1. **INTRODUCTION**

Parkinson's disease symptoms can be different for everyone. Early signs are mild that goes unnoticed. Symptoms usually begin on one side of your body and gets worsen on that side, afterwards it affects both the sides. Parkinson's symptoms may include

• Tremor

• Slowed movement

• Rigid muscles.

• Impaired posture and balance.

• Loss of automatic movements

• Speech changes

• Writing

The Parkinson's disease is due to a loss of neurons that produce a chemical messenger in the brain called dopamine. when there is a decrease in level of the amino acid named dopamine it leads to the abnormal brain activity, which leads to Parkinson’s disease. The cause of Parkinson's disease is still a question mark, but several factors appear to play a role, including:

• Genes

• Environmental

• Triggers

As a result people suffer from this disease for many years before diagnosis. The estimated results have shown that there are 7-10 million people are affected by parkinson’s disease worldwide. People with age above 50 are the one’s who has the higher possibility of getting parkinson’s disease but still an estimated 4 percentage of people who are under the age 50 are diagnosed with parkinson’s disease. There is no cure or prevention for PD. However, the disease can be controlled in early stage. The data mining techniques is used as a effective way for early detection and diagnosis of the disease. Data mining techniques in medicine is a research area that combines sophisticated representational and computing techniques with the insights of expert physicians to produce tools for improving healthcare.Data mining is a statistical method for finding hidden patterns in datasets by constructing predictive or cla ssification models that can be learned from past experience and applied in future cases, so there is a need for a more accurate, objective means of early detection, ideally one which can be used by individuals in their home setting.

**1.1Project Overview**

Parkinson's disease is a neurological movement disorder. Common symptoms include tremor, slowness of movement, stiff muscles, unsteady walk and balance and coordination problems. Biomarkers derived from human voice can offer insight into neurological disorders, such as Parkinson’s disease (PD). PD is a progressive neurodegenerative disorder. Due to the decrease in motor control that is the hallmark of the disease, voice can be used to detect and diagnose PD. We provide evidence to validate this concept here using a voice dataset collected from people with and without PD

**1.2Purpose**

Parkinson's disease affects the CNS of the brain and has yet no treatment unless it's detected early. Late detection leads to no treatment and loss of life. Thus its early detection is significant. **For early detection of the disease**, we utilized machine learning algorithms.

1. **LITERATURE SURVEY**

**2.1 Existing problem**

In existing system, PD is detected at the secondary stage only (Dopamine deficiency) which leads to medical challenges. Also doctor has to manually examine and suggest medical diagnosis in which the symptoms might vary from person to person so suggesting medicine is also a challenge. Thus the mental disorders are been poorly characterized and have many health complications. PD is generally diagnosed with the following clinical methods as,

• MRI or CT scan - Conventional MRI cannot detect early signs of Parkinson's disease

• PET scan - is used to assess activity and function of brain regions involved in movement

• SPECT scan - can reveal changes in brain chemistry, such as a decrease in dopamine

This results in a high misdiagnosis rate (up to 25% by non-specialists) and many years before diagnosis, people can have the disease. Thus existing system is not effective in early prediction and accurate medicinal diagnosis to the affected people

* 1. **References**

[1] Adrien Payan, Giovanni Montana, Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks.

[2] Alemami, Y. and Almazaydeh, L. (2014) Detecting of Parkinson Disease through Voice Signal Features. Journal of American Science

[3] Fayao Liu, Chunhua Shen, Learning Deep Convolutional Features for MRI Based Alzheimer’s Disease Classification.

[4] Hadjahamadi, A.H. and Askari, T.J. (2012) A Detection Support System for Parkinson’s Disease Diagnosis Using Classification and Regression Tree. Journal of Mathematics and Computer Science , 4, 257-263.

[5] Little, M.A., McSharry, P.E., Hunter, E.J. and Ramig, L.O. (2008), Suitability of Dysphonia Measurements for Telemonitoring of Parkinson’s disease. IEEE Transactions on Biomed ical Engineering, 56, 1015-1022.

[6] Muhlenbach, F. and Rakotomalala, R. (2015) Discretization of Continuous Attributes. In: Wang, J., Ed., Encyclopedia of Data Warehousing and Mining, Idea Group Reference, 397-402.

