Comparison Study of Different Classifiers for Detecting Parkinson Disease using Machine Learning Language

Jerry K Thomas[1*], Syama R[2]

[1]Electronics and communication engineering, College of Engineering, Kidangoor Kottayam, India

[2]Assistant professor, Department of ECE College of Engineering, Kidangoor, Kottayam, India

*Corresponding author's e-mail: jerrykakkassery4@gmail.com

doi: https://doi.org/10.21467/proceedings.160.23

ABSTRACT

In Parkinson's disease, dopamine-producing neurons in the brain are disrupted. Communication between brain cells is enabled by it. Dopamine is responsible for the control, adaption and easiness of movements. This disease occurs mainly in aged persons, but in this current scenario Parkinson's disease symptoms showing after age 35. So researchers try to find more ways to recognize the symptoms of Parkinson's disease as early as possible. The purpose of this paper is to present different classifiers that use machine learning to diagnose Parkinson's disease. Here I use 3 different classifiers-SVM, KNN and XGBoost. I will build a model for all 3 classifiers and calculate their accuracy of detecting the Parkinson's disease by giving the same dataset input. From the 3 classifiers I select the more accuracy one. This paper propose select the XGBoost classifiers to detect the Parkinson's disease person. XGBoost gives 94.7.

Keywords: Extreme Gradient Boosting (XGBoost), k-Nearest Neighbour (KNN), Support Vector Machine (SVM)

1 Introduction

Globally, hospitals and many other health centers are then split the dataset for testing and training the model. We will increasingly using healthcare monitoring systems with emerging built an XGBclassifier and then detect the parkinson's disease. technologies, which are becoming a major concern to many countries. As a result of modern technology, humans can lead a smarter life by using flexible interfaces, assistant devices, and mental health management. Parkinson's disease was one of the neurodegenerative disorder that progresses and selectively affects cells in the central nervous system. This degeneration primarily affects dopaminergic neurons in the substantiate nigra pars compacta, which leads to impaired mobility and other motor manifestations such as rigidity, tremor, and slowed movement. Some patients with Parkinson's disease develop other disorders called non-motor symptoms complex. These can include cognitive difficulties in the muscle, symptoms such as depression, higher anxiety, hallucinations, decreased blood pressure, constipation, problem in swallowing, increased sweating, problems sleeping and loss of smell. Here we use machine learning algorithim to detect parkinson's disease.in here 3 different classifiers is used to detect parkinson's disease. They are k-Nearest Neighbour (kNN), Support Vector Machine (SVM) and XGBoost. Machine learning algorithms are used to make inferences from existing data. These inferences can be defined as the estimation of unknown values in any subject. The most basic pattern recognition and classification method is k-NN, which classifies the closest educational examples in the objects based on attribute space. SVM is a Support Vector Machine classifier in which classification is ready by creating a hyperplane of the data that can classify data points distinctly. XGBoost is put an end to scalable tree boosting Technique. Our goal is to develop a model well enough so that it can find the early stage of Parkinson's Disease. In this project, we will build 3 model using XGBoost, SVM, KNN among these classifiers we select more accurate one and use it for detecting parkinson's disease. In this proposed system I used the Python libraries such as numpy, scikit learn First, we will update the data. After that we get the output as features and labels. We will measure the features [1], [4].

Proceedings DOI: 10.21467/proceedings.160; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-965621-9-9

2 Methodology

A model that can find Parkinson's disease early on is outlined in Fig. 1.

2.1 Block Diagram

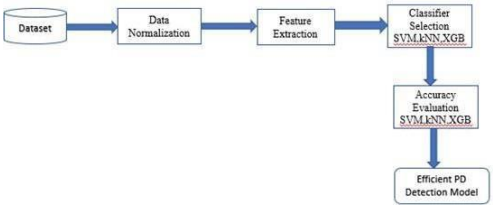

Figure 1: *Proposed model framework*

The details of the proposed framework in Figure 1 have been described below individually.

