Exploring Machine Learning Methods for Developing a Predictive System for Parkinson's Disease

Sumit Das[1]*, Tanusree Saha[1], Ira Nath[1] and Dipansu Mondal[2]

[1]JIS College of Engineering, Kalyani, India.
[2]University of Kalyani, Kalyani, India.

http://dx.doi.org/10.13005/bbra/3248

(Received: 04 April 2024; accepted: 24 June 2024)

The Integration of Machine Learning (ML) techniques holds significant promise in addressing challenges across various sectors, particularly within healthcare and biomedical fields. In this study, we focus on leveraging ML methodologies to address the longstanding issues surrounding the prediction and treatment of Parkinson's Disease (PD). PD prediction has historically suffered from inaccuracies and inconsistent treatments. Our research aims to mitigate these challenges by developing a predictive system tailored specifically to PD datasets. To achieve this, we systematically explore various ML algorithms for binary classification tasks, comparing their efficacy in predicting PD. By analyzing and comparing the performance of these algorithms, we aim to establish a robust pathway for accurately examining and diagnosing PD, thereby reducing discrepancies and associated risks. Our findings underscore the importance of employing ML techniques in developing effective decision support systems for PD prediction. By synthesizing results from multiple algorithms, our study not only contributes to filling existing research gaps but also provides actionable insights for the development of advanced medical applications. Overall, this research offers a comprehensive evaluation of ML approaches in the context of PD prediction, highlighting their potential to revolutionize diagnostic processes and improve patient outcomes. Our work not only enhances our understanding of PD but also underscores the transformative impact of ML in addressing complex medical challenges.

Keywords: Binary Classification; Healthcare; Machine-Learning; Predictive Modeling; Parkinson's-Disease.

Machine Learning (ML) has emerged as a prominent technological trend due to its modernity and intricate technical applications, permeating various sectors with its versatility and capabilities[1]. In the realm of artificial intelligence, ML stands as a pivotal sub-domain, adept at harnessing data from diverse sources and formats to derive actionable insights[2]. Despite its prowess in managing extensive datasets, challenges persist, particularly in the domain of healthcare, where accurate data classification remains a formidable hurdle[3]. Within healthcare, the application of ML algorithms holds significant promise, particularly in disease prediction. By uncovering intricate patterns within medical data, these algorithms demonstrate remarkable predictive capabilities, underscoring their potential to revolutionize healthcare practices[4]. The significance of ML

*Corresponding author E-mail: sumit.das@jiscollege.ac.in algorithms in healthcare is highlighted in recent literature surveys, which emphasize their role in enhancing disease prediction and diagnostic accuracy.

In this study, we aim to delve deeper into the landscape of ML algorithms in healthcare, with a specific focus on disease prediction, particularly in the context of Parkinson's disease. Parkinson's disease(PD), a neurodegenerative turmoil characterized by motor or non-motor signs, presents unique challenges in terms of accurate prediction and early detection. Despite advancements in medical technology, the identification of PD remains elusive, often plagued by miss-predictions and pre-diagnostic errors. To address these challenges, we propose a comprehensive analysis of ML methods for binary classification, with a keen emphasis on accuracy and performance evaluation. By systematically comparing and contrasting various ML algorithms, we seek to identify the most suitable approach for developing a predictive system tailored to PD detection. Our overarching goal is to establish a coherent pathway for accurately identifying individuals at risk of PD while mitigating the occurrence of miss-predictions and diagnostic inaccuracies. The significance of our research lies in its potential to contribute to the refinement of PD prediction methodologies, thereby enhancing early detection and intervention strategies. By leveraging the power of ML algorithms, we aspire to pave the way for more precise and reliable PD diagnostics, ultimately improving patient outcomes and quality of life.

This study endeavors to link the gaps between ML methodologies and healthcare applications, particularly in the realm of PD prediction. Through meticulous analysis and evaluation, we aim to advance the field by offering insights into the optimal utilization of ML algorithms for disease detection and management.

Literature Survey

Machine Learning (ML) has emerged as a prominent trend in various industries due to its modernity and intricate technical applications[5]. In the realm of healthcare, ML holds significant promise, particularly in the context of neurological disorders like Parkinson's disease (PD). PD is a progressive neurological ailment characterized by motor symptoms like tremors, stiffness, and slow movement[6]. The prevalence of PD has increased dramatically over the years, with a significant impact on global health[7]. Alzheimer's disease (AD) stands as the predominant manifestation of neurodegenerative dementia, now recognized as one of the most economically burdensome chronic illnesses. Automated diagnosis and management of Alzheimer's disease could significantly impact both society and patient welfare. Among the most prevalent symptoms of AD is language disorder, a direct consequence of cognitive decline[8]. The diagnosis of PD relies on the analysis of gait kinematics and other spatiotemporal characteristics[9]. However, due to the variability in symptoms and disease progression, PD is often misdiagnosed, leading to delays in treatment. Employing machine learning and voice analysis for diagnosing Parkinson's disease holds potential for offering non-invasive, cost-effective, and potentially more accessible diagnostic approaches. Nevertheless, additional research and validation are imperative to guarantee the reliability and accuracy of these methodologies in clinical settings[10].

To address this challenge, researchers have been exploring the application of ML algorithms to improve the accuracy and efficiency of PD detection. Through the integration of supervised learning methodologies with suitable feature selection techniques applied to voice datasets, researchers have the opportunity to create precise and comprehensible models for forecasting Parkinson's Disease. This endeavor holds the potential to facilitate early detection and intervention efforts[11]. Numerous scholarly inquiries have delved into the realm of speech impairment within the context of Parkinson's disease. These investigations have scrutinized a multitude of factors influencing speech challenges, assessed the efficacy of diverse therapeutic interventions, and evaluated the ramifications of speech impairment on various facets of patients' well-being[12].

Continuing research endeavors seek to deepen our comprehension of the fundamental mechanisms underlying speech and swallowing challenges in Parkinson's disease, with a specific focus on refining therapeutic strategies. This encompasses endeavors to unravel the intricate involvement of brain circuits in motor regulation and to explore innovative treatment modalities such as deep brain stimulation (DBS) and transcranial magnetic stimulation (TMS) aimed at enhancing speech and swallowing capabilities[13]. Recently, there has been a growing interest in utilizing ML and artificial intelligence (AI) approaches for disease prediction and diagnosis[14]. The emergence of deep learning (DL) has reshaped the landscape of artificial intelligence (AI) tools, igniting significant enthusiasm for AI applications in recent times[15]. In this study, we focus on the implementation of binary classification techniques, including logistic regression, SVM, kNN, Naive Bayes, decision trees, and Random Forest, to analyze the performance of PD detection algorithms[16]. The primary objective of XAI methodologies is to offer clarity and comprehensibility to AI models, enabling clinicians to grasp the reasoning behind their predictions. This is particularly critical in the medical field, where decisions carry substantial implications[17]. By adhering to these procedures and conducting thorough evaluations of the algorithms using pertinent datasets, one can proficiently gauge their precision and efficacy in forecasting Parkinson's Disease[18].

Furthermore, the advancement of ML, particularly deep learning (DL), has revolutionized medical diagnosis by enabling the analysis of large volumes of data with unprecedented accuracy[19]. Throughout the diagnostic journey, clinicians integrate various forms of information, including patient complaints, medical imaging, and laboratory findings. However, existing deep-learning models designed to assist in diagnosis have not fully addressed the need to incorporate multimodal information. The development of unified multimodal transformer-based models holds promise in enhancing patient triage and expediting clinical decision-making processes[20]. This technological advancement has empowered healthcare professionals to make quicker and more accurate prognoses, diagnoses, and treatment decisions. Overall, the integration of ML and AI techniques in medical diagnosis signifies a paradigm shift in healthcare, where these technologies complement the expertise of healthcare professionals, leading to improved patient outcomes and quality of care.

