L. D. COLLEGE OF ENGINEERING

(Affiliated to Gujarat Technological University)

Navrangpura, Ahmedabad-380015

A Mini Project Report

On

"Parkinson's disease detection"

B.E. IIII, SEMESTER – VII

(COMPUTER ENGINEERING)

1. *Vishrut Goti (180280107034)*
2. *Gajera Sagar (180280107027)*
3. *Dungrani Smit (180280107025)*

Department of Computer Engineering
L. D. College of Engineering
Ahmedabad-380015

October-2021

L. D. College of Engineering

Department of Computer Engineering

2021-22

CERTIFICATE

Date:08/10/2021

This is to certify that the mini project entitled "**Parkinson's disease detection"** has been carried out by **Goti Vishrut (180280107034) Gajera sagar (180280107027) Dungarani Smit (180280107025)** under my guidance in partial fulfillment of the course **Machine Learning (3170724)** in 7th Semester of Bachelor of Computer Engineering during the academic year 2021-22.

Name of Faculty

Prof. Pragnesh Patel
Department of Computer Engineering
L. D. College of Engineering

CANDIDATE’S DECLARATION

We have finished our mini project report entitled “**Parkinson’s disease detection**” and submittedto our respective guide. We are in 7th semester and we have tried to give our best. We have doneour work honestly and in a good way.

First Candidate’s Name : Goti Vishrut

Enrollment No. : 180280107034

Branch : Computer

Second Candidate’s Name : Gajera Sagar
Enrollment No : 180280107027

Third Candidate’s Name : Dungarani Smit

Enrollment No : 180280107025

Submitted to:

L. D. College of Engineering,
Ahmedabad.

Affiliated to: Gujarat Technological University

INDEX

Sr No	Title		Page No
	Acknowledgement		5
	Abstract		5
1	**Introduction**		6
	1.1	Introduction to System	
	1.2	Objective of system	
	1.3	Problem Definition	
2	**Proposed Systems**		
	2.1	Procedure and tools	6
	2.2	Dataset	6
	2.3	Algorithm	7
	2.4	Implementation and result	8
	2.5	Comparison of result with other algorithms	14

Acknowledgement

We would like to extend our heartily thanks with a deep sense of gratitude and respect to all those who has provided us immense help and guidance during our project. We would like to express our sincere thanks to our guide Prof. Pragnesh Patel for providing a vision about the system and for giving us an opportunity to undertake such a great challenging and innovative work. We are grateful for the guidance, encouragement, understanding and insightful support given in the development process. Last but not the least we would like to mention here that we are greatly indebted to each and everybody who has been associated with our project at any stage but whose name does not find a place in this acknowledgement.

Abstract

In this project we predict whether any person has Parkinson's disease or not. The prediction is based on patient's data like frequency of their sound, amplitude, etc. We have used support vector machine algorithm and logistic regression algorithm of classification model.

Introduction to System:

- ► In this project we predict that someone is infected by Parkinson’s disease or not by analyzing other features like patient’s sound frequency, amplitude, etc.

Objective of the system:

- A chronic and progressive movement disorder that initially causes tremor in one hand, stiffness or slowing of movement.

- Parkinson's disease is a nervous system disorder .

- Patient's condition gets worst over a time .Therefor patient needs to be treated as soon as possible.

- Machine learning will help us to determine whether someone is affected by Parkinson's disease or not. Patient gets an early treatment.

Procedure and tools:

Hardware: Laptop, RAM, GPU
Software: Google Colaboratory

Steps:

1) Data loading using pandas library.

2) Data size detemination and getting more information about the data set.

3) Checking for missing values in each column.

4) Getting some statistical measures about the data like mean , count ,std , etc.

5) Determining the distribution of target Variable and grouping the data based on the target variable .

6) Separating features and target variable .

7) Splitting the data into training and testing data.

8) Standarization.

Dataset:

- The data set is collected from kaggle.com/nidagular/parkinsondetection.

-The data set was created by Max Little of the University of Oxford, in collaboration with the National Center for Voice and Speech, who recorded the speech signals.

