",format(X_train.shape))
print("X_test shape---->()",format(X_test.shape))
print("y_train shape---->()",format(X_train.shape))
print("y_test shape---->()",format(X_test.shape))
```

```
X_train shape---->() (146, 22)
X_test shape---->() (49, 22)
y_train shape---->() (146, 22)
y_test shape---->() (49, 22)
```

#StandardScaler standardizes a feature by subtracting the mean and then scaling to unit variance.

In [9]:
```
from sklearn.preprocessing import StandardScaler
sc1=StandardScaler()

X_train=sc1.fit_transform(X_train)
X_test=sc1.transform(X_test)
```



In [10]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import RFECV  #Recursive feature elimination with cross-validation
```

Recursive Feature Elimination With Cross-Validation indicates the features which are important with importance ranking. This enables us to build the model with optimal dimensions.

Initializing Models and fitting the models with data

In [11]:
```
clf_rf_4 = RandomForestClassifier(n_estimators = 680,criterion='entropy',random_state=0)
clf_rf_4
```

Out[11]:
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='entropy',
                       max_depth=None, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=680,
                       n_jobs=None, oob_score=False, random_state=0, verbose=0,
                       warm_start=False)
```

In [12]:
```
rfecv = RFECV(estimator=clf_rf_4, step=1, cv=5,scoring='accuracy')
rfecv
```

Out[12]:
```
RFECV(cv=5,
      estimator=RandomForestClassifier(bootstrap=True, class_weight=None,
                                       criterion='entropy', max_depth=None,
                                       max_features='auto', max_leaf_nodes=None,
                                       min_impurity_decrease=0.0,
                                       min_impurity_split=None,
                                       min_samples_leaf=1, min_samples_split=2,
                                       min_weight_fraction_leaf=0.0,
                                       n_estimators=680, n_jobs=None,
                                       oob_score=False, random_state=0,
```

```
                                min_weight_fraction_leaf=0.0,
                                n_estimators=680, n_jobs=None,
                                oob_score=False, random_state=0,
                                verbose=0, warm_start=False),
      min_features_to_select=1, n_jobs=None, scoring='accuracy', step=1,
      verbose=0)
```

In [13]:
```
rfecv = rfecv.fit(X_train, y_train)
```

In [14]:
```
rfecv.grid_scores_
```

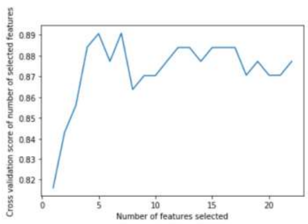
Out[14]:
```
array([0.81604269, 0.84292282, 0.85600985, 0.88408867, 0.89052545,
       0.87719212, 0.89075534, 0.8636289 , 0.87029557, 0.87029557,
       0.87719212, 0.88385878, 0.88385878, 0.87719212, 0.88385878,
       0.88385878, 0.88385878, 0.87052545, 0.87719212, 0.87052545,
       0.87052545, 0.87719212])
```

In [15]:
```
print('Optimal number of features :', rfecv.n_features_)
rfecv.predict_proba(X)