[7] Olanrewaju, R.F., Sahari, N.S., Musa, A.A. and Hakiem, N. (2014) Application of Neural Networks in Early Detection and Diagnosis of Parkinson’s Disease. International Conference on Cyber and IT Service Management.

[8] Saman Sarraf, Danielle D. DeSouza, John Anderson, Ghassem Tofighi, DeepAD: Alzheimer′s Disease Classification via Deep Convolutional Neural Networks using MRI and fMRI, Cold Spring Harbor Laboratory Press.

* 1. **Problem Statement Definition**

Parkinson's disease is a neurological movement disorder. Common symptoms include tremor, slowness of movement, stiff muscles, unsteady walk and balance and coordination problems. Biomarkers derived from human voice can offer insight into neurological disorders, such as Parkinson’s disease (PD). PD is a progressive neurodegenerative disorder. Due to the decrease in motor control that is the hallmark of the disease, voice can be used to detect and diagnose PD. We provide evidence to validate this concept here using a voice dataset collected from people with and without PD.

1. **IDEATION & PROPOSED SOLUTION**

**3.1 Empathy Map Canvas**



**3.2 Ideation & Brainstorming**

**Brainstorm Top 3 Ideas**

• Idea 1: It provides a feature that allows users to place their values and interact with the friendly user assistance bot The chatbot that can be built in the application allows users to enter their respective inputs and allowable in interacting with the chatbot for having a positive conversation. User can expect their output at that instant after the time the input is feed into the model.

• Idea 2: It provides good classification ability in determining of whether the image is true or fake the prediction must be extremely true and as expected by the user. User might be beginner in using the application and they may not be aware of true and fake results that some other application may show in front of them. Using this classification techniques from machine learning, the application can classify the behaviour as true and fake based on the input data that is feed into the model.

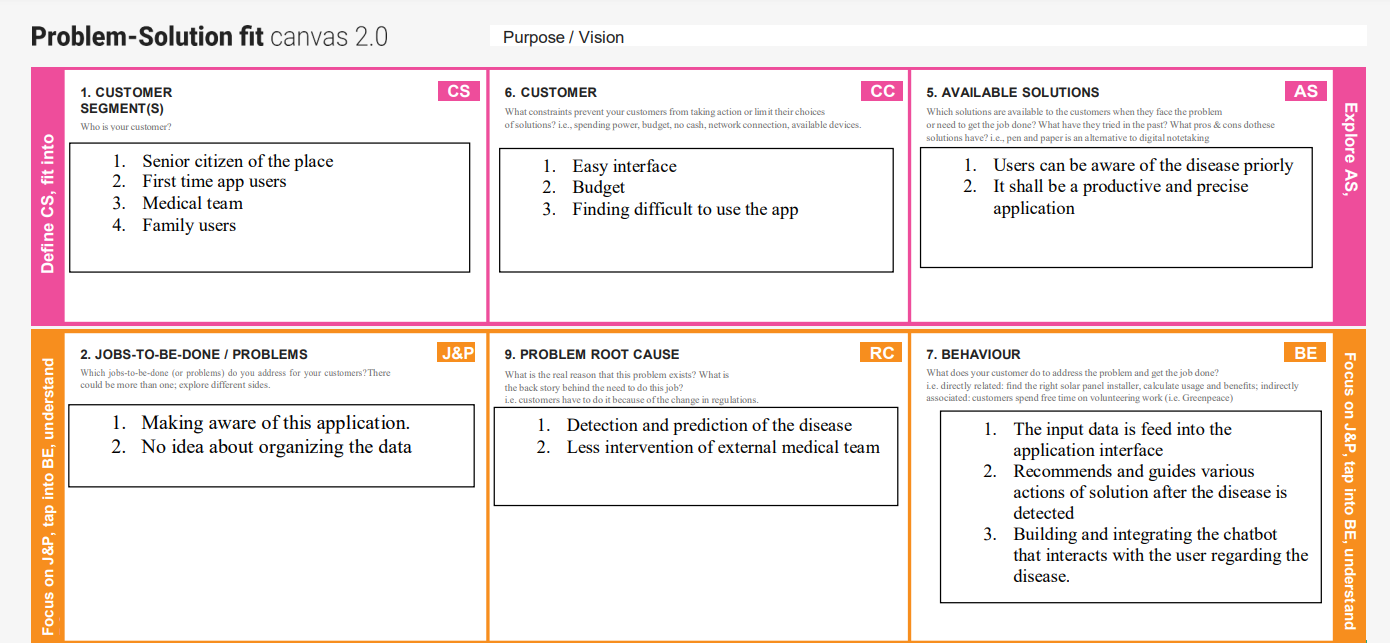
• Idea 3: It offers flexible and comfortable UI experience for first time users and mentally challenged people Usually; users visit the website and there is chance that they just leave or close the application due to some complicated features that is built in it. For to resolve such issues, the application’s UI/UX design must be considered majorly of how it can be used easily by any users. So, our application can provide better UI experience for the users to check up their status of the disease impacting on their time.

