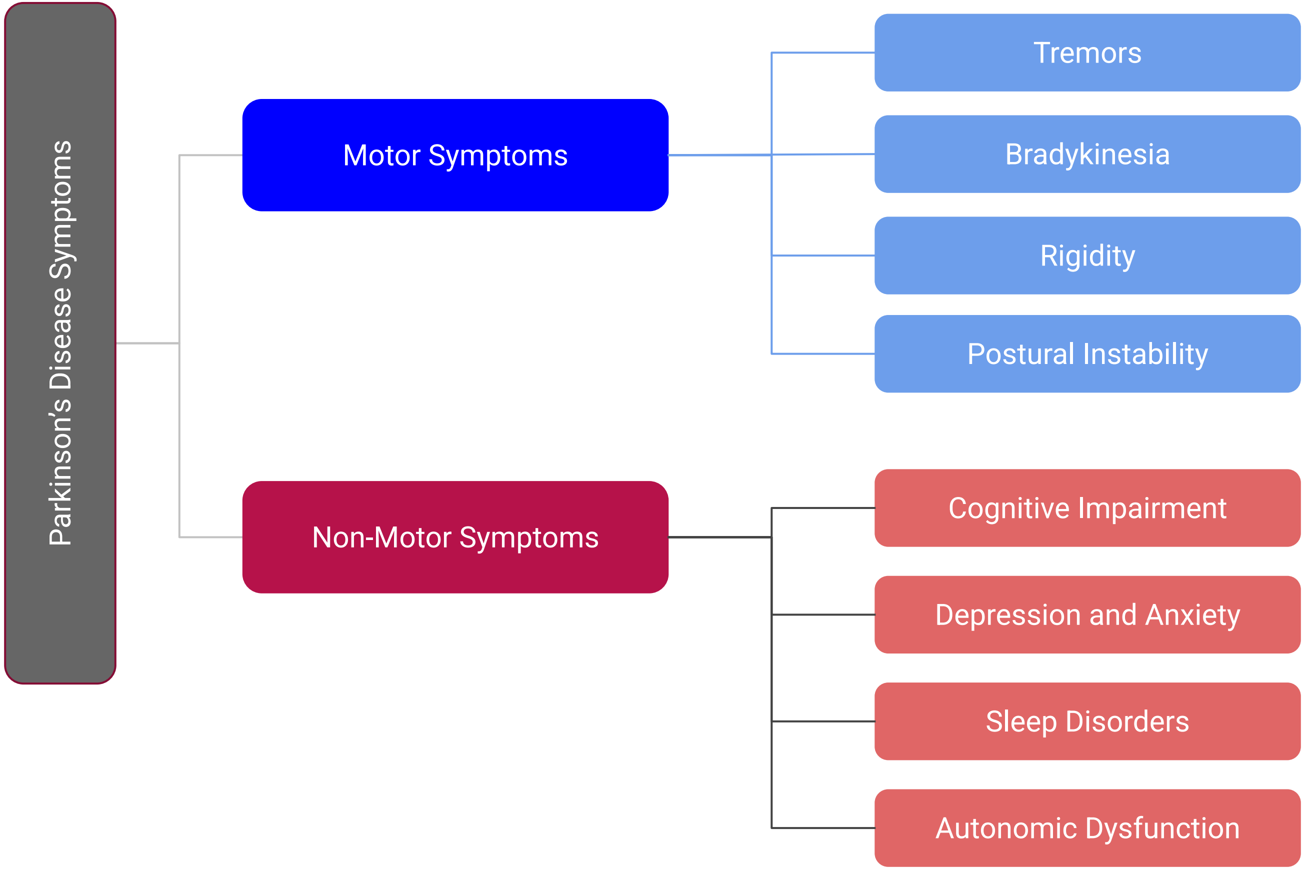
Improving Parkinson’s Diagnosis with Artificial Intelligence

Harnessing Data Science for Early Detection and Better Outcomes

Contributors: Chris Alvizures, Diya Sadekar, Paige Andrews, Rowan Clark, Shalini Pampati, Tugce Terzioglu