METHODOLOGY

The methodology employed in this study underscores a systematic approach toward leveraging ML methods for the prediction and detection of PD as depicted in Figure 1. The methodology encompasses several key stages, each tailored to facilitate the effective investigation and elucidation of data, ultimately culminating in the development of a predictive model.

(i) Importing Dependencies: The initial step involves importing necessary libraries and dependencies essential for the execution of the ML pipeline[21]. As the behavior and quality of an ML system are contingent upon the input features, careful consideration is given to the selection and incorporation of relevant libraries to ensure optimal performance.

(ii) Data Collection: The dataset used in this study is sourced from Kaggle, a renowned platform for data science competitions and datasets. Following the acquisition, the dataset is loaded into the pandas data frame, enabling comprehensive data analysis. Information regarding the dataset's dimensions, including the number of rows and columns, is extracted, facilitating a preliminary understanding of the data's structure and composition. Additionally, measures such as identifying missing values and statistical analyses are conducted to assess the dataset's integrity and completeness[22].

(iii) Data Preprocessing: Data preprocessing constitutes a critical phase wherein raw data is refined and organized into a cohesive format suitable for subsequent analysis[23]. This involves tasks such as data cleaning, normalization, and transformation to rectify inconsistencies and enhance the dataset's quality and coherence.

(iv) Exploratory Data Analysis (EDA): EDA serves as a fundamental exploratory tool aimed at uncovering patterns, trends, and anomalies within the dataset. By systematically scrutinizing the data, researchers gain valuable insights into underlying relationships and phenomena, thereby informing subsequent modeling and analysis strategies[24].

(v) Feature Engineering: Feature engineering entails the extraction and transformation of raw data into informative features conducive to model learning and interpretation[25]. Through this process, relevant attributes are identified and engineered to encapsulate essential characteristics pertinent to PD prediction. Features serve as pivotal inputs to the ML pipeline, enabling the algorithm to discern meaningful patterns and associations within the data.

(vi) Test-Train Split: The dataset is partitioned into distinct subsets, namely the training and validation sets, employing a train-test split methodology[26]. The training set is used for model training and parameter estimation, while the validation set facilitates model evaluation and performance assessment. This iterative process enables researchers to gauge the model's efficacy and generalizability, thereby refining and optimizing model performance.

(vii) Data Standardization: Data standardization, also referred to as normalization, involves rescaling the attributes to adhere to a standardized scale, typically with a mean of 0 and variance of 1. This normalization process ensures uniformity across feature distributions, thereby mitigating biases and disparities in value ranges. Standardization enhances the interpretability and stability of the ML model, facilitating more robust and reliable predictions[27]. The algorithms are illustrated briefly as follows:

K Nearest Neighbor

K-Nearest Neighbor (KNN) algorithm stands out as a fundamental yet potent tool in machine learning, valued for its simplicity, versatility, and non-parametric nature. While suitable for both classification and regression tasks, its predominant application lies in classification prediction. Functioning on the principle of proximity, KNN categorizes new data points by aligning them with previously trained instances, thereby organizing them into cohesive clusters or segments. This process hinges on the assumption of similarity between the new observation and the existing dataset, assigning the former to the classification that most closely resembles the latter. Prioritizing proximity, the algorithm arranges input data based on their likeness to neighboring instances, thereby determining their classification. Noteworthy is KNN's adeptness in handling large datasets while preserving classification accuracy and performance[28].

Support vector machine

The Support Vector Machine (SVM) model represents a powerful machine learning approach grounded in computational and statistical principles, focusing on the investigation of the VC dimension and the empirical risk minimization concept. This methodology offers distinct advantages in addressing challenges associated with pattern recognition tasks, particularly in scenarios with limited sample sizes, data heterogeneity, and computational intricacies. Notably, SVM effectively circumvents issues like the "curse of dimensionality" and mitigates risks of "over-learning," showcasing robustness against various complexities. Supported by a

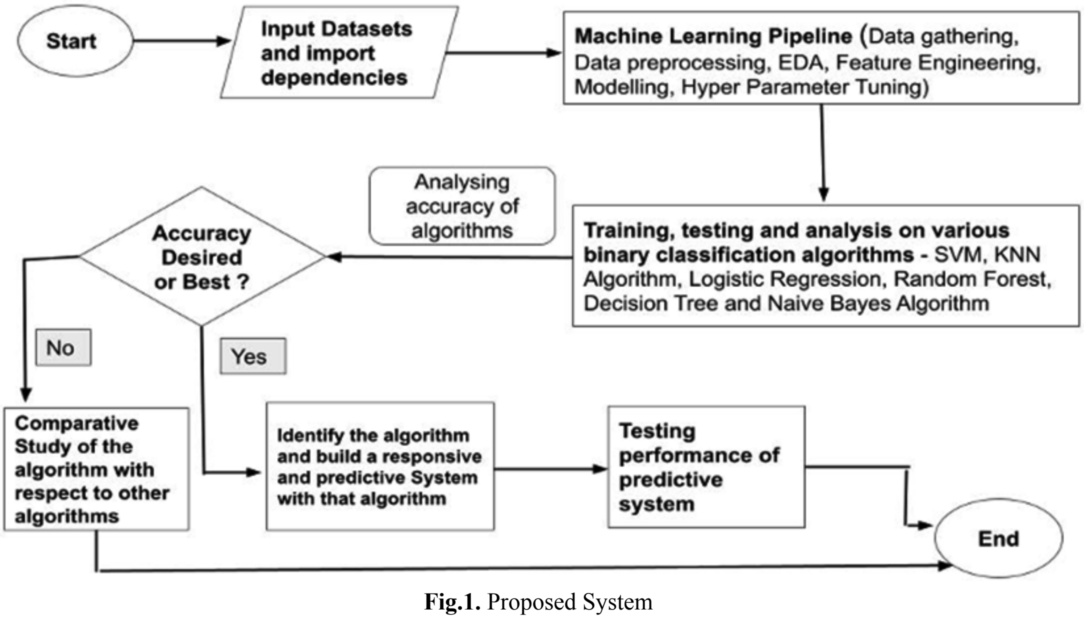

Start
Input Datasets and import dependencies
Machine Learning Pipeline (Data gathering, Data preprocessing, EDA, Feature Engineering, Modelling, Hyper Parameter Tuning)
Training, testing and analysis on various binary classification algorithms - SVM, KNN Algorithm, Logistic Regression, Random Forest, Decision Tree and Naive Bayes Algorithm
Analysing accuracy of algorithms
Accuracy Desired or Best ?
No
Yes
Comparative Study of the algorithm with respect to other algorithms
Identify the algorithm and build a responsive and predictive System with that algorithm
Testing performance of predictive system
End

Fig.1. Proposed System

solid theoretical foundation and a straightforward mathematical framework, SVM has significantly propelled advancements in pattern recognition, regression analysis, function estimation, time series forecasting, and related domains[29].