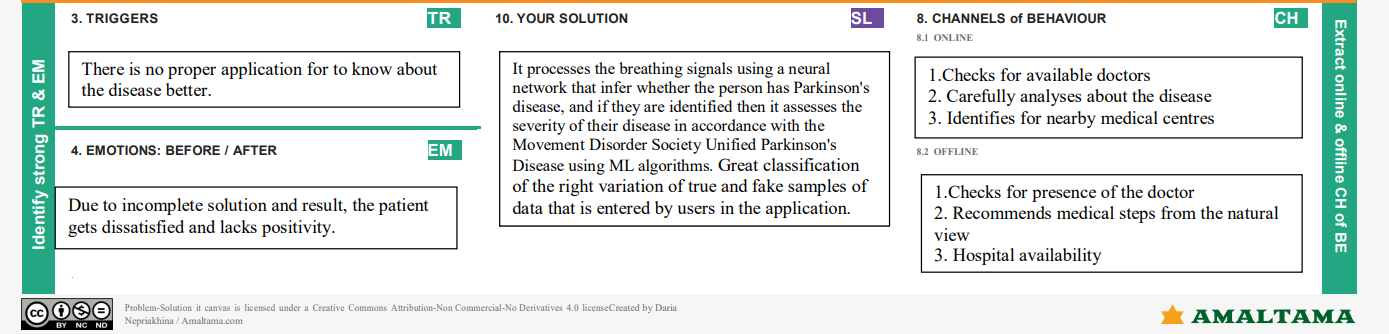
**3.3 Proposed Solution**

It is a classification problem where we must predict whether a person has Parkinson’s disease or not. The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations based on training data. In classification problem, we must predict discrete values based on a given set of independent variables. Binary classification, in this classification we have to predict either of the two given classes. For example: classifying whether a person has Parkinson’s disease or not (yes/no).

By using machine learning methods, the problem can be addressed with very less error rate. The voice dataset of Parkinson's disease from the UCI Machine learning library is used as input. Also, our proposed system provides accurate results by integrating spiral drawing inputs of normal and Parkinson’s affected patients. We propose a hybrid and accurate results analysing patient both voice and spiral drawing data. This application offers medical advice and solutions as the next step after user is confirmed based on the presence of Parkinson’s disease. This can be used direct by medical team for analysing and offering the solutions at much positive scaling time

**3.4 Problem Solution fit**





1. **REQUIREMENT ANALYSIS**

**4.1Functional requirement**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User account registration | Registration through Google account and forms |
| FR-2 | Input data | Application received the data and processes its roles |
| FR-2 | User Authorization | Verifying the user’s account |
| FR-3 | Data classification | Classification of the real data for the user |
| FR-4 | Accuracy verification | Accuracy is determined in the application |
| FR-5 | Time efficient usage | Interaction with the chatbot till the result gets generated for the user |
| FR-6 | Medical recommendations | User receives the medical suggestions and assistance for to offer speed |
| FR-7 | Data extraction | User gets their personal disease report data from the application |