- This data set contains 195 rows. Among which 147 have positive result and 48 have negative result for Parkinson's disease.

- The data set has 23 attributes and most of them are various sound frequencies collected from patient.

-Status attribute plays as an output in this data.

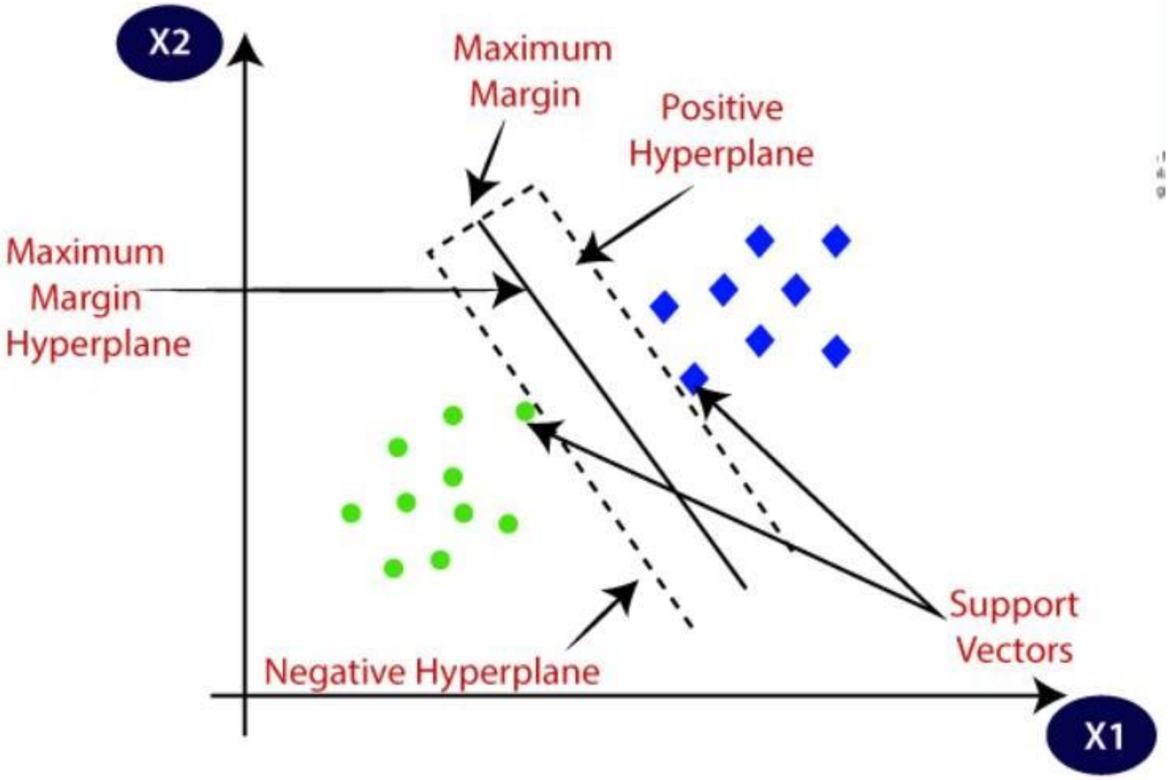
MDVP:Fo(Hz) - Average vocal fundamental frequency
MDVP:Fhi(Hz) - Maximum vocal fundamental frequency
MDVP:Flo(Hz) - Minimum vocal fundamental frequency
MDVP:Jitter(%),MDVP:Jitter(Abs),MDVP:RAP,MDVP:PPQ,Jitter:DDP - Several measures of variation in fundamental frequency
MDVP:Shimmer,MDVP:Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5,MDVP:APQ,Shimmer:DDA - 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
DFA - Signal fractal scaling exponent
spread1,spread2,PPE - Three nonlinear measures of fundamental frequency variation

Algorithm:

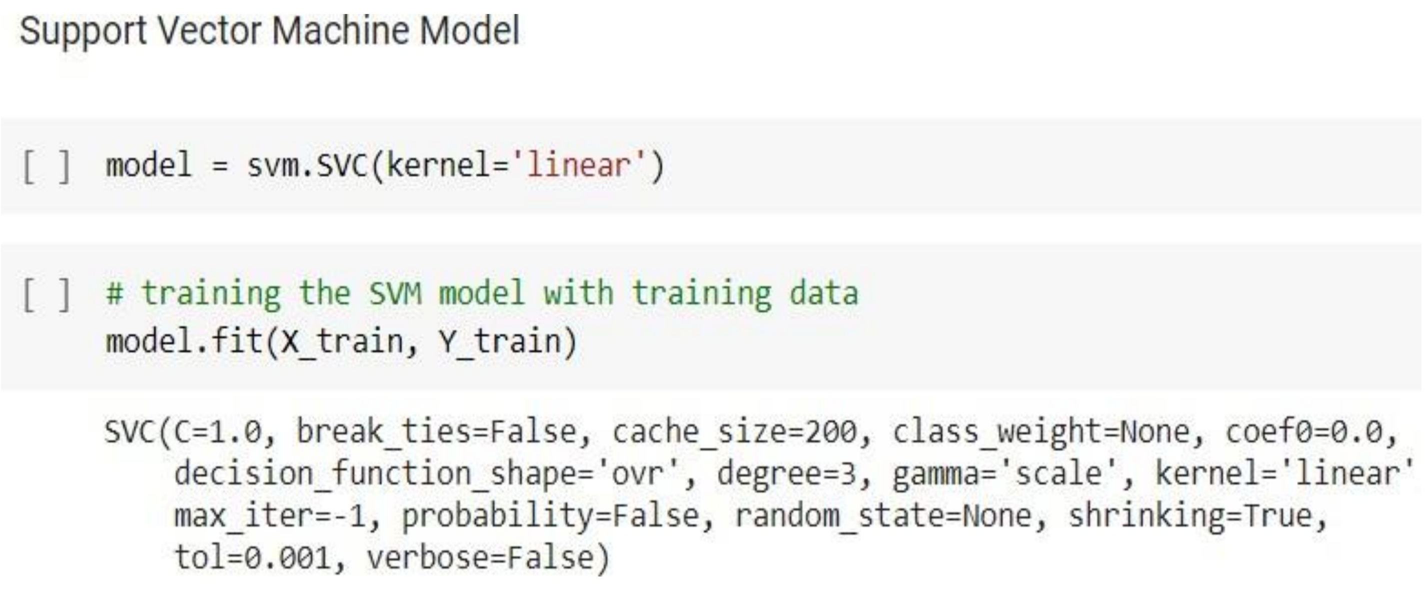
- SVM stands for Support Vector Machine. SVM is a supervised machine learning algorithm that is commonly used for classification and regression challenges.

- Support Vector Machine algorithm is mainly used to solve classification problems. Support vectors are nothing but the coordinates of each data item. Support Vector Machine is a frontier that differentiates two classes using hyper-plane.

-The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.


X2
Maximum Margin
Positive Hyperplane
Maximum Margin Hyperplane
Support Vectors
Negative Hyperplane
X1