Logistic Regression

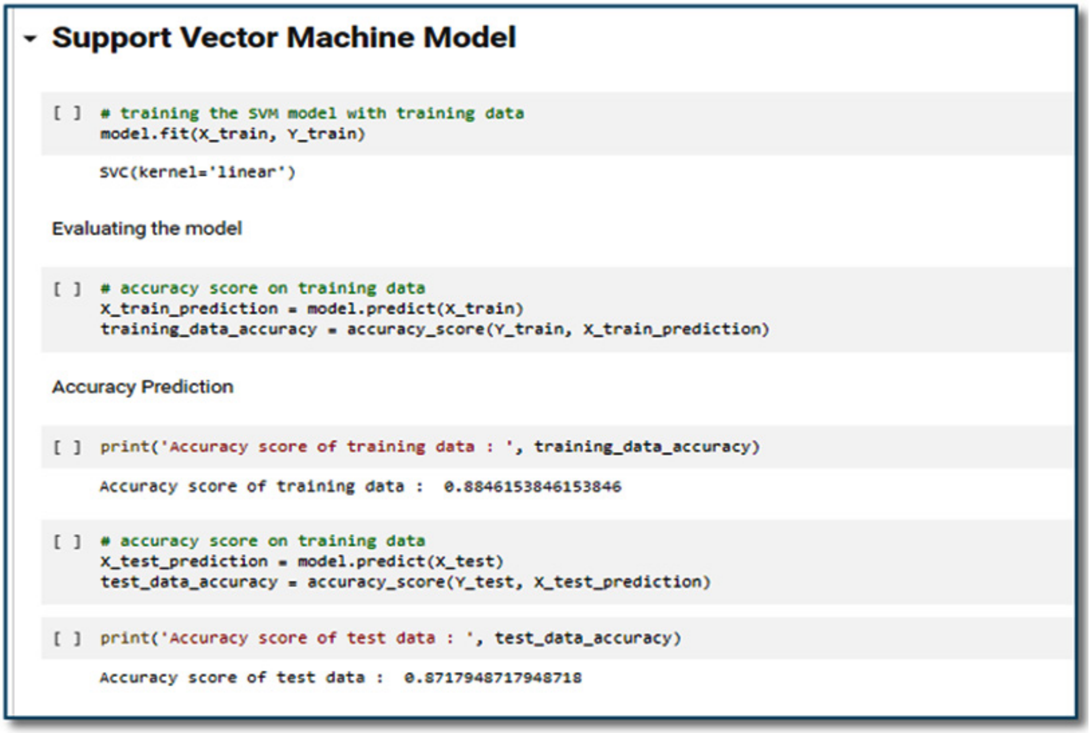
Logistic regression stands as a statistical technique employed for modeling the likelihood of discrete outcomes based on input variables. Particularly in classification tasks, it serves as a valuable analytical tool for determining the probability of a new sample belonging to a specific category. Given the classification nature of various aspects. Leveraging its capabilities, logistic regression contributes significantly to addressing classification challenges inherent in cyber security domains[30].

Naive Bayes

It is a supervised learning procedure, that employs the Bayes theorem to tackle classification, particularly prevalent in text categorization scenarios necessitating substantial training data. Serving as a foundational and efficient classification strategy, the Naive Bayes Classifier enables the development of rapid machine learning systems conducive to accurate predictions. As a probabilistic classifier, it operates by deriving predictions from the probabilities associated with respective objects[31].

Decision Tree

It is also a supervised learning method, and serves as a versatile tool capable of addressing both classification as well as regression, although it finds its primary application in classification scenarios. Within this tree-like structure, internal nodes represent attributes of the dataset, branches signify the decision process, and every leaf furnishes the final wrapping up. The pivotal components of a Decision Tree include Decision Nodes, pivotal in major decision-making with multiple branches, and Leaf Nodes, which represent the outcomes of specific decisions without further branching. Leveraging dataset characteristics, Decision Trees enable decision-making processes, experimentation, and testing. Functioning as a graphical representation, Decision Trees systematically explore various pathways to arrive at potential solutions for a given problem[32].



Support Vector Machine Model

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

     SVC(kernel='linear')
```

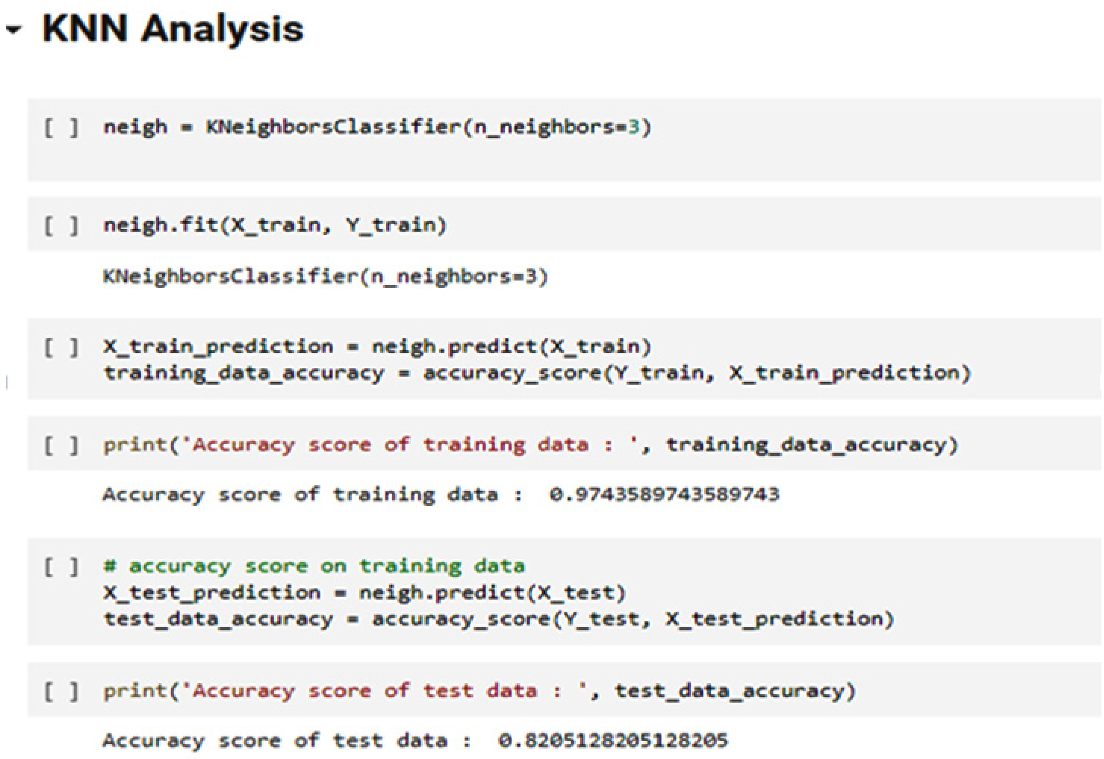
Evaluating the model

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

Accuracy 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
```

Fig. 2. Evaluation of accuracy and complexity for SVM



KNN Analysis

```
[ ]  neigh = KNeighborsClassifier(n_neighbors=3)
```

```
[ ]  neigh.fit(X_train, Y_train)

     KNeighborsClassifier(n_neighbors=3)
```