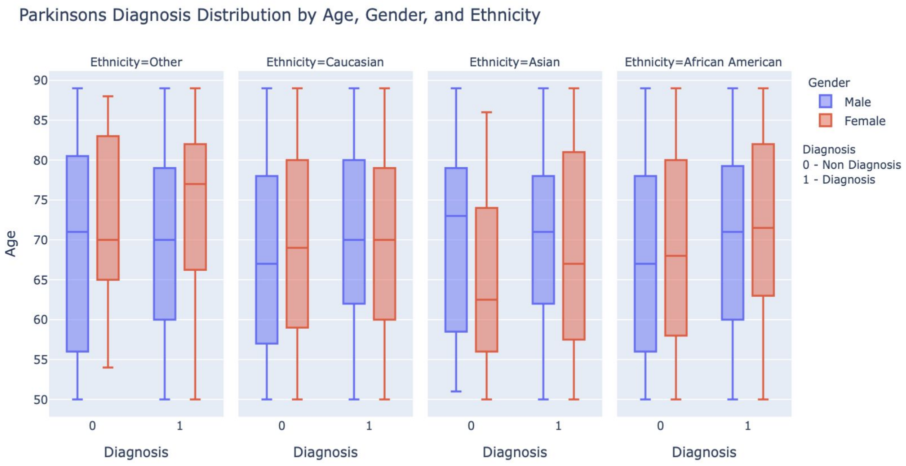
Understanding Parkinson’s Disease



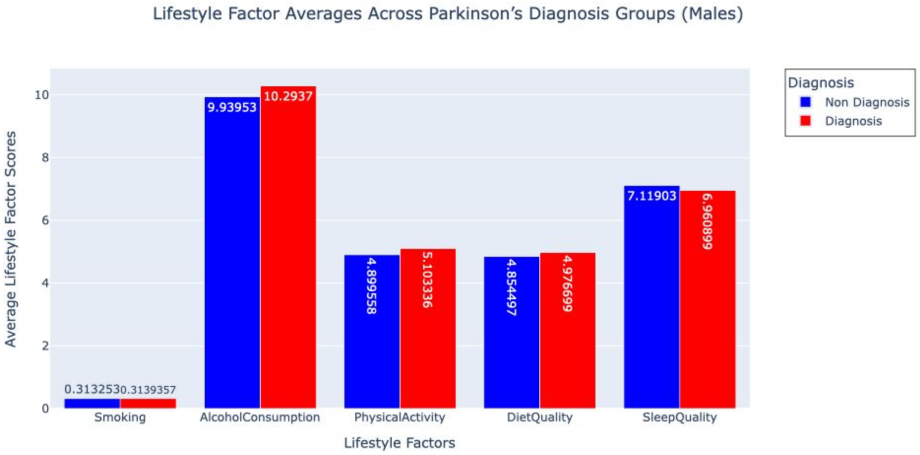
Unified Parkinson’s Disease Rating Scale(MDS-UPDRS)

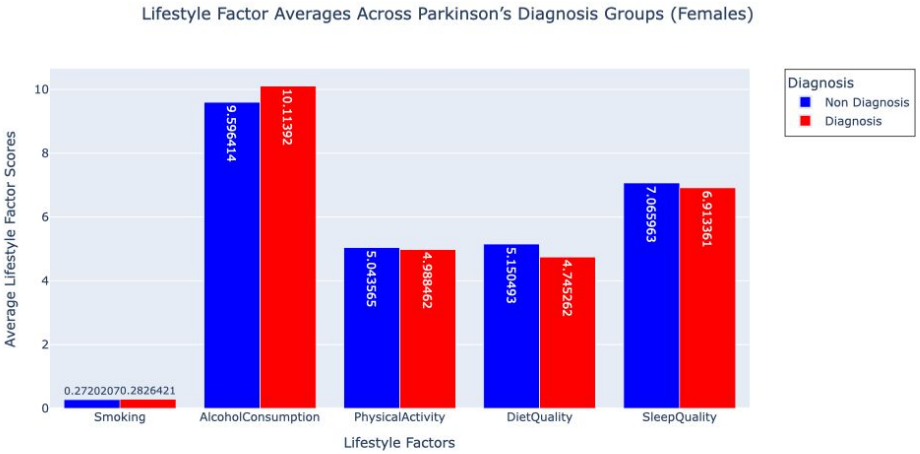
The **MDS-UPDRS** is a widely-used clinical tool to measure the severity and progression of Parkinson’s disease

- 50 question assessment divided into **4 parts**: Non-Motor Experiences, Motor Experiences, Motor Examination, and Motor Complications
- **Scoring**: each item is scored from **0 (normal)** to **4 (severe disability)**
- Commonly used in clinical trials and research to track disease progression and evaluate treatment effectiveness