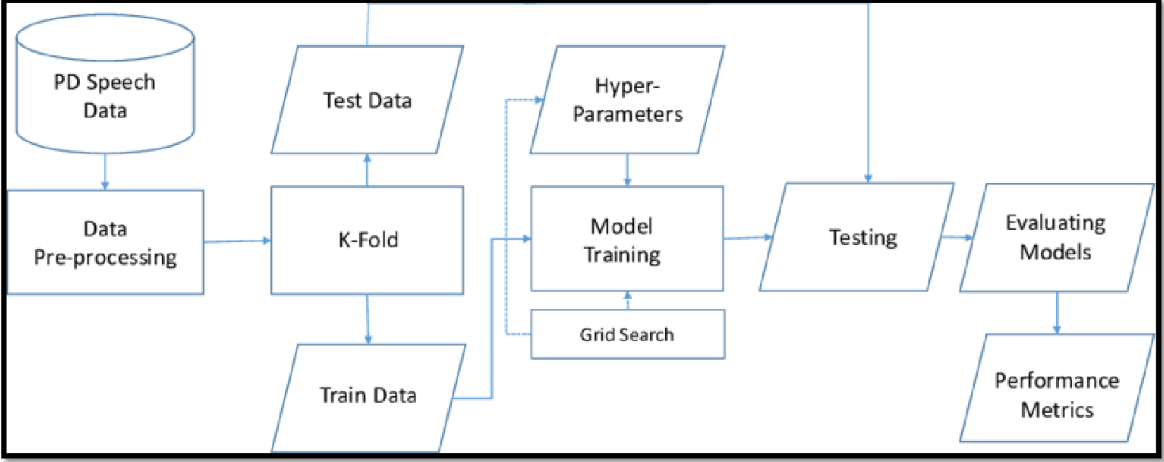
**4.2Non-Functional requirements**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | Usability | The application can be used for accurate prediction and classifier of the true and fake input data sample |
| NFR-2 | Security | User’s data is well encrypted using stable machine learning algorithms |
| NFR-3 | Reliability | The application is monitored periodically in terms of its constant prediction ability, quality, and availability towards the user |
| NFR-4 | Performance | It classifies the images and predicts the disease with careful accuracy output |

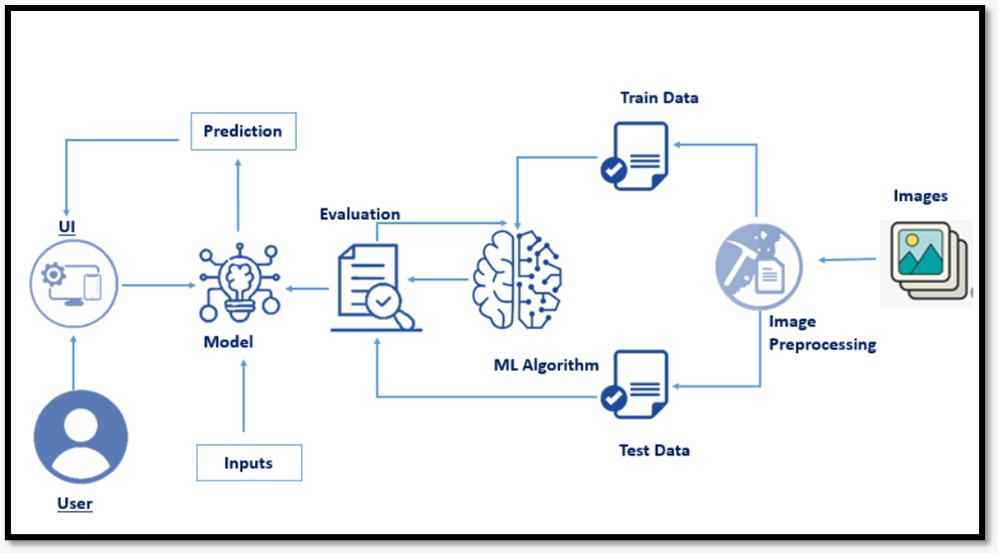
|  |  |  |
| --- | --- | --- |
| NFR-5 | Availability | The application is active throughout the day. While awaiting the prediction result, User can interact with the chatbot for to spend time in knowing important |
| NFR-6 | Scalability | It does not request money or bank details to setup their account and download their final medical result from the application |

