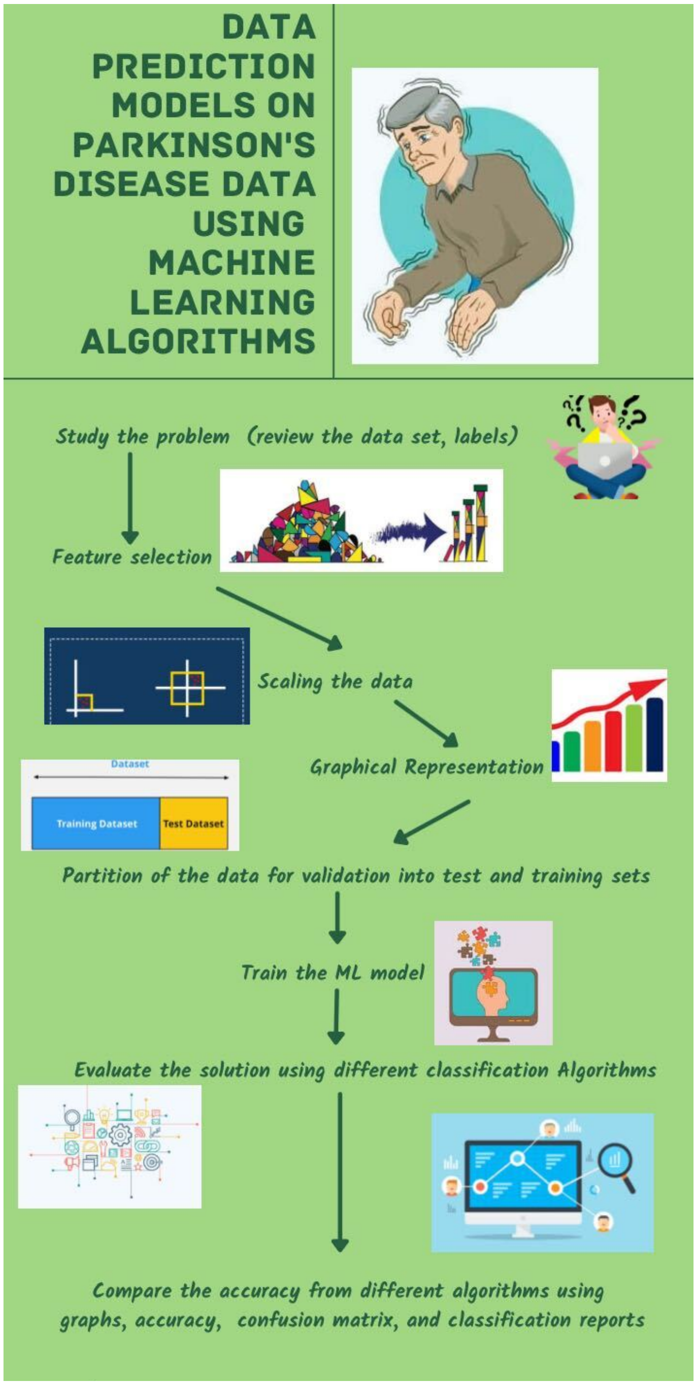
Team Data Science:

Objective: Data Prediction Models on Parkinson’s disease Data using Machine Learning Algorithms in Python.

We started with the the tutorial on XGBoost algorithm given here - Link

Basic Workflow:



DATA PREDICTION MODELS ON PARKINSON'S DISEASE DATA USING MACHINE LEARNING ALGORITHMS

Study the problem (review the data set, labels)

Feature selection

Scaling the data

Graphical Representation

Partition of the data for validation into test and training sets

Train the ML model

Evaluate the solution using different classification Algorithms

Compare the accuracy from different algorithms using graphs, accuracy, confusion matrix, and classification reports

The approach to handle these problems is to get to know more about the data, it's features and finding the best fitting algorithms.

ABOUT THE DATA :

Parkinson's disease is a progressive disorder of the central nervous system affecting movement and inducing tremors and stiffness. It has 5 stages. This is chronic and has no cure yet. It is a neurodegenerative disorder affecting dopamine-producing neurons in the brain.

We took the UCI ML Parkinsons dataset for this. The dataset has 24 columns and 195 records and is only 39.7 KB.

Data Set Characteristics: Multivariate
Number of Instances: 197
Area: Life
Attribute Characteristics: Real
Number of Attributes: 23
Date Donated: 2008-06-26
Associated Tasks: Classification
Missing Values? N/A

Source:

The dataset was created by Max Little of the University of Oxford, in collaboration with the National Centre for Voice and Speech, Denver, Colorado, who recorded the speech signals. The original study published the feature extraction methods for general voice disorders.

Data Set Information:

This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each column in the table is a particular voice measure, and each row corresponds to one of 195 voice recordings from these individuals ("name" column). The main aim of the data is to discriminate healthy people from those with PD, according to the "status" column which is set to 0 for healthy and 1 for PD.

The data is in ASCII CSV format. The rows of the CSV file contain an instance corresponding to one voice recording. There are around six recordings per patient, the name of the patient is identified in the first column.
Further details are contained in the following reference -- if you use this dataset, please cite:
Max A. Little, Patrick E. McSharry, Eric J. Hunter, Lorraine O. Ramig (2008), 'Suitability of dysphonia measurements for telemonitoring of Parkinson's disease', IEEE Transactions on Biomedical Engineering (to appear).