Implementation:-

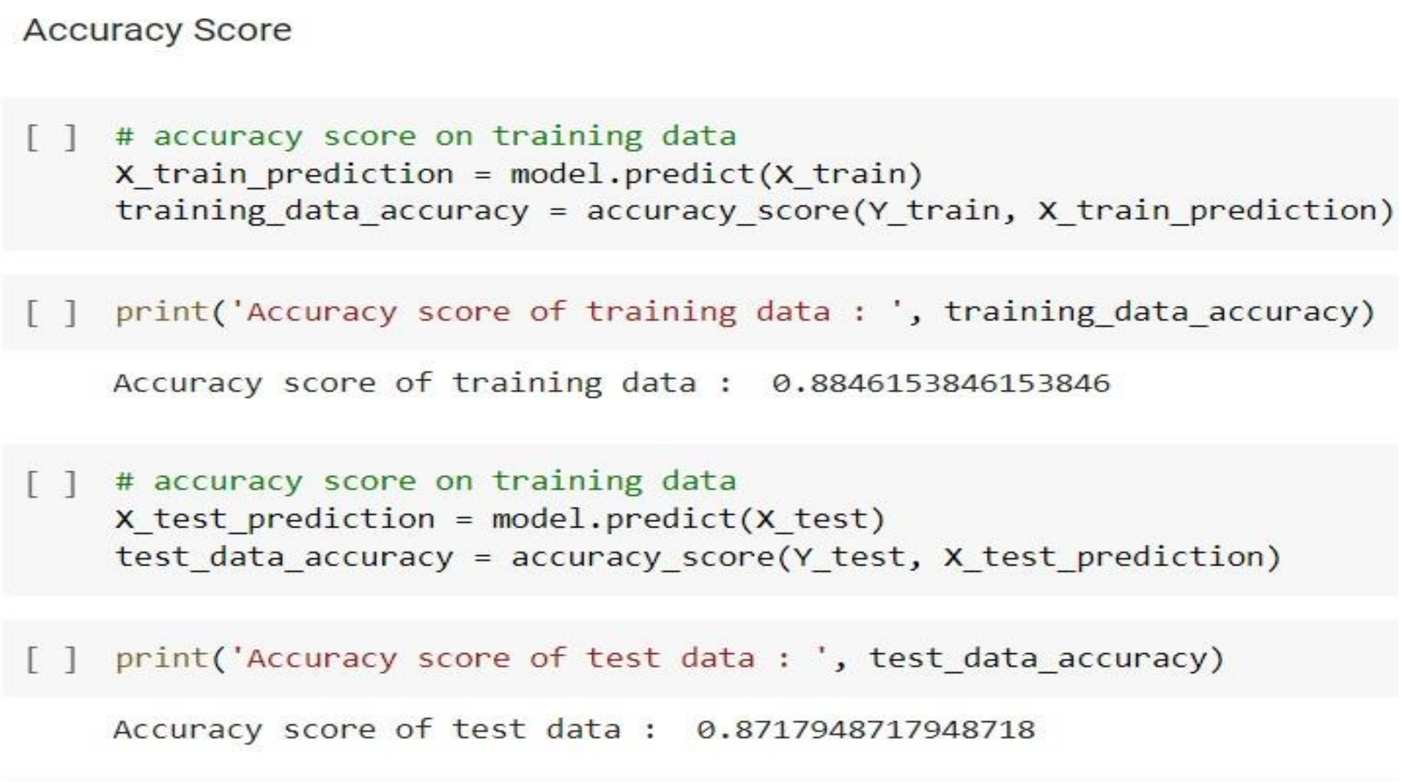


Support Vector Machine Model

```
[ ]  model = svm.SVC(kernel='linear')
```

```
[ ]  # training the SVM model with training data
     model.fit(X_train, Y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear'
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```



Accuracy Score