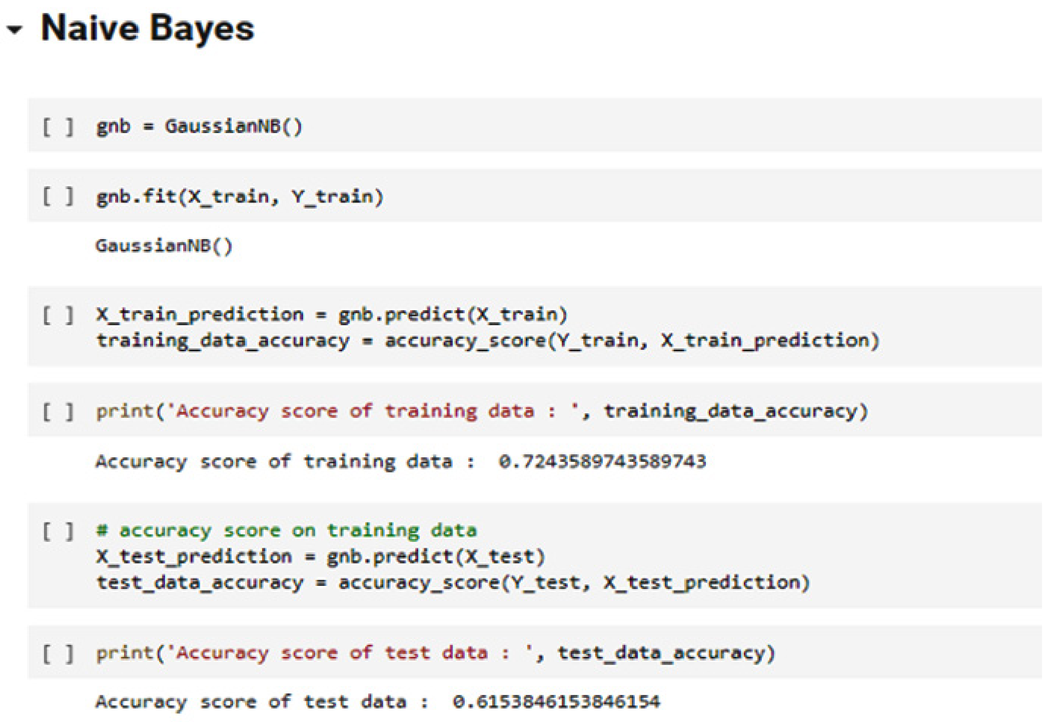
```
[ ]  X_train_prediction = neigh.predict(X_train)
     training_data_accuracy = accuracy_score(Y_train, X_train_prediction)
```

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

     Accuracy score of training data :  0.9743589743589743
```

```
[ ]  # accuracy score on training data
     X_test_prediction = neigh.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 :  0.8205128205128205
```

Fig. 3. Evaluation of KNN accuracy



Naive Bayes

```
[ ]  gnb = GaussianNB()
```

```
[ ]  gnb.fit(X_train, Y_train)

     GaussianNB()
```

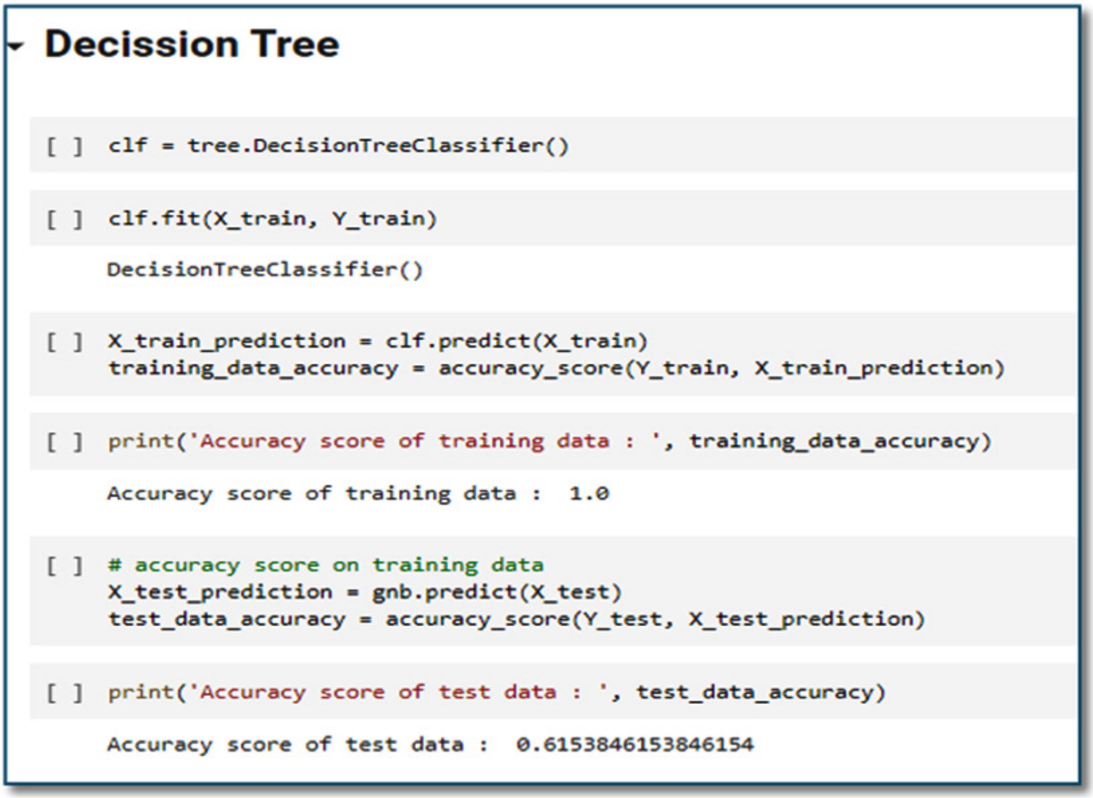
```
[ ]  X_train_prediction = gnb.predict(X_train)
     training_data_accuracy = accuracy_score(Y_train, X_train_prediction)
```

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

     Accuracy score of training data :  0.7243589743589743
```

```
[ ]  # accuracy score on training data
     X_test_prediction = gnb.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 :  0.6153846153846154
```

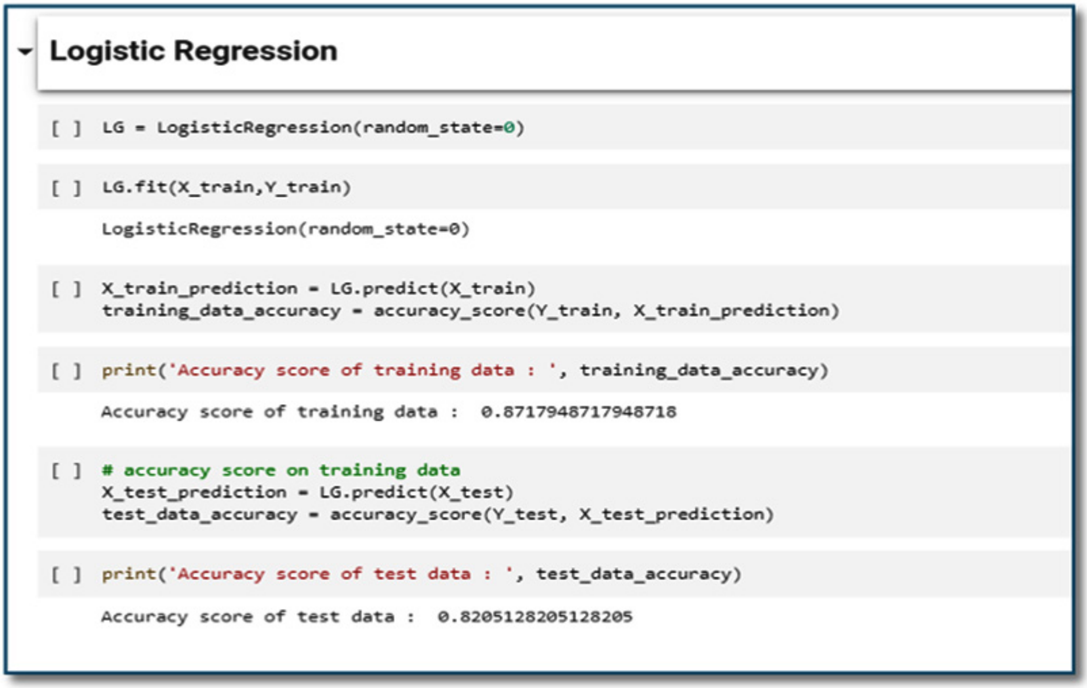
Fig. 4. Evaluation of accuracy and complexity for the Naive Bayes Algorithm