Attribute Information:

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
DFA - Signal fractal scaling exponent
spread1,spread2,PPE - Three nonlinear measures of fundamental frequency variation

PYTHON PACKAGES USED:

Numpy: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Pandas: pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

Scikit-learn: Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Seaborn, Matplotlib: Seaborn and Matplotlib are two of Python's most powerful visualization libraries. Seaborn uses fewer syntax and has stunning default themes and Matplotlib is more easily customizable through accessing the classes.

Os, warnings: The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules.

1. Data Preprocessing

Data is loaded, features and label variables are separated out and whole data is TRANSFORMED.

Data is transformed to make it better-organized. Transformed data may be easier for both humans and computers to use. Properly formatted and validated data improves data quality and protects applications from potential landmines such as null values, unexpected duplicates, incorrect indexing, and incompatible formats.

Splitting the data into train and test sets is done by the function present in sci-kit learn library. The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. It is a fast and easy procedure to perform, the results of which allow you to compare the performance of machine learning algorithms for your predictive modeling problem. Although simple to use and interpret, there are times when the procedure should not be used, such as when you have a small dataset and situations where additional configuration is required, such as when it is used for classification and the dataset is not balanced.

2. Analysis using Models

Initially we used the XGBoost algorithm for model preparation. And following we used different algorithms just to compare the efficiency of the models built and which one would be most accurate. Let's look at different models used:

- XGBOOST

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks.

- Logistic Regression

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

- Support Vector Machine

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. They are extremely popular because of their ability to handle multiple continuous and categorical variables. An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).

- Gaussian Naive Bayes

Naïve Bayes algorithm is a classification technique based on applying Bayes' theorem with a strong assumption that all the predictors are independent of each other. In simple words, the assumption is that the presence of a feature in a class is independent of the presence of any other feature in the same class.

- K-Neighbour Classifier

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification of predictive problems in industry. The following two properties would define KNN well −
Lazy learning algorithm − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
Non-parametric learning algorithm − KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data.

- Random Forest Classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

- Decision Tree Classifier

Decision tree analysis is a predictive modelling tool that can be applied across many areas. Decision trees can be constructed by an algorithmic approach that can split the dataset in different ways based on different conditions. Decisions trees are the most powerful algorithms that falls under the category of supervised algorithms. They can be used for both classification and regression tasks. The two main entities of a tree are decision nodes, where the data is split and leaves, where we get the outcome.

- Bagging Classifier

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.

3. Classification reports and Visualization of results.

To measure the performance of each model, we accounted the following parameters:

Accuracy: Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of the number of correct predictions to the total number of input samples.

Confusion Matrix: Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model. There are 4 important terms :

- True Positives : The cases in which we predicted YES and the actual output was also YES.
- True Negatives : The cases in which we predicted NO and the actual output was NO.
- False Positives : The cases in which we predicted YES and the actual output was NO.
- False Negatives : The cases in which we predicted NO and the actual output was YES.

AUC-ROC curve: Area Under Curve(AUC) is one of the most widely used metrics for evaluation. It is used for binary classification problems. AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example.

Comparing accuracy of all the models:

MODEL	ACCURACY
XGBoost	94.87179487179486
Logistic Regression	89.83050847457628
Support Vector Machine	84.7457627118644
Guassian Naive Bayes	77.96610169491525
K Neighbour	86.4406779661017
Random Forest	93.22033898305084
Decision Tree	84.7457627118644
Bagging	89.83050847457628

By looking at this table, we can conclude that XGBoost is the most accurate.

Contribution :-

S. No	Name	Slack Username	Contribution	Partner (If any)
1	Sanniya Middha	@Sanniya01	Logistic Regression	Rachna Behl (@Rachna)
2	Sophie Fang	@Sophie	Random Forest, Heat map, Feature importance map, Analysis of plot differences	
3	Shalini Gupta	@Miss_Indori Delight	Logistic Regression,K-Nearest Neighbors(Including Scatter Plot) and Support Vector	Bhavya Saini (@Bhavyasind)
4	Shruti Poojary	@ShrutiP	Random Forest (including FacetGrid plots,Workflow Advertisement designing)	@Dibyendu1153533
5	Bhushan Wagh	@XR2	Logistic Regression, K-Nearest Neighbors,Gaussian Naïve Bayes, Support Vector,Stacking,Decision Tree,Bagging,Random Forest(Including Scatter Plot) Histograms,Box,Bar,Pair Plot,HeatMap	
6	Ikechukwu Okoye	@Ikechukwu	Support Vector and Naive Bayes Classifier with necessary accuracy metrics and plots	@Mercii
7	Arinola	@Arinola	Wrote out the team's project protocol for the advertisement submission. Logistic linear regression	@Dibyendu1153533
8	David Guevara-Apaza	@yoodavoo	EDA, pie chart, correlation and heatmap, histograms, Bagging Algorithm, feature importance, ROC	@Chukwu_emeliela
9	Prathamesh Bobale	@Pratham99	Scatter plot, logistic regression	
10	Dibyendu Biswas	@Dibyendu1153533	Workflow Advertisement designing, Logistic linear regression	@ShrutiP
13	Aginah Chukwuemelie	@Chukwu_emeliela	EDA, pie chart, correlation and heatmap, histograms, Bagging Algorithm, feature importance, ROC	David Guevara-Apaza

14	Anirudh	@anirudh1009	Gaussian Naive Bayes, K-Nearest neighbour	
15	Foluso Ogunfile	@fogunfile	Logistic regression, nearest centroid classifier	
16	Bhavya Saini	@Bhavyasind	Logistic & K-Nearest Neighbors : Scatter Plot Write up	Algorithm with Shalini Gupta
17	Vinay Joshi	@vinyjoshi	Github Repo, All algorithms, Code for Markdown.	