```
[ ]  # accuracy score on training data
     X_train_prediction = model.predict(X_train)
     training_data_accuracy = accuracy_score(Y_train, X_train_prediction)
```

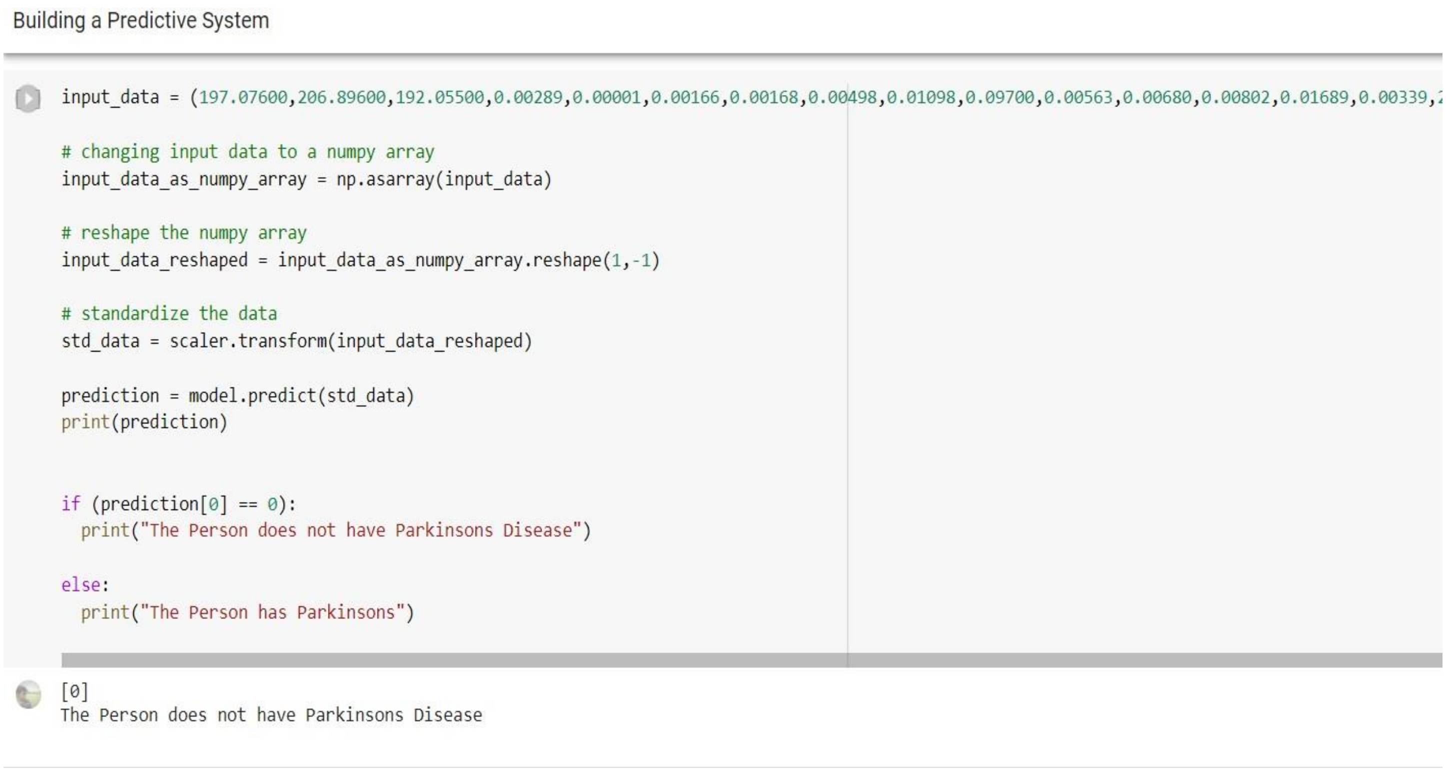
```
[ ]  print('Accuracy score of training data : ', training_data_accuracy)
```

```
Accuracy score of training data :  0.8846153846153846
```

```
[ ]  # accuracy score on training data
     X_test_prediction = model.predict(X_test)
     test_data_accuracy = accuracy_score(Y_test, X_test_prediction)
```