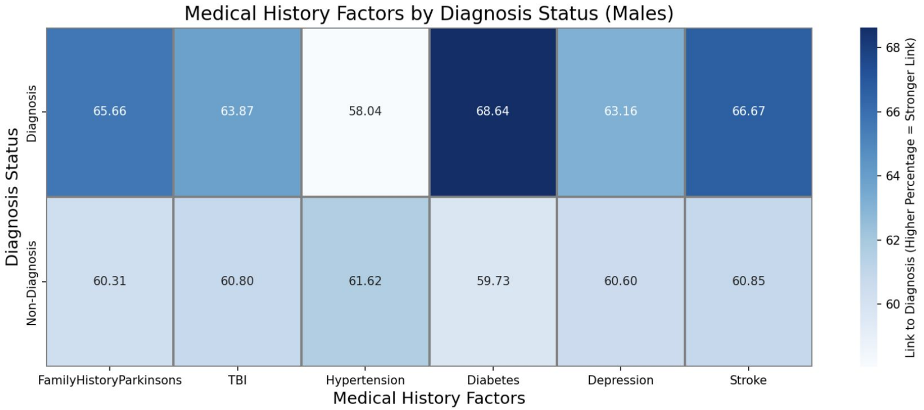


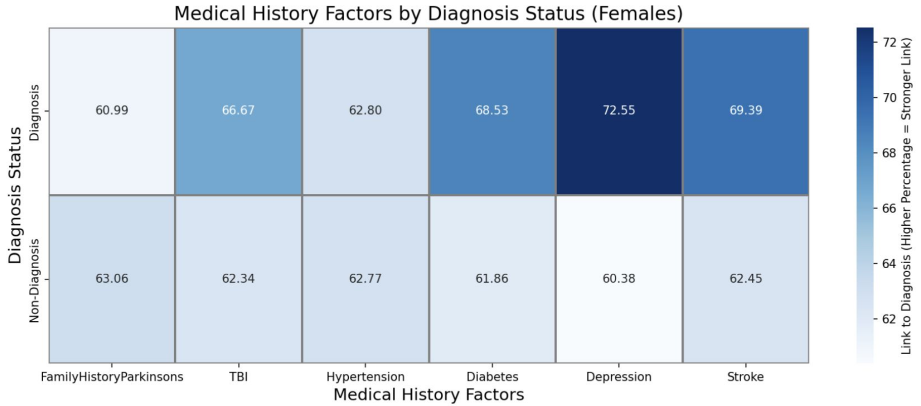
- Age is a strong predictor, older individuals are more likely to be diagnosed
- Males show a higher diagnosis rate than females across most ethnicities
- Some ethnic groups may have different distributions, suggesting potential genetic or environmental influences



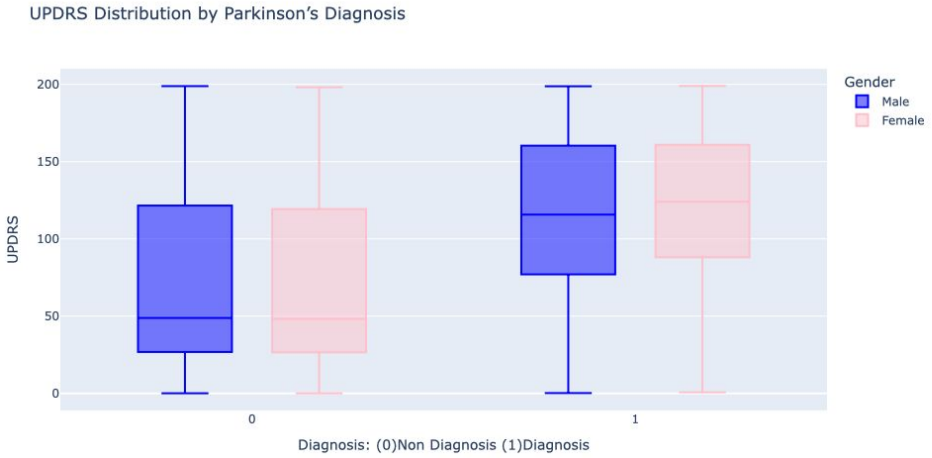


- Lower physical activity and poor sleep quality seem associated with higher diagnosis rates
- Diet quality might also differ between diagnosed and non-diagnosed groups
- Gender differences exist, some lifestyle factors might be more relevant for men versus women