1. **PROJECT DESIGN**

**5.1Data Flow Diagrams**



**5.2Solution & Technical Architecture**



**5.3User Stories**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requirement (Epic)** | **User**  **Story**  **Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer (Public user) | Account creation | USN-1 | As a user, I can connect my google into the application | I can access my account / application dashboard | High | Sprint-1 |
| Input data | Adding data | USN-2 | As a user, I can feed my data as the input into the application for it to classify the true fake data | I can cross verify the data that entered in the initial step | High | Sprint-1 |
| Data validation | Checking accuracy | USN-3 | As a user, I can check the ability and accuracy of the model in obtaining the required information | I can log into my account and check  the capability of the model | Medium | Sprint-2 |
| Classification | Data  classification | USN-4 | As a user, I can view the real data | I can verify my data with the real data | Medium | Sprint-2 |
| App work | Work flow | USN-5 | As a user, I can examine the  working action of the application model | I can view how the application works and responds to the actions imposed | High | Sprint-2 |
| Image  classification | Checking for the disease | USN-6 | As a user, I can verify with the application that the image is identified with the actual disease with the help of the trained and tested data’s | I can confirm that the data shows the accurate result | Low | Sprint-3 |
| User interaction | AI powered chatbot | USN-7 | As a user, I can interact with the automated chatbot to engage my time till the application processed the accurate result in a meanwhile | I can see the results from the interaction with the chatbot | Low | Sprint-3 |
| Medical assistance | Medical suggestions | USN-8 | As a user, I can get medical advises and recommendations for to boost the action of curing the disease | I can get enough assistance by getting the suggestions for curing the disease | High | Sprint-3 |
| Data extraction | Obtaining the data | USN-9 | As a user, I can retrieve the result data from the application for data storage for further medical research uses. | I can download the result in the form of data as a proof to show to medical teams | Medium | Sprint-4 |

1. **PROJECT PLANNING & SCHEDULING**

**Sprint Planning & Estimation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| Sprint-1 | Viewing Home Page for the web application | USN-1 | As a user, I can view the home page which has a description of the disease as well as options to sign up or log in. | 4 | Low | Jemimah  Aishwarya  Naresh Sudhan  Aravindhan |
| Sprint-1 | Sign Up Page | USN-2 | As a user, I can register for the application by entering my name, phone number, email, password, and confirming my password. | 4 | High | Jemimah  Aishwarya  Naresh Sudhan  Aravindhan |
| Sprint-1 | Login | USN-3 | As a user, I can log into the application by entering email & password after creation of the account. | 2 | High | Jemimah  Aishwarya  Naresh Sudhan  Aravindhan |
| Sprint-2 | Authorization | USN-4 | As a user, I will receive confirmation email once I have registered for the application. | 6 | High | Jemimah  Aishwarya  Naresh Sudhan  Aravindhan |
| Sprint-2 | Dashboard | USN-5 | As a user, I can research and know the sample disease images of Parkinson. Also collecting sample data to learn more about the disease. | 6 | High | Jemimah  Aishwarya  Naresh Sudhan  Aravindhan |
| Sprint-2 | Data Collection (Dataset) | USN-6 | I need to collect data (images of spirals and waves drawn by healthy people and Parkinson’s patients). | 6 | Medium | Jemimah  Aishwarya  Naresh Sudhan  Aravindhan |
| Sprint-2 | Data checking | USN-7 | I need to learn and understand the data | 2 | Medium | Jemimah  Aishwarya  Naresh Sudhan  Aravindhan |
| Sprint-3 | Data Pre-Processing and EDA | USN-8 | I need to prepare, clean the data, and process the data for modelbuilding by doing pre- processing activities such as EDA and data visualization. | 4 | High | Jemimah  Aishwarya  Naresh Sudhan  Aravindhan |
| Sprint-3 | Data visualization | USN-9 | I need to visualize the data for to check for any outliers and processing the data accordingly. | 7 | Medium | Jemimah  Aishwarya  Naresh Sudhan  Aravindhan |
| Sprint-3 | Model Building (Training and testing) | USN-10 | I need to build the model using Data mining processes such as Random ForestClassifier, K Nearest Neighbor (KNN) from regression, classification, and clustering techniques. | 4 | High | Jemimah  Aishwarya  Naresh Sudhan  Aravindhan |
| Sprint-3 | Assessing the model using metrics | USN-11 | I need to measure the performance of the model using regression metrics | 5 | Medium | Jemimah  Aishwarya  Naresh Sudhan  Aravindhan |
| Sprint-4 | Application Building | USN-12 | I need to build the website for the model application using HTML, CSS, JavaScript etc followed by user sign up page creation in sprint  1. It is then completed by designing the application website. | 4 | Medium | Jemimah  Aishwarya  Naresh Sudhan  Aravindhn |
| Sprint-4 | Model verification | USN-13 | I need to check that model works fine in the application for the user. | 6 | High | Jemimah  Aishwarya  Naresh Sudhan  Aravindhan |
| Sprint-4 | Model Deployment (IBM Cloud) | USN-14 | I need to deploy the Machine Learning model that was built using cloud environment from IBM. And configuring the data of the user in IBM warehouse service called as db2. | 5 | Medium | Jemimah  Aishwarya  Naresh Sudhan  Aravindhan |
| Sprint-4 | Results | USN-15 | As a user, I can receive a diagnosis in addition to recommendations | 5 | High | Jemimah  Aishwarya  Naresh Sudhan  Aravindhan |