2.2 PD Dataset

The first step in the classification process is collecting data. For the voice analysis, a machine learning repository is used which includes the voice data from both PD and healthy people. The dataset used consists of 757 instances and 755 attributes. It was collected from 189 PD patients in which 107 were men, 82 were women. The healthy individuals in which 41 women and 23 men. Data collection was accomplished by performing 3 repetition of sustained phonation [1].

2.3 Performance Metrics for Model Evaluation

The model is builded and outputs results in the form of probability or in a class format. The 2nd step is to discover how well the model performs using the data based on metrics. Choosing the right metrics to valuate the machine learning model is more important as it depending how performance is calculated and compared.

2.3.1 Confusion Matrix

A confusion matrix is a table with two dimensions.it contains one for the target value and one for the predicted value. To explain the idea of the confusion matrix, assume that you are working on a binary classification problem in which there are two classes: 1 and 0. The actual labels are shown in rows and the predicted labels are shown in columns. The confusion matrix will tell you how well each model predicts each label. The closer two models are to each other in terms of their predictions, the more accurate they are.

2.4 Machine Learning Algorithim

Machine learning is a process of inference from existing data. This involves calculating unknown values in a subject, and the machine learning algorithms that I used in this proposed are explained below [5], [6].

2.4.1 k-Nearest Neighbour (kNN)

The k-NN algorithm is a pattern recognition and classification method that uses Euclidean, Manhattan, Chebyshev, and Angular distance measurement are some methods to classify the objects. The common formulas and details of the k-NN algorithm are described below. k-NN thesis was first comes to light in 1967 by Cover and Hart. It is one of the simplest pattern recognition and classification ways that sort out the objects based on the nearest educational observations in the attribute space. There are different ways for measuring the distance of a new sample from the classified samples. The Euclidean distance calculation method is commonly used by everyone. After computing the neighbour distances based on all of the outcomes in a new sample's data set, it is determined which class to belong to based on the neighbourhood

Proceedings DOI: 10.21467/proceedings.160
ISBN: 978-81-965621-9-9

Series: AIJR Proceedings
ISSN: 2582-3922

status. As a result, because there are more triangles than k = 3, the class of the updated sample is also set on as triangle. Result shown in figure 2 [1].

```
from sklearn.metrics import confusion_matrix,accuracy_score
cm=confusion_matrix(y_test,y_pred)

(accuracy_score(y_test,y_pred)*100)
```

```
89.74358974358975
```

Figure 2: *Output of the kNN Classifier*

2.4.2 Support Vector Machine (SVM)

It is a supervised machine learning technique in which a hyperplane is created to group data points and it can be used to distinguish between them. For that we need to calculat the optimal margin for this hyperplane, which is an advantage of the SVM classifier.

This tool is effective in high-dimensional data, so it's being explored with different kernels. The cost function determines how much weight should be given to each data point, and gamma (auto) sets the default value. The SVM algorithm was designed specifically to solve the duple classification problems. To separate the classes from each other effectively a hyperplane is obtained. The classification is commonly characterised by class labels 1 and +1. The data to be classified can be linearly separated or not at all. In the real world many classification problems consist of more than two classes. To avoid these problems, multi-class SVM classifier used genrally. It can be ready by 2 or more combining duple classifiers. Result shown in figure 3 [4].

```
#fitting the model in SVM
from sklearn.svm import SVC
classifi2 = SVC()
classifi2.fit(x_train,y_train)
y2_pred = classifi2.predict(x_test)
print(accuracy_score(y_test, y2_pred)*100)
```

```
87.17948717948718
```