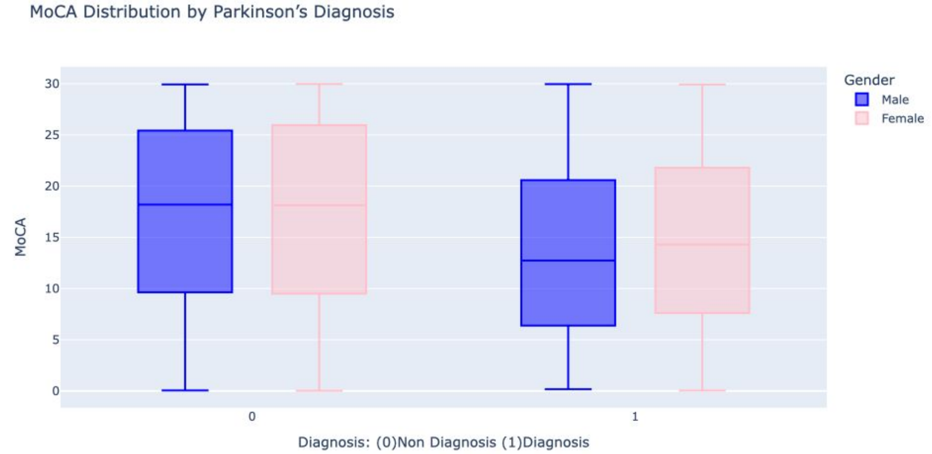


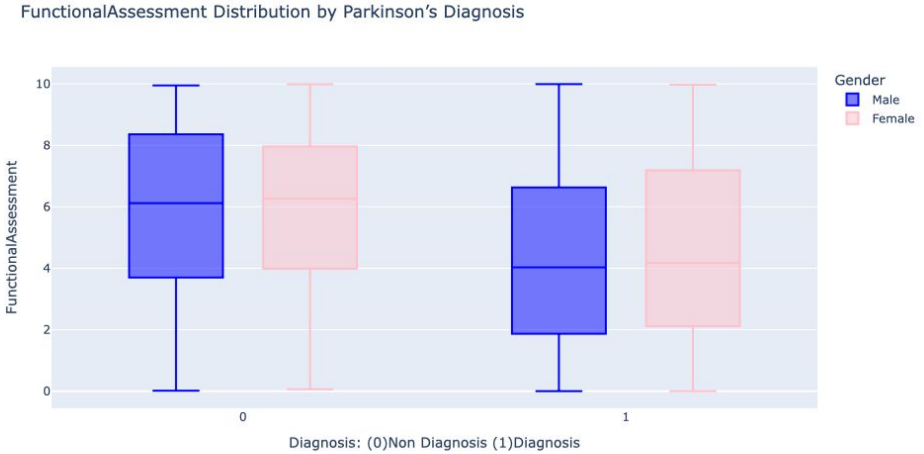


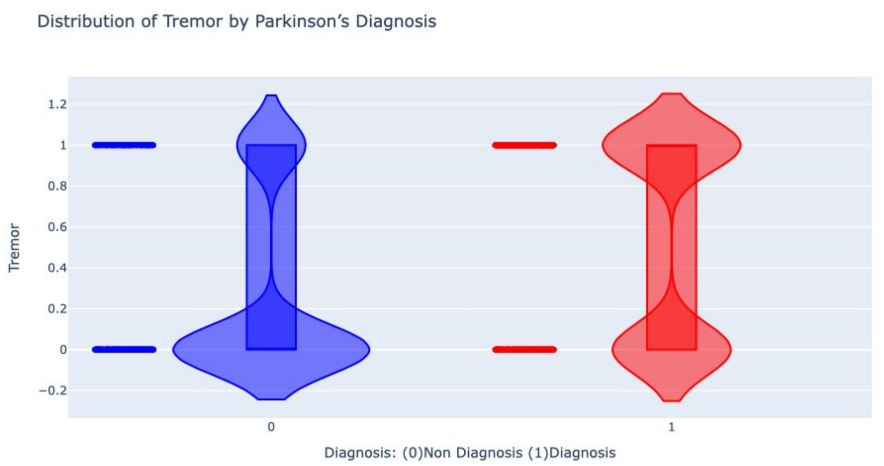
- Family history is a strong predictor of Parkinson's
- Conditions like traumatic brain injury (TBI), hypertension, and depression show a noticeable correlation with diagnosis
- Males with TBI and hypertension have a higher diagnosis rate, whereas depression seems more influential for females