**Sprint Delivery Schedule**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on**  **Planned End Date)** | **Sprint Release Date (Actual)** |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 20 | 05 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

1. **CODING & SOLUTIONING**

**import** warnings

warnings**.**filterwarnings("ignore") *#Not to display the warnings*

**import** numpy **as** np

**import** pandas **as** pd

**import** os**,** sys

**from** sklearn.preprocessing **import** MinMaxScaler

**from** xgboost **import** XGBClassifier

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.metrics **import** accuracy\_score *#Modelmetrics*

pip install lux

parkinson\_data **=** pd**.**read\_csv('parkinsons.data')

print(parkinson\_data)

parkinson\_data

parkinson\_data**.**head(n**=**20)

parkinson\_data**.**tail(50)

parkinson\_data**.**shape

*#(rows,columns)*

parkinson\_data**.**isnull()**.**sum()

parkinson\_data**.**describe()**.**round(2)**.**style**.**background\_gradient(cmap**=**'Blues')

parkinson\_data['PPE']**.**tolist()

**import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

variable **=** parkinson\_data["status"]**.**value\_counts()

variable\_data **=** pd**.**DataFrame({'status':variable**.**index,'values':variable**.**values})

sns**.**barplot(x**=**'status',y**=**'values',data**=**variable\_data)

**def** distplots(col):

sns**.**distplot(parkinson\_data[col])

plt**.**show()

**for** i **in** list(parkinson\_data**.**columns)[1:]:

distplots(i)

sns**.**distplot(parkinson\_data["PPE"])

sns**.**distplot(parkinson\_data['D2'])

**def** boxplots(col):

sns**.**boxplot(parkinson\_data[col])

plt**.**show()

**for** i **in** list(parkinson\_data**.**select\_dtypes(exclude**=**["object"])**.**columns)[1:]:

boxplots(i)

plt**.**figure(figsize**=**(20,20))

correlation\_data**=**parkinson\_data**.**corr()

sns**.**heatmap(correlation\_data,annot**=True**)

x **=** parkinson\_data**.**drop(["status","name"],axis**=**1)

y **=** parkinson\_data["status"]

**from** imblearn.over\_sampling **import** RandomOverSampler

**from** imblearn.under\_sampling **import** RandomUnderSampler

**from** collections **import** Counter *#For priortizing the importance to store elements as dictionary keys, and their counts as values.*

print(Counter(y))

ROS **=** RandomOverSampler() *#To compensate the imbalance part present in the data*

x\_ROS,y\_ROS **=** ROS**.**fit\_resample(x, y)

print(Counter(y\_ROS))

Scaler\_data **=** MinMaxScaler((**-**1,1))

x **=** Scaler\_data**.**fit\_transform(x\_ROS)

y **=** y\_ROS

**from** sklearn.decomposition **import** PCA

Princple\_CA **=** PCA(.95)