```
[ ]  print('Accuracy score of test data : ', test_data_accuracy)
```

```
Accuracy score of test data :  0.8717948717948718
```



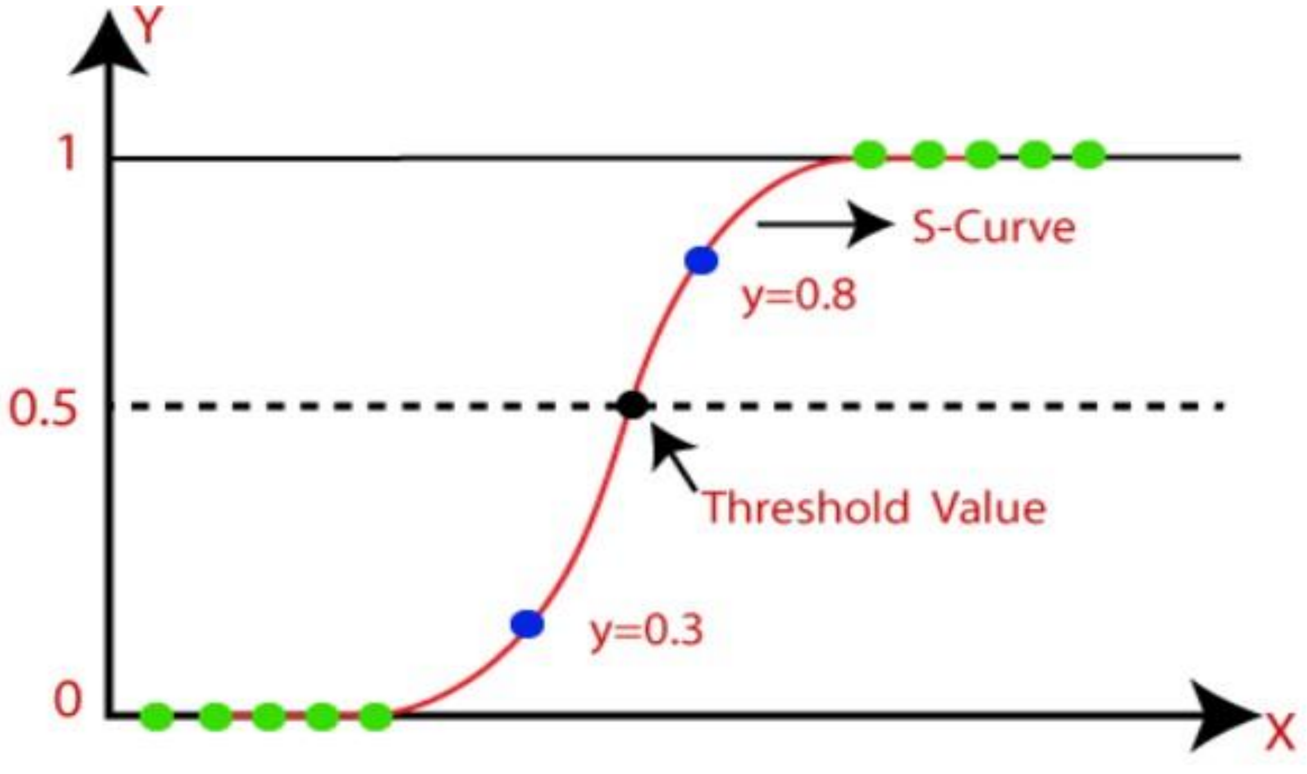
Building a Predictive System

```
input_data = (197.07600,206.89600,192.05500,0.00289,0.00001,0.00166,0.00168,0.00498,0.01098,0.09700,0.00563,0.00680,0.00802,0.01689,0.00339,2

# changing input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the numpy array
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

# standardize the data
std_data = scaler.transform(input_data_reshaped)

prediction = model.predict(std_data)
print(prediction)


if (prediction[0] == 0):
  print("The Person does not have Parkinsons Disease")