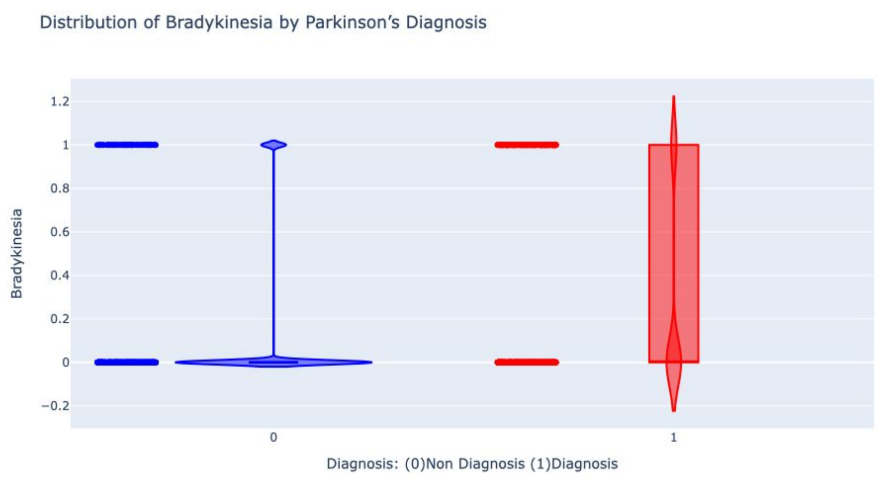


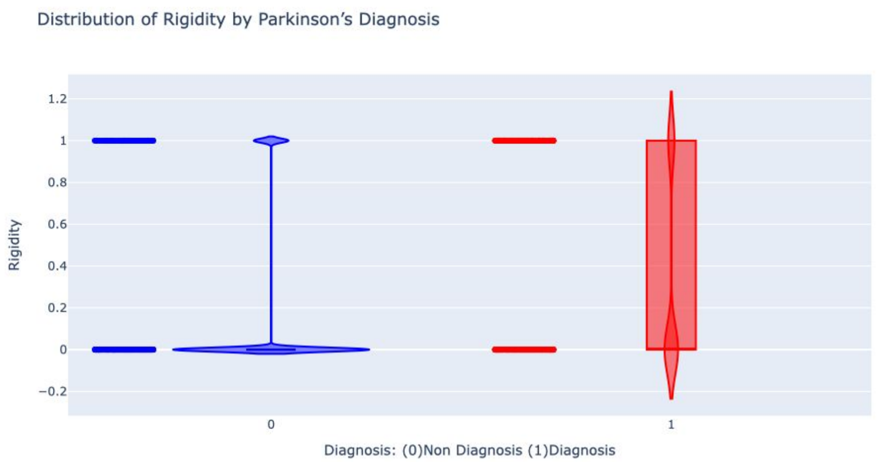
- Diagnosed individuals have worse scores in functional & cognitive assessments
- Symptoms like tremor, rigidity, and bradykinesia strongly separate diagnosed individuals from non-diagnosed
- Clinical evaluation remains crucial for early detection