Decission Tree

```
[ ]  clf = tree.DecisionTreeClassifier()

[ ]  clf.fit(X_train, Y_train)

     DecisionTreeClassifier()

[ ]  X_train_prediction = clf.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 :  1.0

[ ]  # accuracy score on training data
     X_test_prediction = gnb.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 :  0.6153846153846154
```

Fig. 5. Evaluation of accuracy and complexity for the Decision Tree Algorithm



Logistic Regression

```
[ ]  LG = LogisticRegression(random_state=0)

[ ]  LG.fit(X_train,Y_train)

     LogisticRegression(random_state=0)

[ ]  X_train_prediction = LG.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 :  0.8717948717948718

[ ]  # accuracy score on training data
     X_test_prediction = LG.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 :  0.8205128205128205
```

Fig. 6. Evaluation of Logistic Regression Accuracy

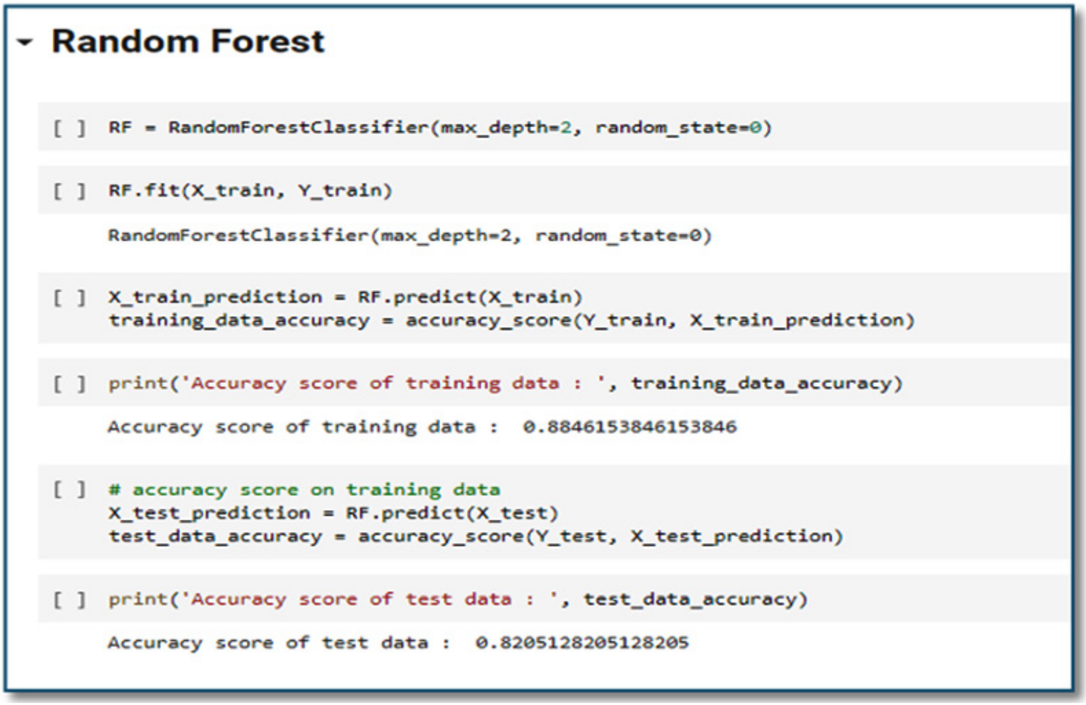
Random Forest

It stands as a prominent ML method within the domain of supervised learning, renowned for its versatility in addressing both regression and classification tasks. Embracing the concept of ensemble learning, Random Forest amalgamates multiple classifiers to tackle intricate problems, thereby enhancing model performance. This approach leverages the collective wisdom of diverse classifiers to navigate complexities inherent in data-driven challenges. The algorithm's efficacy in handling a spectrum of machine learning problems is underscored by its reliance on ensemble-based methodologies[33].

In each model, a dataset comprising labeled samples is utilized to explore the statistical relationship between attributes and objectives. The model demonstrating superior performance is subsequently evaluated against a distinct dataset, not employed during the training phase. This assessment ensures the creation and upkeep of a predictive model capable of generalizing well to new data. Drawing upon the methodologies discussed above, this summary encapsulates the scientific pursuit of model development and validation.

Table 1. Accuracy of training and testing across various algorithms

Algorithm Name	Training Data Accuracy	Testing Data Accuracy
SVM (Support Vector Machine)	88.46%	87.17%
KNN Algorithm	97.43%	87.05%
Naive Bayes Algorithm	72.43%	61.53%
Decision Tree Algorithm	100%	61.53%
Logistic Regression Algorithm	87.17%	82.15%
Ramdom Forest Algorithm	88.46%	82.05%



▾ Random Forest

```
[ ] RF = RandomForestClassifier(max_depth=2, random_state=0)
```

```
[ ] RF.fit(X_train, Y_train)

    RandomForestClassifier(max_depth=2, random_state=0)
```

```
[ ] X_train_prediction = RF.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 = RF.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 :  0.8205128205128205
```

Fig. 7. Evaluation of Random Forest accuracy

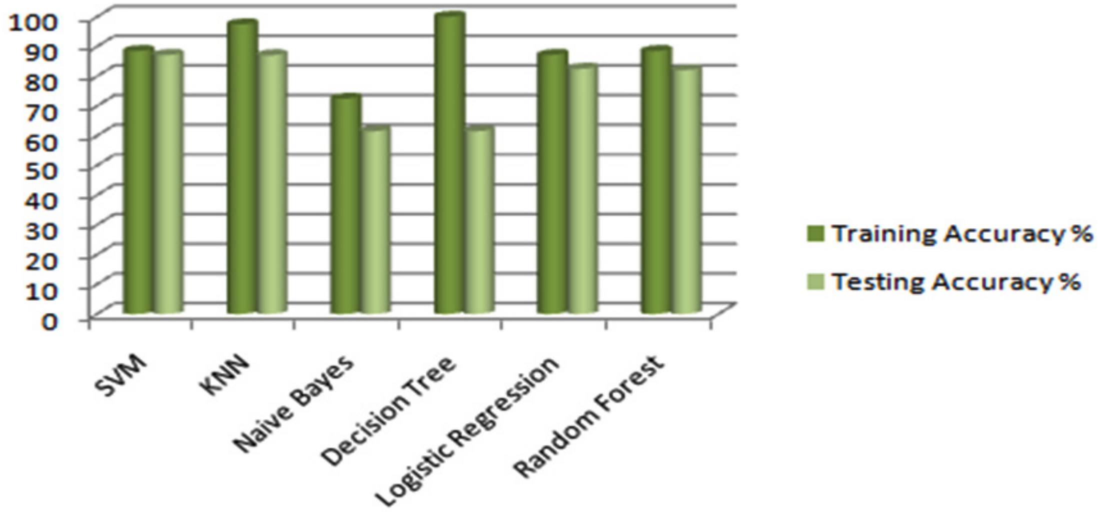


Fig. 8. Comparison of accuracy among various algorithms

RESULTS AND DISCUSSION

This study contributes to the advancement of knowledge by conducting a comparative analysis of various binary classification algorithms. Through rigorous experimentation, distinct levels of accuracy are observed across different algorithms, emphasizing the significance of accuracy as a performance metric in Machine Learning. The machine learning pipeline begins with the importation of several libraries, followed by iterative stages of characteristic engineering techniques, including data homogeny and mock-up training. Utilizing algorithms from the sklearn, such as SVM, KNN, Naive Bayes, and decision trees, the predictive model is developed to discern the presence of Parkinson's disease. Comparative examination and complexity analysis reveals that SVM methods exhibit the highest accuracy on testing datasets, indicating their efficacy in predictive modeling. Consequently, a robust predictive model is established, paving the way for enhanced diagnosis and management of Parkinson's disease.