else:
  print("The Person has Parkinsons")
```

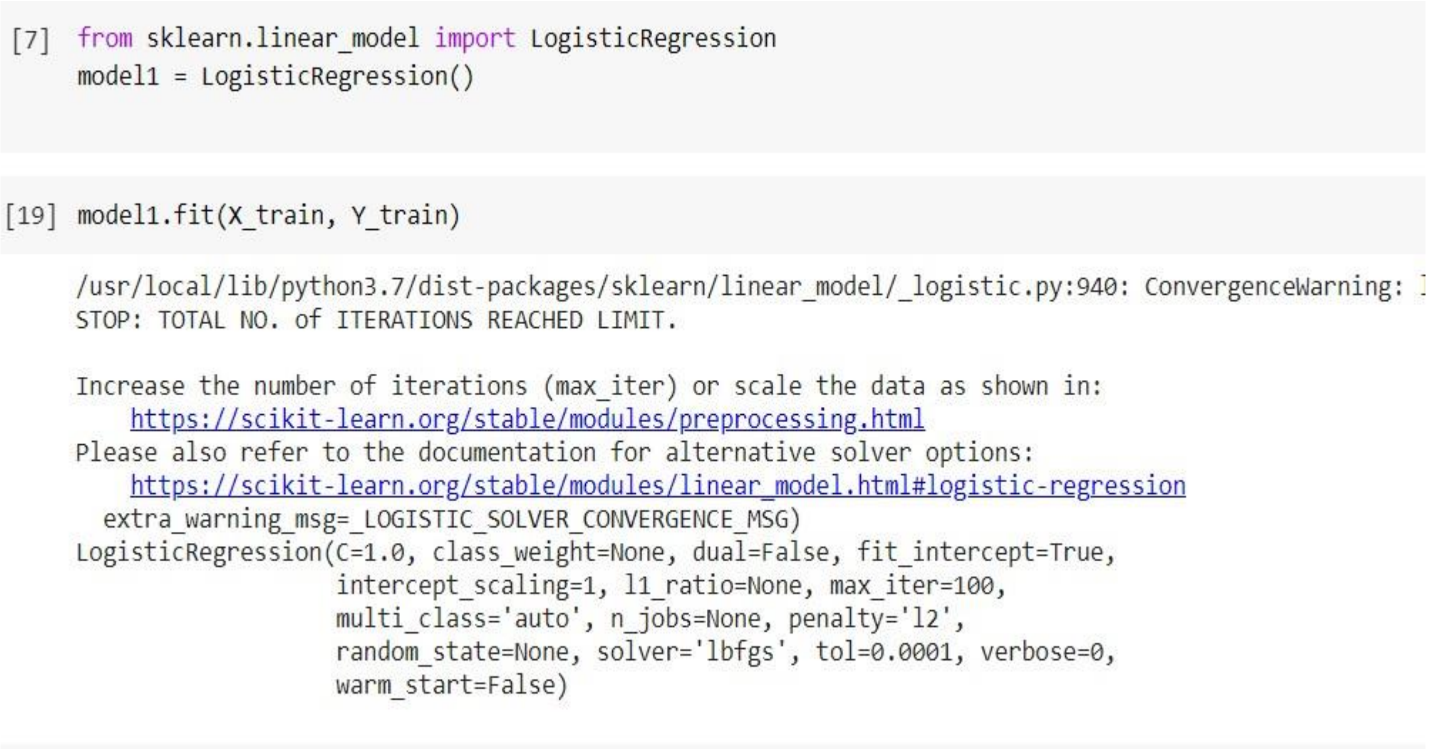
```
[0]
The Person does not have Parkinsons Disease
```

Algorithm:

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

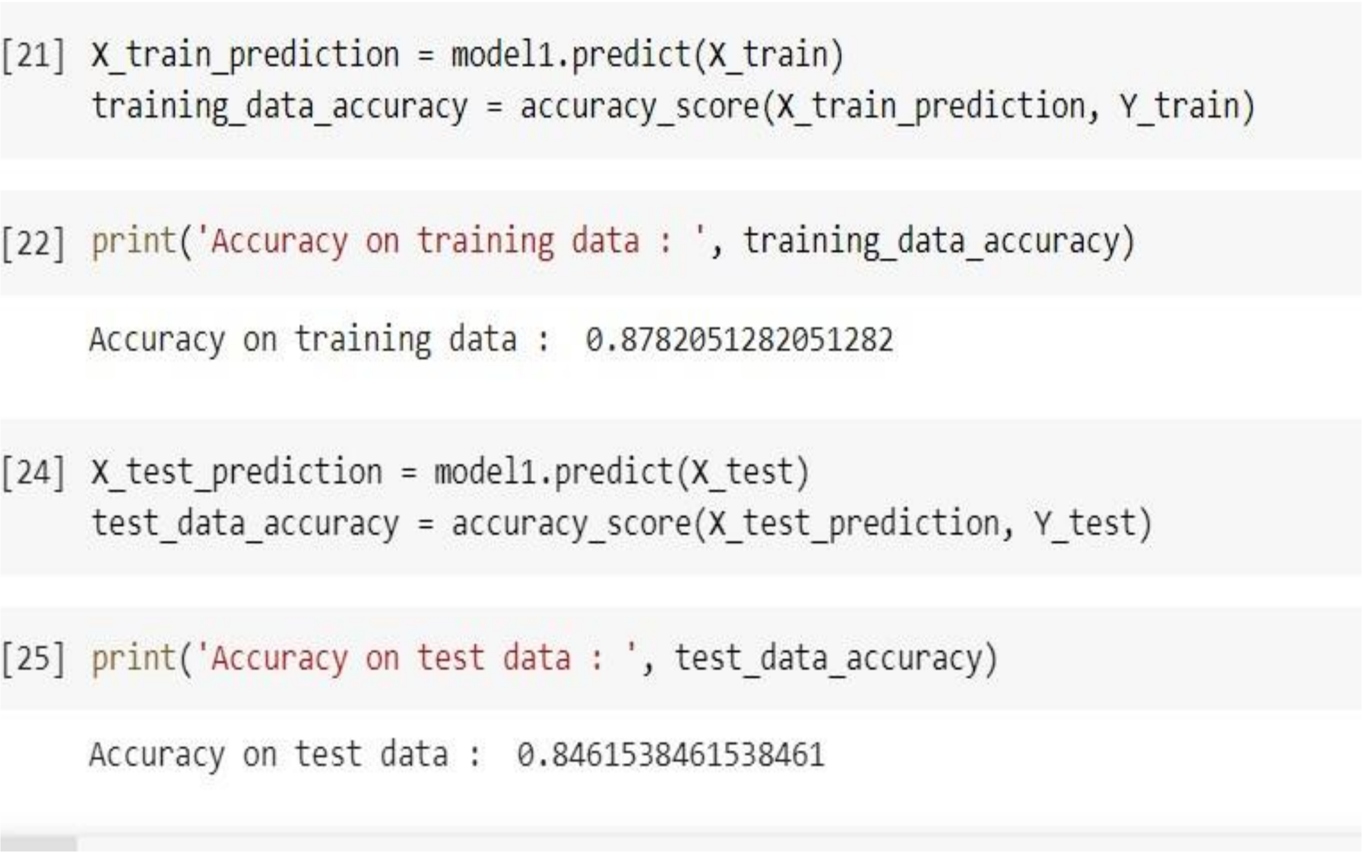


Implementation:-