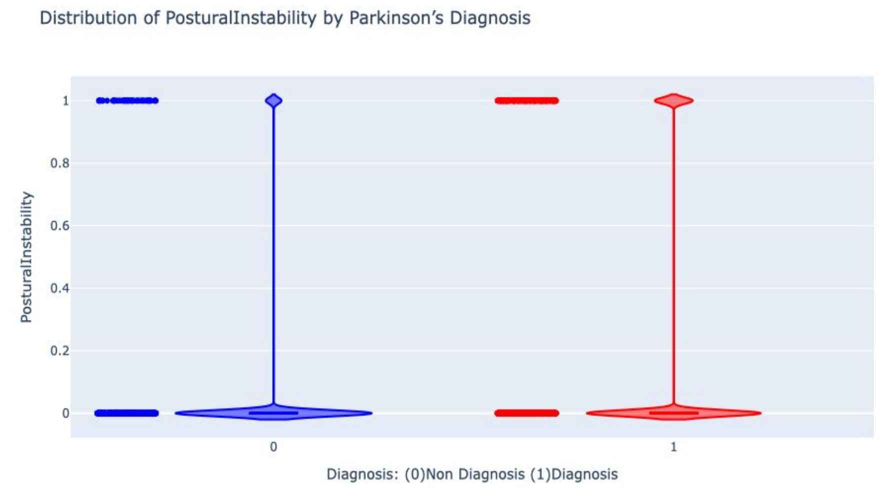


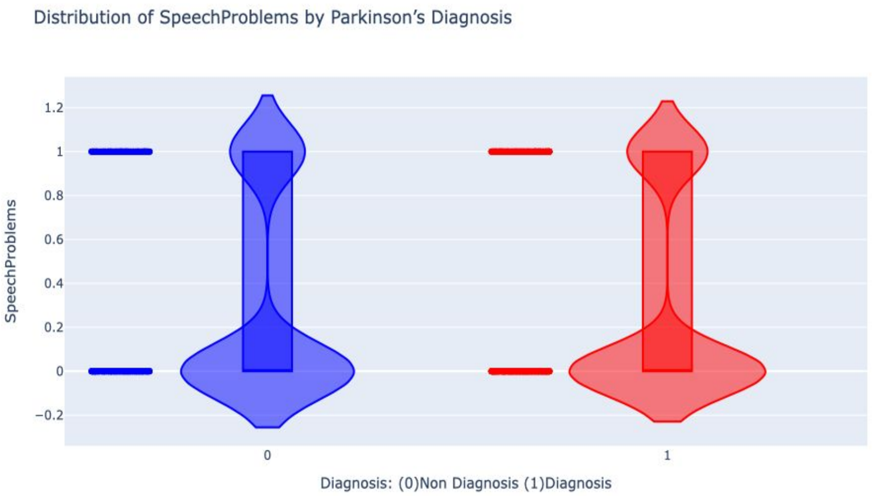


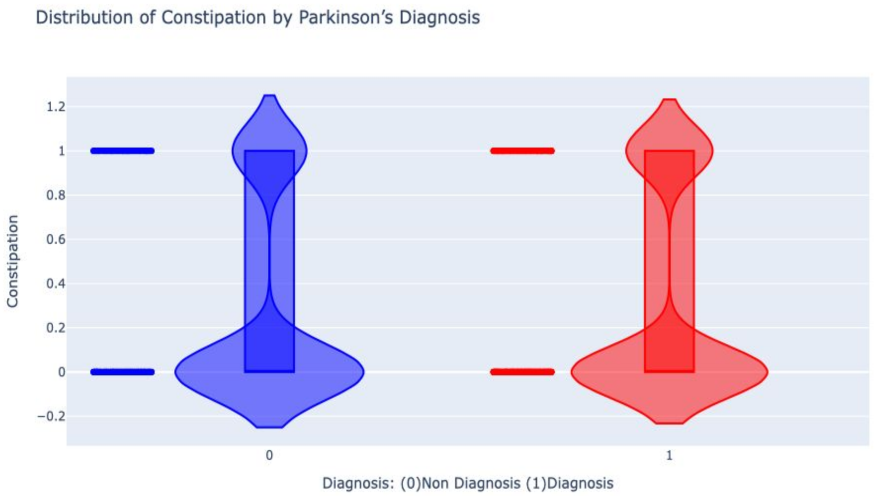

Distribution of Tremor by Parkinson's Diagnosis
Tremor
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
0
1
Diagnosis: (0)Non Diagnosis (1)Diagnosis

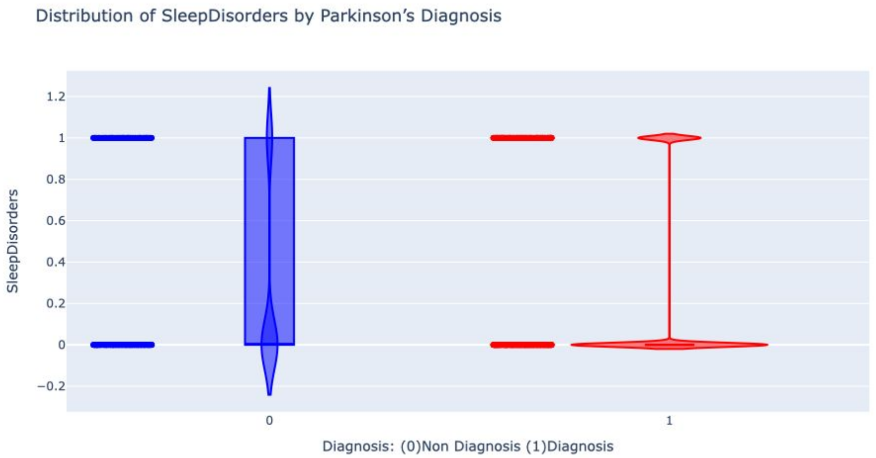

Distribution of Bradykinesia by Parkinson's Diagnosis
Bradykinesia
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
0
1
Diagnosis: (0)Non Diagnosis (1)Diagnosis