Support Vector Machine(SVM)

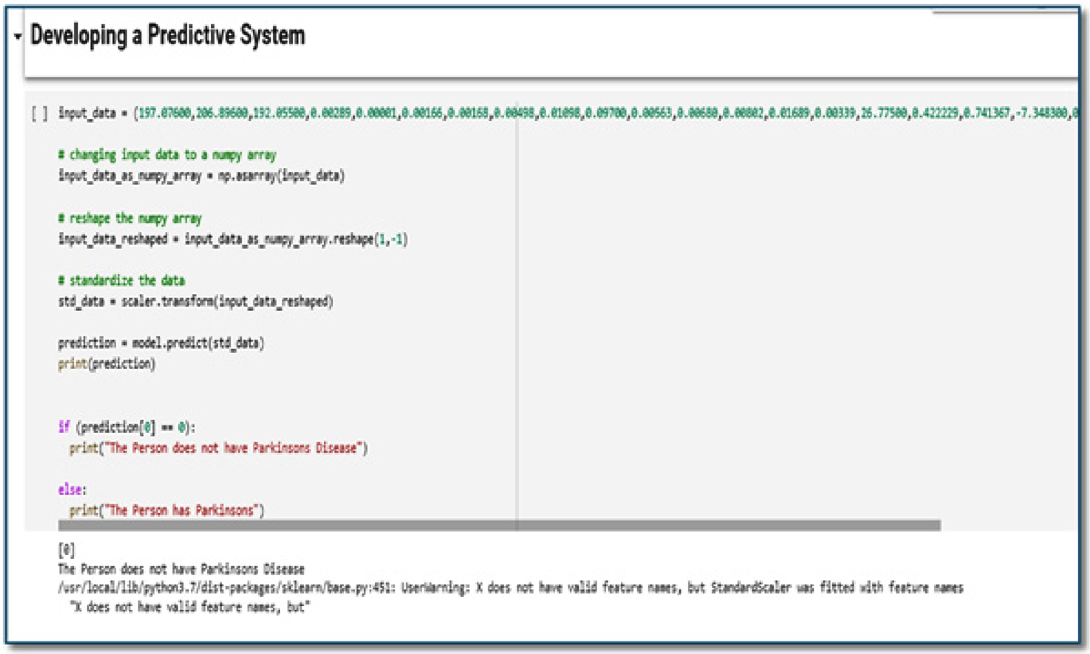
The Support Vector Machine (SVM) model demonstrates promising performance, achieving an accuracy of 88.46% on the training dataset and 87.17% on the testing dataset. These results underscore the efficacy of SVM in accurately classifying data points. Figure 2 visually depicts the accuracy and complexity analysis of SVM, providing insights into its performance characteristics. This notable accuracy on both training and testing datasets suggests that SVM holds potential as a reliable predictive model. Further discussion and analysis are warranted to elucidate the factors contributing to SVM's success and to explore potential avenues for refinement and optimization.

K-Nearest Neighbor

The K-Nearest Neighbor (KNN) algorithm demonstrates commendable performance, achieving an accuracy of 97.43% on the training dataset and 87.05% on the testing dataset. These findings highlight the robustness of KNN in accurately classifying data points. Figure 3 presents the accuracy and complexity of KNN, providing valuable insights into its performance characteristics. Notably, the high accuracy on the training dataset indicates the model's ability to effectively capture underlying patterns in the data. However, the slightly lower accuracy on the testing dataset suggests the need for further investigation to address potential over-fitting or generalization issues. Future research endeavors may focus on optimizing the parameters of the KNN algorithm to enhance its performance in real-world applications.

Naive Bayes

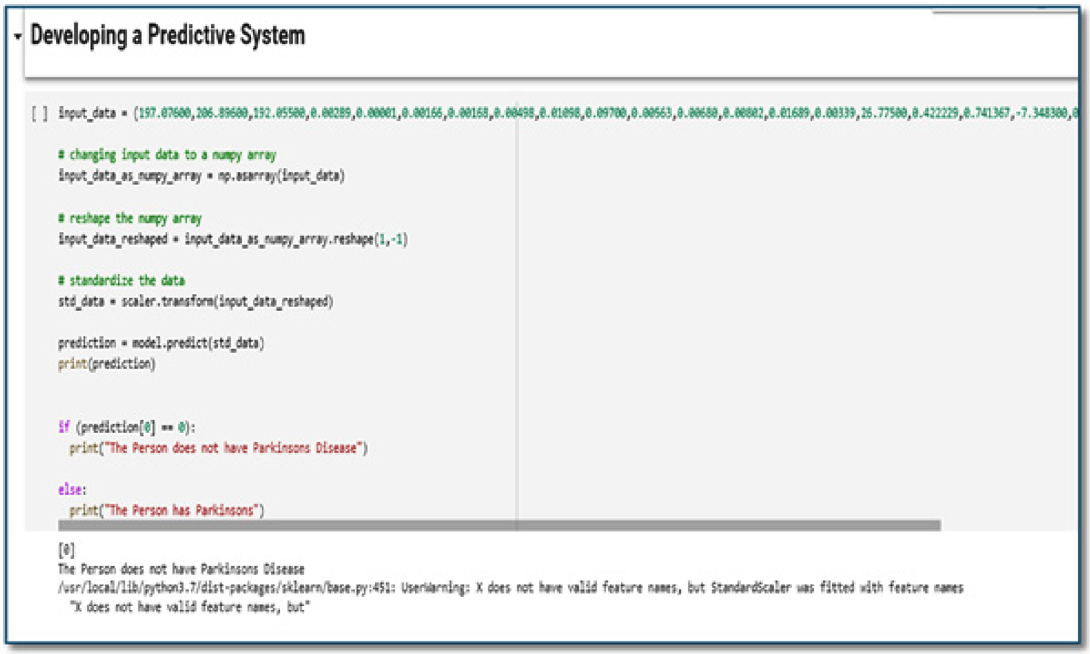
The Naive Bayes Algorithm demonstrates moderate performance, achieving an accuracy of 72.43% on the training dataset and 61.53% on the testing dataset. These results indicate its capability to classify data points with reasonable accuracy. Figure 4 illustrates the accuracy of the Naive Bayes, providing insights into its performance characteristics. While the algorithm performs adequately on the training dataset, the lower accuracy on the testing dataset suggests potential challenges in generalization. Further investigation is warranted to identify factors contributing to this discrepancy and to explore strategies for improving the algorithm's performance. Future research efforts may focus on refining the model's



Developing a Predictive System

```
[ ] input_data = (197.07600,206.89600,192.05500,0.00289,0.00001,0.00166,0.00168,0.00498,0.01098,0.09700,0.00563,0.00680,0.00802,0.01689,0.00339,26.77500,0.422229,0.741367,-7.348300,0

    # 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")
```