```
[7]  from sklearn.linear_model import LogisticRegression
     model1 = LogisticRegression()
```

```
[19] model1.fit(X_train, Y_train)

     /usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:940: ConvergenceWarning: l
     STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

     Increase the number of iterations (max_iter) or scale the data as shown in:
         https://scikit-learn.org/stable/modules/preprocessing.html
     Please also refer to the documentation for alternative solver options:
         https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
       extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
     LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                        intercept_scaling=1, l1_ratio=None, max_iter=100,
                        multi_class='auto', n_jobs=None, penalty='l2',
                        random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                        warm_start=False)
```



```
[21] X_train_prediction = model1.predict(X_train)
     training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

```
[22] print('Accuracy on training data : ', training_data_accuracy)

     Accuracy on training data :  0.8782051282051282
```

```
[24] X_test_prediction = model1.predict(X_test)
     test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
[25] print('Accuracy on test data : ', test_data_accuracy)

     Accuracy on test data :  0.8461538461538461
```

Comparison with another algorithm:

- In this project comparison between linear regression and support vector machine is done and comparison is done by accuracy factor.
- We have a limited amount of data and that is why both models give nearly the same accuracy over test and training data.
- I genuinely believe that logistic regression model will work more precisely on larger dataset because of its probabilistic nature.

Conclusion:

This project can predict whether someone is infected by Parkinson or not so that patient can get an early treatment.