Distribution of Rigidity by Parkinson's Diagnosis
Rigidity
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
0
1
Diagnosis: (0)Non Diagnosis (1)Diagnosis


Distribution of PosturalInstability by Parkinson's Diagnosis
PosturalInstability
1
0.8
0.6
0.4
0.2
0
0
1
Diagnosis: (0)Non Diagnosis (1)Diagnosis


Distribution of SpeechProblems by Parkinson's Diagnosis
SpeechProblems
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
0
1
Diagnosis: (0)Non Diagnosis (1)Diagnosis


Distribution of Constipation by Parkinson's Diagnosis
Constipation
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
0
1
Diagnosis: (0)Non Diagnosis (1)Diagnosis


Distribution of SleepDisorders by Parkinson's Diagnosis
SleepDisorders
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
0
1
Diagnosis: (0)Non Diagnosis (1)Diagnosis

DataFrame Creation

This dataset comprises detailed medical records of 2,105 Parkinson's disease patients. It covers a wide range of critical information, including:

- **Demographics:** Age, gender, and education level
- **Lifestyle Factors:** Smoking habits, alcohol consumption, and physical activity
- **Medical History:** Conditions such as hypertension and diabetes
- **Clinical Measurements:** Blood pressure, cholesterol levels, and functional assessments
- **Cognitive & Functional Evaluations:** UPDRS and MoCA scores
- **Symptoms:** Tremor, rigidity, and speech problems
- **Diagnosis Indicator:** Parkinson's presence (binary classification)

Data Cleaning and Preprocessing:

- Removal of the unnecessary Columns like Doctor in charge and Patient ID
- Cleaning up the duplicate values

Machine Learning Model Objective

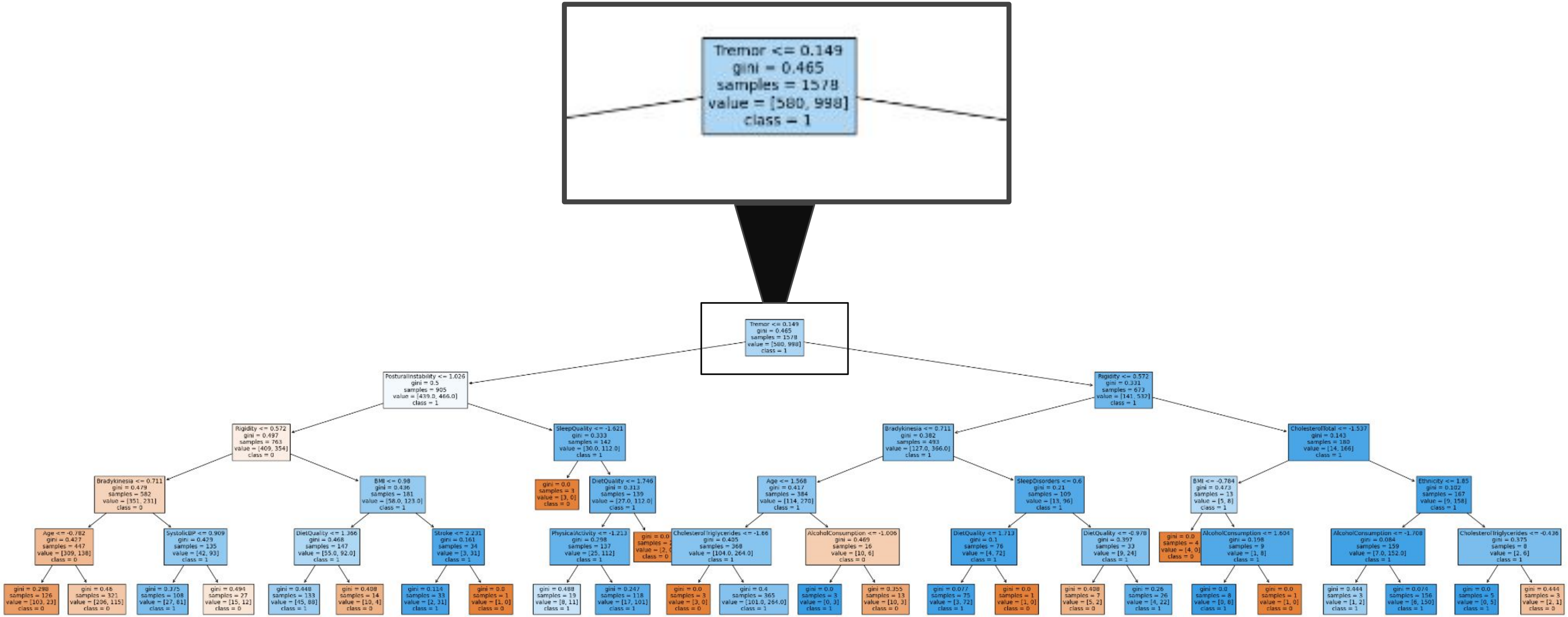
Improving diagnostic accuracy and early detection with machine learning