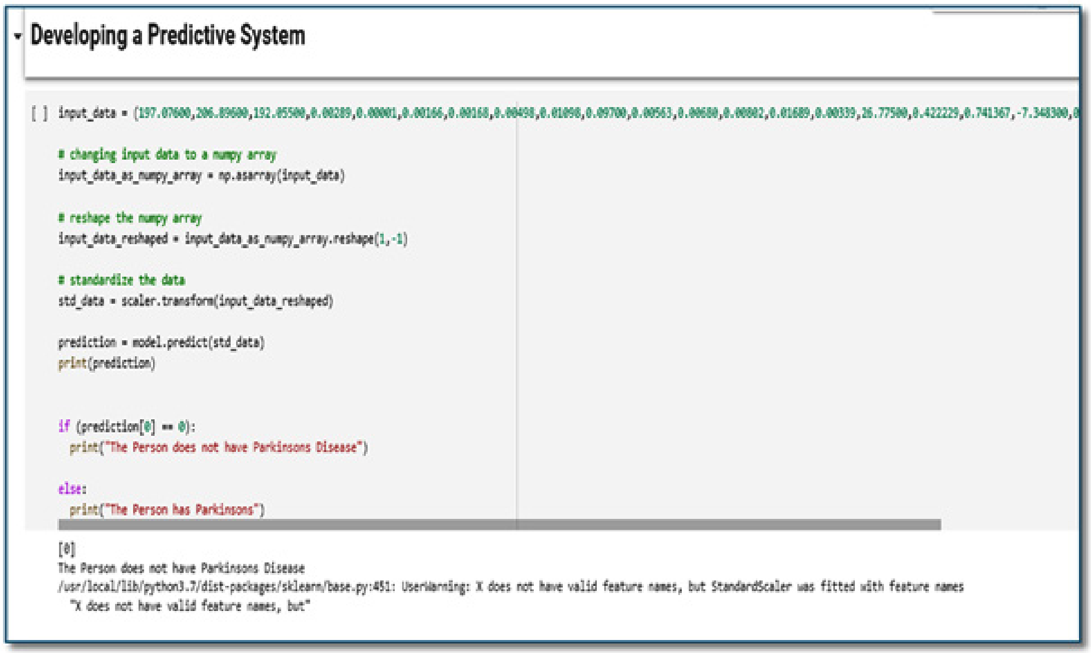
```
[0]
The Person does not have Parkinsons Disease
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature names
  "X does not have valid feature names, but"
```



Developing a Predictive System

```
[ ] input_data = (197.07600,206.89600,192.05500,0.00289,0.00001,0.00166,0.00168,0.00498,0.01098,0.09700,0.00563,0.00680,0.00802,0.01689,0.00339,26.77500,0.422229,0.741367,-7.348300,0

    # 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
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature names
  "X does not have valid feature names, but"
```



Developing a Predictive System

```
[ ] input_data = (197.07600,206.89600,192.05500,0.00289,0.00001,0.00166,0.00168,0.00498,0.01098,0.09700,0.00563,0.00680,0.00802,0.01689,0.00339,26.77500,0.422229,0.741367,-7.348300,0

    # 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
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature names
  "X does not have valid feature names, but"
```

Fig. 9. Result of Predictive System

parameters or exploring alternative approaches to enhance its predictive capabilities in diverse contexts.

Decision Tree Algorithm

The Decision Tree Algorithm exhibits a remarkable accuracy of 100% on the training dataset, indicating its adeptness at capturing intricate patterns within the data. However, its performance on the testing dataset is comparatively lower, with an accuracy of 61.53%. Figure 5 visually represents the accuracy of the Decision Tree, shedding light on its performance metrics. While the algorithm achieves perfect accuracy on the training dataset, its lower accuracy on the testing dataset suggests potential over-fitting or lack of generalization. Further investigation is necessary to identify the underlying causes and to explore strategies for improving the algorithm's performance on unseen data. Future research endeavors may focus on optimizing the Decision Tree Algorithm parameters or employing ensemble techniques to enhance its predictive capabilities across diverse datasets.

Logistic Regression Algorithm

The Logistic Regression Algorithm demonstrates favorable performance, achieving an accuracy of 87.17% on the training dataset and 82.15% on the testing dataset. These results suggest its efficacy in accurately classifying data points. Figure 6 visually depicts the accuracy of the Logistic Regression, providing insights into its performance characteristics. While the algorithm exhibits strong performance on both training and testing datasets, the slightly lower accuracy on the testing dataset indicates the need for further evaluation. Potential areas for improvement may include fine-tuning model parameters or exploring feature engineering techniques to enhance predictive accuracy. Continued research efforts are warranted to optimize the Logistic Regression Algorithm for diverse applications and datasets.

Random Forest Algorithm

The Random Forest Algorithm demonstrates strong performance, achieving an accuracy of 88.46% on the training dataset and 82.05% on the testing dataset. These results underscore its effectiveness in accurately classifying data points. Figure 7 provides a visual representation of the accuracy of the Random Forest, offering valuable insights into its performance characteristics. While the algorithm performs well on both training and testing datasets, there is a slight drop in accuracy on the testing dataset compared to the training dataset. This discrepancy suggests the need for further investigation into potential over-fitting or generalization issues. Future research may focus on optimizing the Random Forest Algorithm parameters or exploring ensemble techniques to enhance its predictive capabilities across diverse datasets.

Through these relative explorations and complexity investigation across multiple classification methods, it is evident that the SVM demonstrates superior precision on the test data, as illustrated in Table 1 and Figure 8.

As a result of our research efforts, a predictive model capable of discerning the presence of Parkinson's disease in patients has been devised. It was revealed through our analysis that among all algorithms investigated, the highest performance was exhibited by the SVM, followed closely by the KNN method. The significance of future endeavors focused on smart systems or developments about Parkinson's syndrome is underscored by these findings. This observation is supported by Figure 9, which visually represents the comparative performance of the algorithms.

Hence, we have constructed an intelligent predictive model aimed at discerning the presence of Parkinson's disease in patients. Among all algorithms examined in this study, the SVM algorithm exhibited the highest performance, with the KNN Algorithm following closely. These findings underscore the significance of further research in the realm of intelligent systems and Parkinson's disease-related developments, highlighting promising avenues for future endeavors.

CONCLUSION

In this study, we conducted a thorough analysis of algorithmic complexity and performed a comparative examination of various algorithms, laying the groundwork for the development of a robust predictive machine learning framework aimed at addressing discrepancies in Parkinson's disease prediction and enhancing overall prediction accuracy. While classification techniques have been extensively studied in the past, our exploration of different methodologies has revealed diverse outcomes depending on dataset characteristics, algorithm tuning, and enhancement strategies.

The primary strength of this work lies in the establishment of a structured framework for the development of the predictive system, derived from comprehensive algorithm analysis and comparison. This framework holds the potential to mitigate barriers in diagnosis and prediction, thereby facilitating improved healthcare outcomes. However, a notable weakness is the system's reliance on datasets, as manipulation or inaccuracies in data may lead to erroneous predictions. Opportunities for future research lie in exploring multi-directional approaches to algorithm selection and development, with the potential for creating specialized systems beyond prediction tasks.

Despite these strengths and opportunities, the primary threat to our developed system stems from advancements in dataset quality and algorithmic improvements, which may render other algorithms more predictive and accurate, thereby diminishing the efficacy of our system. Continuous monitoring and adaptation will be essential to ensure the relevance and effectiveness of our predictive framework in an evolving landscape of machine learning research and application.