X\_PCA **=** Princple\_CA**.**fit\_transform(x)

print(x**.**shape)

print(X\_PCA**.**shape)

x\_train,x\_test,y\_train,y\_test **=** train\_test\_split(X\_PCA,y, test\_size**=**0.2, random\_state**=**7)

**from** sklearn.metrics **import** confusion\_matrix, accuracy\_score, f1\_score

List\_metrics **=** []

List\_accuracy **=** []

*#Logistic Regression*

**from** sklearn.linear\_model **import** LogisticRegression

Classification\_model **=** LogisticRegression(C**=**0.4,max\_iter**=**1000,solver**=**'liblinear')

Log\_Regression **=** Classification\_model**.**fit(x\_train, y\_train)

y\_pred **=** Classification\_model**.**predict(x\_test) *#Prediction*

Log\_Regression\_accuracy **=** accuracy\_score(y\_test, y\_pred) *#Accuracy*

print("The accuracy score with Logistic regression is:",Log\_Regression\_accuracy)

*#Decision Tree Classificaton using supervised machine learning for classifiying the data with confident accuracy*

**from** sklearn.tree **import** DecisionTreeClassifier

Classification\_tree **=** DecisionTreeClassifier(random\_state**=**14)

Decision\_tree **=** Classification\_tree**.**fit(x\_train, y\_train)

y\_pred2 **=** Classification\_tree**.**predict(x\_test) *#Prediction*

Dec\_tree\_accuracy **=** accuracy\_score(y\_test, y\_pred2) *#Accuracy*

print("The accuracy score with Decision Tree Classifier is:",Dec\_tree\_accuracy)

*#Random Forest Classifier is used for its high dimensionality and accuracy capabilities, here information gain is priortized*

**from** sklearn.ensemble **import** RandomForestClassifier

Classification\_random **=** RandomForestClassifier(random\_state**=**14)

RFE **=** Classification\_random**.**fit(x\_train, y\_train)

y\_pred3 **=** Classification\_random**.**predict(x\_test) *#Prediction*

Ran\_For\_accuracy **=** accuracy\_score(y\_test, y\_pred3) *#Accuracy*

print("The accuracy score with Random Forest Classifier(Information gain) is:",Ran\_For\_accuracy)

*#Random Forest Classifier with entropy condition*

**from** sklearn.ensemble **import** RandomForestClassifier

Classification\_entropy **=** RandomForestClassifier(criterion**=**'entropy')

RFE **=** Classification\_entropy**.**fit(x\_train,y\_train)

y\_pred4 **=** Classification\_entropy**.**predict(x\_test)

Random **=** accuracy\_score(y\_test, y\_pred4)

print("The accuracy score with Random Forest Classifier(Entropy) is:",Random)

*#Using Support Vector Machine (SVM) for to enhance the similarity and to increase the scaling factor of the model*

**from** sklearn.svm **import** SVC

Parkinson\_model **=** SVC(cache\_size**=**100)

Support\_vector\_machine **=** Parkinson\_model**.**fit(x\_train, y\_train)

y\_pred5 **=** Parkinson\_model**.**predict(x\_test)

Support\_accuracy **=** accuracy\_score(y\_test, y\_pred5)

print("The accuracy score with Support Vector Machine is:",Support\_accuracy)

*#K Nearest Neighbor Classifier for better effectiveness*

**from** sklearn.neighbors **import** KNeighborsClassifier

KNN\_parkinson **=** KNeighborsClassifier(n\_neighbors**=**3)

K\_Nearest\_Neighbor\_Classifier **=** KNN\_parkinson**.**fit(x\_train, y\_train)

KNN\_predict **=** KNN\_parkinson**.**predict(x\_test)

KNN\_accuracy **=** accuracy\_score(y\_test, KNN\_predict)

print("The accuracy score with K Nearest Neighbor Algorithm is:",KNN\_accuracy)

*#GaussianNB*

**from** sklearn.naive\_bayes **import** GaussianNB

GNB **=** GaussianNB()

Model\_NB **=** GNB**.**fit(x\_train,y\_train)

pred\_gnb **=** Model\_NB**.**predict(x\_test)

GNB\_accuracy **=** accuracy\_score(y\_test, pred\_gnb)

print("The accuracy score with Gaussian Naive Bayes is:",GNB\_accuracy)

print("\nLet's see the overall accuracy of the built model that is been created below, view the overall accuracy score below!")