Figure 3: *Output of the SVM Classifier*

2.4.3 XGBoost

XGBoost is a scalable gradient boosting technique that builds a new prediction model and if there any error in the previous model it corrects the error. Until there is no further improvement possible. Models are added serially. It will learn how to put weight on weak learners in order to develop a strong learner. It is quick on and known to take all the resources. Missing values can be handled properly, and models can be further boosted based on already fitted models, thus increasing accuracy, even further XGBoosting is a machine learning algorithm that is perfect for reducing the amount of computing resources used in the shortest amount of time. It was introduced by Tianqi Chen in 2014, and since then it has gained a lot of popularity among data scientists. XGBoosting is very simple to use and has advanced features that make it a versatile tool for many different platforms. In order to implement this algorithm, you will need to have access to the appropriate software and hardware. XG boosting requires a loss function to calculate, a weak learner to modify decision trees to minimize the error, and an additive model to calculate loss and untangle regression, classification, and ranking issues. The XGBoost algorithm supercharges gradient boosting duty. this open- source software library When applied produces billions of outcomes quickly. It's very fast, accurate, accessible, and scalable so it's no surprise that it has been adopted by many different companies.. The algorithm can quickly produce billions of results. Along with the tree learning algorithms, it can correct direct models also. This means it can manage multiple calculation on one machine. Result shown in figure 4 [2], [3].

```
# DataFlair - Calculate the accuracy
y_pred=model.predict(x_test)
print(accuracy_score(y_test, y_pred)*100)

94.87179487179486
```

Figure 4: *Output of the XGBoost Classifier*

3 Results and Discussion

In this project, detection of the presence of Parkinson's Disease in healthy and unhealthy individuals using various factors. Here, an XGB- Classifier, kNN and SVM for this and made use of the ski-kit learn library to prepare the dataset. XGB gives an accuracy of 94.87. Accuracy of 3 classifiers given in table 1.

Table 1: *Accuracy comparison*

Algorthim	**Accuracy**
Knn	89.74%
SVM	87.17%
XGB	94.82%

Table 2: *Output of the Confusion Matrix*

	Predicted healthy	Predicted parkinsons
True healthy	5	2
True parkinsons	0	32

4 Conclusion

Different types of classifiers are explored for their potential in the diagnosis of PD using speech signals. In this Python machine learning project, the detection of the presence of Parkinson's Disease in individuals using various factors. The comparison includes three traditional classifiers SVM, kNN and XGB. Here an XGBClassifier for this and made use of the sklearn library to prepare the dataset. This gives an accuracy of 94.87. Output of the Confusion Matrix shown in table 2.

5 Publisher's Note

AIJR remains neutral with regard to jurisdictional claims in institutional affiliations.

How to Cite

Thomas & Syama (2023). Comparison Study of Different Classifiers for Detecting Parkinson Disease using Machine Learning Language. *AIJR Proceedings*, 189-192. https://doi.org/10.21467/proceedings.160.23

References

[1] M. Ali et al., "Parkinson ' s Disease Read the Story of a Patient Diagnosed with Disease," no. 2267, p. 60008, 2020.

[2] Fahim, Nasif Wasek, Samik Ahmed Eshti, Khadiza Akter Nura, Md Jubayer Hossain Abir, and Mueem Nahid Ibn Mahbub (2020)" Parkinson Disease Detection: Using XGBoost Algorithm to Detect Early Onset Parkinson Disease", Research Gate, pp: 1-5

[3] J. Browniee, "A Gentle Introduction to XGBoost for Applied Machine Learning," Mach. Learn. Mastery, pp. 1–20, 2016, [Online].

[4] R. Betarbet, T. B. Sherer, and J. T. Greenamyre, "Animal models of Parkinson's disease. [Review] [89 refs]," Bioessays, 2002.

[5] J. A. M. Sidey-Gibbons and C. J. Sidey-Gibbons, "Machine learning in medicine: a practical introduction," BMC Med. Res. Methodol., vol. 19, no. 1, pp. 1–18, 2019, doi: 10.1186/s12874-019- 0681-4.

[6] S. Bind, A. K. Tiwari, and A. K. Sahani, "A Survey of Machine Learning Based Approaches for Parkinson Disease Prediction," Int. J. Comput. Sci. Inf. Technol., 2015.