ACKNOWLEDGMENTS

The authors extend their gratitude to the Management of JIS College of Engineering, JIS GROUP, for their generous provision of essential resources and facilities throughout this endeavor.

Conflict of interest

Authors declare that they have no conflict of interest.

Funding Sources

There is no funding sources.

Ethics Approval Statement

This article does not contain any studies with human participants or animals performed by any of the authors.

Authors' Contribution

Informed consent was obtained from all individual participants included in the study and each author has equal contribution.

Data availability

The datasets analyzed during the current study are available in the Kaggle repository and the web link is as follows: https://www.kaggle.com/

datasets/vikasukani/parkinsons-disease-data-set This dataset is cited in reference number 22.

REFERENCES

1. Habehh H, Gohel S. Machine Learning in Healthcare. *Curr Genomics*. 2021;22(4):291-300. doi:10.2174/138920292266621070512435 9
2. Woodman RJ, Mangoni AA. A comprehensive review of machine learning algorithms and their application in geriatric medicine: present and future. *Aging Clin Exp Res*. 2023;35(11):2363-2397. doi:10.1007/s40520-023-02552-2
3. Pudjihartono N, Fadason T, Kempa-Liehr AW, O'Sullivan JM. A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction. *Front Bioinforma*. 2022;2. doi:10.3389/fbinf.2022.927312
4. Salhout SM. Machine learning in healthcare strategic management: a systematic literature review. *Arab Gulf J Sci Res*. 2023;ahead-of-print(ahead-of-print). doi:10.1108/AGJSR-06-2023-0252
5. Kumar Y, Koul A, Singla R, Ijaz MF. Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda. *J Ambient Intell Humaniz Comput*. Published online January 13, 2022. doi:10.1007/s12652-021-03612-z
6. Mei J, Desrosiers C, Frasnelli J. Machine Learning for the Diagnosis of Parkinson's Disease: A Review of Literature. *Front Aging Neurosci*. 2021;13. doi:10.3389/fnagi.2021.633752
7. Henrich MT, Oertel WH, Surmeier DJ, Geibl FF. Mitochondrial dysfunction in Parkinson's disease – a key disease hallmark with therapeutic potential. *Mol Neurodegener*. 2023;18(1):83. doi:10.1186/s13024-023-00676-7
8. Ammar RB, Ayed YB. Language-related features for early detection of Alzheimer Disease. *Procedia Comput Sci*. 2020;176:763-770. doi:10.1016/j.procs.2020.09.071
9. Alshammri R, Alharbi G, Alharbi E, Almubark I. Machine learning approaches to identify Parkinson's disease using voice signal features. *Front Artif Intell*. 2023;6:1084001. doi:10.3389/frai.2023.1084001
10. Iyer A, Kemp A, Rahmatallah Y, et al. A machine learning method to process voice samples for identification of Parkinson's disease. *Sci Rep*. 2023;13(1):20615. doi:10.1038/s41598-023-47568-w
11. Ali AM, Salim F, Saeed F. Parkinson's Disease Detection Using Filter Feature Selection and a Genetic Algorithm with Ensemble Learning. *Diagnostics*. 2023;13(17):2816. doi:10.3390/diagnostics13172816
12. Wiesman AI, Donhauser PW, Degroot C, et al. Aberrant neurophysiological signaling associated with speech impairments in Parkinson's disease. *Npj Park Dis*. 2023;9(1):1-13. doi:10.1038/s41531-023-00495-z
13. Do N, Mitchell S, Sturgill L, Khemani P, Sin MK. Speech and Swallowing Problems in Parkinson's Disease. *J Nurse Pract*. 2022;18(8):848-851. doi:10.1016/j.nurpra.2022.05.019
14. Speech & Swallowing in Parkinson's | Parkinson's Foundation. Published 2024. Accessed March 10, 2024. https://www.parkinson.org/library/fact-sheets/speech-swallowing
15. Bohr A, Memarzadeh K. The rise of artificial intelligence in healthcare applications. *Artif Intell Healthc*. Published online 2020:25-60. doi:10.1016/B978-0-12-818438-7.00002-2
16. Albuquerque J, Medeiros AM, Alves AC, Bourbon M, Antunes M. Comparative study on the performance of different classification algorithms, combined with pre- and post-processing techniques to handle imbalanced data, in the diagnosis of adult patients with familial hypercholesterolemia. *PLoS ONE*. 2022;17(6):e0269713. doi:10.1371/journal.pone.0269713
17. What is Explainable AI (XAI)? | IBM. Published 2024. Accessed March 10, 2024. https://www.ibm.com/topics/explainable-ai
18. Dixit S, Bohre K, Singh Y, et al. A Comprehensive Review on AI-Enabled Models for Parkinson's Disease Diagnosis. *Electronics*. 2023;12(4):783. doi:10.3390/electronics12040783
19. Das S, Sanyal MK. Machine intelligent diagnostic system (MIDs): an instance of medical diagnosis of tuberculosis. *Neural Comput Appl*. 2020;32(19):15585-15595. doi:10.1007/s00521-020-04894-8
20. A transformer-based representation-learning model with unified processing of multimodal input for clinical diagnostics | Nature Biomedical Engineering. Accessed March 10, 2024. https://www.nature.com/articles/s41551-023-01045-x
21. Data Dependencies | Machine Learning. Google for Developers. Published 2024. Accessed March 10, 2024. https://developers.google.com/machine-learning/crash-course/data-dependencies/video-lecture
22. Data collection and pre-processing techniques - Qualcomm Developer Network. Published 2024. Accessed March 10, 2024. https://developer.qualcomm.com/software/qualcomm-neural-processing-sdk/learning-resources/

ai-ml-android-neural-processing/data-collection-pre-processing

23. ML | Data Preprocessing in Python - GeeksforGeeks. Published 2024. Accessed March 10, 2024. https://www.geeksforgeeks.org/data-preprocessing-machine-learning-python/
24. What is Exploratory Data Analysis? | IBM. Published 2024. Accessed March 10, 2024. https://www.ibm.com/topics/exploratory-data-analysis
25. What is Feature Engineering? | Domino Data Science Dictionary. Published 2024. Accessed March 10, 2024. https://domino.ai/data-science-dictionary/feature-engineering
26. Training and Test Sets: Splitting Data | Machine Learning. Google for Developers. Published 2024. Accessed March 10, 2024. https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data
27. Toward a "Standard Model" of Machine Learning · Issue 4.4, Fall 2022. Published 2024. Accessed March 10, 2024. https://hdsr.mitpress.mit.edu/pub/zkib7xth/release/2
28. Jain V. Introduction to KNN Algorithms. Analytics Vidhya. Published January 31, 2022. Accessed March 10, 2024. https://www.analyticsvidhya.com/blog/2022/01/introduction-to-knn-algorithms/
29. Support Vector Machine - an overview | ScienceDirect Topics. Published 2024. Accessed March 10, 2024. https://www.sciencedirect.com/topics/computer-science/support-vector-machine
30. Pant A. Introduction to Logistic Regression. Medium. Published January 22, 2019. Accessed March 10, 2024. https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148
31. What Are Naïve Bayes Classifiers? | IBM. Published 2024. Accessed March 10, 2024. https://www.ibm.com/topics/naive-bayes
32. Decision tree learning. In: *Wikipedia*. ; 2024. Accessed March 10, 2024. https://en.wikipedia.org/w/index.php?title=Decision_tree_learning&oldid=1211641048
33. Zhou S. *Random Forests and Regularization*. phd. University of Pittsburgh; 2022.