Overall\_accuracy\_percentage **=** Log\_Regression\_accuracy**+**Dec\_tree\_accuracy**+**Ran\_For\_accuracy**+**Random**+**Support\_accuracy**+**KNN\_accuracy**+**GNB\_accuracy

Average\_accuracy **=** (Overall\_accuracy\_percentage)**/**7

print("The accuracy of all the combined metrics for the model is:",Average\_accuracy**/**0.01)

**from** sklearn.ensemble **import** VotingClassifier

VC **=** VotingClassifier(estimators**=**[('Classification\_model',Classification\_model),('Classification\_tree',Classification\_tree),('Classification\_random',Classification\_random),('Classification\_entropy',Classification\_entropy),('Support\_vector\_machine',Support\_vector\_machine),('K\_Nearest\_Neighbor\_Classifier',K\_Nearest\_Neighbor\_Classifier),('Model\_NB',Model\_NB)],voting**=**'hard',flatten\_transform**=True**)

Model\_VC **=** VC**.**fit(x\_train, y\_train)

Model\_prediction **=** VC**.**predict(x\_test)

Model\_accuracy **=** accuracy\_score(y\_test,pred\_gnb)

print(Model\_accuracy)

Model\_XG **=** XGBClassifier(random\_state**=**0)

Model\_XG**.**fit(x\_train,y\_train)

y\_predict **=** Model\_XG**.**predict(x\_test)

print(accuracy\_score(y\_test,y\_predict)**\***100)

1. **RESULTS**

**F1 score:** 97.14285714285714

Classification report for Model XGBClassifier():

precision recall f1-score support

0 0.74 0.83 0.78 24

1 0.88 0.80 0.84 35

accuracy 0.81 59

macro avg 0.81 0.82 0.81 59

weighted avg 0.82 0.81 0.81 59

1. **ADVANTAGES**

Accurately diagnosing PD is important so that patients can receive the proper treatment and advice regarding care. In addition, diagnosing PD early is important because treatments such as levodopa/carbidopa are more effective when administered early on in the disease.

Advantages including the Real Time Speech Analysis which perfectly shows how noise and other factors affect Parkinson's Disease Detection

**DISADVANTAGES**

Due to the high inter-study variance in the data source and presentation of results, it was challenging to directly compare outcomes associated with each type of model across studies, as some studies failed to indicate whether model performance was evaluated using a test set, and/or results given by models that did not yield the best per-study performance. Results of published studies were discussed and summarized based on data and machine learning models used, and for data modalities such as PET (n = 4) or CSF (n = 5), the number of studies were too small despite the high total number of studies included. Therefore, it was improbable to assess the general performance of machine learning techniques when PET or CSF data are used.

1. **CONCLUSION**

Managing PD in day-to-day life is very challenging for an individual. Therefore, a good screening procedure will be beneficial, especially in circumstances where a physician’s treatment is not necessary. Thus, for the diagnosis of PD, ML algorithms were evaluated. The main aim of this review was to identify existing ML-based research to diagnose PD in terms of handwritten patterns, voice attributes, and gait dataset and to determine the most appropriate technique to diagnose the PD with an accuracy rate.

1. **FUTURE SCOPE**

This scoping review summarized studies by investigating the use of neural networks, specifically deep learning algorithms, for early diagnosis of PD based on various data collected from different public and private sources (91 studies), including medical image, biomedical voice, and sensor signal, for both PD and healthy control samples. Included studies were categorized into different groups based on the neural network model, type of PD symptoms, and type of dataset. Additionally, the most used dataset and best performance model were highlighted based on the detection of particular symptoms of PD in this review. All technical experiment methods were reported, including submodel, dataset volume, training, testing, evaluation metrics, and validation type. We indicated any real-time implementation used in each hospital or university setting, and based on this review, we recommended particular suggestions for healthcare professionals. Future work could be a meta-analysis to examine each study and provide a comprehensive comparison between them in terms of quality.

1. **APPENDIX**

https://drive.google.com/file/d/1T5Qi\_U2MPoroZ1i9lCpKdWE8eMT3pk-8/view?usp=sharing