print('Best features :', X.columns[(rfecv.get_support())])
```

```
Optimal number of features : 7
Best features : Index(['MDVP:Fo(Hz)', 'Shimmer:APQ5', 'MDVP:APQ', 'Shimmer:DDA', 'spread1',
       'spread2', 'PPE'],
      dtype='object')
```

Ploting with Number of features and selected features

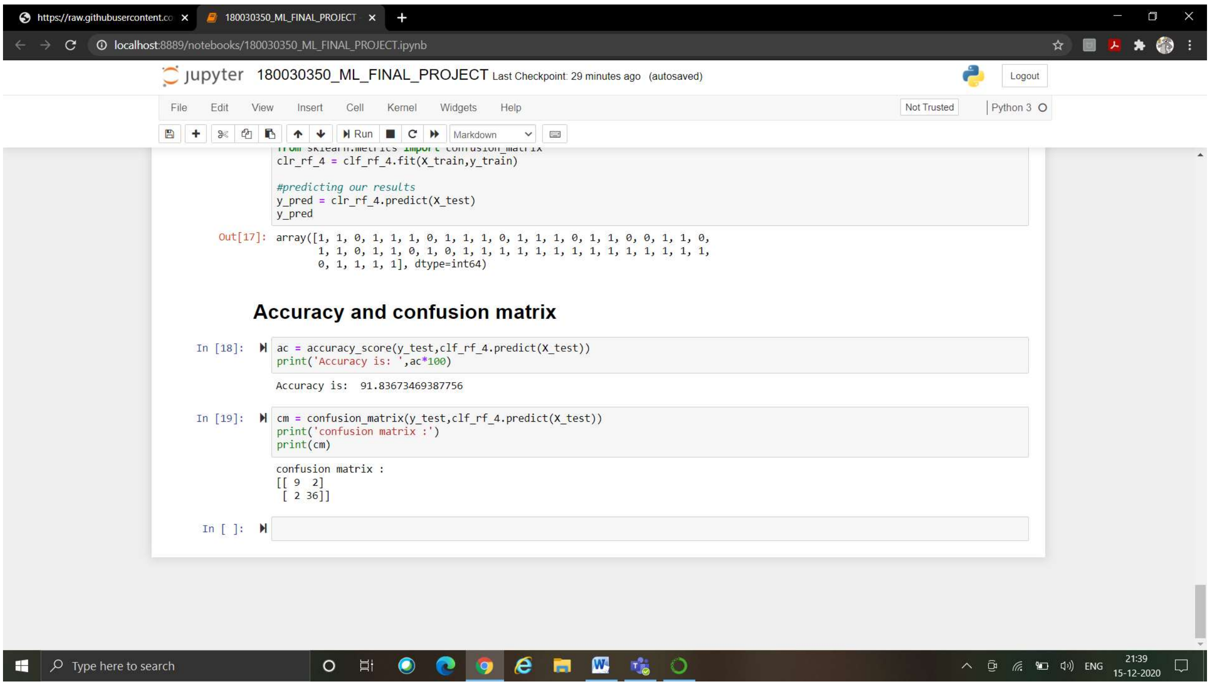
In [16]:
```
import matplotlib.pyplot as plt
plt.figure()
plt.xlabel("Number of features selected")
plt.ylabel("Cross validation score of number of selected features")
plt.plot(range(1, len(rfecv.grid_scores_) + 1), rfecv.grid_scores_)
plt.show()
```



Predicting the data

In [17]:
```
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
clr_rf_4 = clf_rf_4.fit(X_train,y_train)

#predicting our results
y_pred = clr_rf_4.predict(X_test)
y_pred
```

Out[17]:
```
array([1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0,
       1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       0, 1, 1, 1, 1], dtype=int64)
```



If the status is 1 then the patient has Parkinson’s Disease. Otherwise, no disease.

CONCLUSION

In this machine learning project, we learned to detect the presence of Parkinson's Disease in individuals using various factors. We used an RandomForestClassifier for this and made use of the sklearn library to prepare the dataset. This gives us an accuracy of 91%, by using a optimization technique "RFECV"(Recursive feature elimination with cross-validation) which will use the most important features of data.

Our Feature scope is to try this model for bigger dataset (real world example) and improve the model to get the best accuracy.

References:

Information about Parkinson’s disease -

https://www.google.com/search?q=parkinson%27s+disease&oq=parkinson&aqs=chrome.0.35i39i457j69i57j0j0i433l2j69i65j69i60j69i61.3101j0j7&sourceid=chrome&ie=UTF-8

Some more references:

https://www.researchgate.net/publication/304450806_Diagnosis_of_Parkinson's_disease_progress_and_future_prospects

http://www.cbsr.ia.ac.cn/users/szheng/?page_id=71

THANK YOU