- Predictive model assesses Parkinson’s likelihood using comprehensive patient data and diagnostic results
- Trained on health factors, medical history, and cognitive/functional assessments for reliable predictions
- Goal: Achieve 85%+ accuracy to enable early detection and timely clinical intervention
- Bottom line: Enhance diagnostic precision to improve outcomes and support healthcare professionals

Deciding Which Model to Use

Classification Models	Accuracy Score
Logistic Regression	0.8292
K-Nearest Neighbor	0.72
Decision Tree	0.67
Neural Network	0.8216

Final Choice: **Neural Network** for most room for optimization


Tremor <= 0.149
gini = 0.465
samples = 1578
value = [580, 998]
class = 1
PosturalInstability <= 1.026
gini = 0.5
samples = 905
value = [439.0, 466.0]
class = 1
Rigidity <= 0.572
gini = 0.331
samples = 673
value = [141, 532]
class = 1
Rigidity <= 0.572
gini = 0.497
samples = 763
value = [409, 354]
class = 0
SleepQuality <= -1.621
gini = 0.333
samples = 142
value = [30.0, 112.0]
class = 1
Bradykinesia <= 0.711
gini = 0.382
samples = 493
value = [127.0, 366.0]
class = 1
CholesterolTotal <= -1.537
gini = 0.143
samples = 180
value = [14, 166]
class = 1
Bradykinesia <= 0.711
gini = 0.479
samples = 582
value = [351, 231]
class = 0
BMI <= 0.98
gini = 0.436
samples = 181
value = [58.0, 123.0]
class = 1
gini = 0.0
samples = 3
value = [3, 0]
class = 0
DietQuality <= 1.746
gini = 0.313
samples = 139
value = [27.0, 112.0]
class = 1
Age <= 1.568
gini = 0.417
samples = 384
value = [114, 270]
class = 1
SleepDisorders <= 0.6
gini = 0.21
samples = 109
value = [13, 96]
class = 1
BMI <= -0.784
gini = 0.473
samples = 13
value = [5, 8]
class = 1
Ethnicity <= 1.85
gini = 0.102
samples = 167
value = [9, 158]
class = 1
Age <= -0.782
gini = 0.427
samples = 447
value = [309, 138]
class = 0
SystolicBP <= 0.909
gini = 0.429
samples = 135
value = [42, 93]
class = 1
DietQuality <= 1.366
gini = 0.468
samples = 147
value = [55.0, 92.0]
class = 1
Stroke <= 2.231
gini = 0.161
samples = 34
value = [3, 31]
class = 1
PhysicalActivity <= -1.213
gini = 0.298
samples = 137
value = [25, 112]
class = 1
CholesterolTriglycerides <= -1.66
gini = 0.405
samples = 368
value = [104.0, 264.0]
class = 1
AlcoholConsumption <= -1.006
gini = 0.469
samples = 16
value = [10, 6]
class = 0
DietQuality <= 1.713
gini = 0.1
samples = 76
value = [4, 72]
class = 1
DietQuality <= -0.978
gini = 0.397
samples = 33
value = [9, 24]
class = 1
gini = 0.0
samples = 4
value = [4, 0]
class = 0
AlcoholConsumption <= 1.604
gini = 0.198
samples = 9
value = [1, 8]
class = 1
AlcoholConsumption <= -1.708
gini = 0.084
samples = 159
value = [7.0, 152.0]
class = 1
CholesterolTriglycerides <= -0.436
gini = 0.375
samples = 8
value = [2, 6]
class = 1
gini = 0.298
samples = 126
value = [103, 23]
class = 0
gini = 0.46
samples = 321
value = [206, 115]
class = 0
gini = 0.375
samples = 108
value = [27, 81]
class = 1
gini = 0.494
samples = 27
value = [15, 12]
class = 0
gini = 0.448
samples = 133
value = [45, 88]
class = 1
gini = 0.408
samples = 14
value = [10, 4]
class = 0
gini = 0.114
samples = 33
value = [2, 31]
class = 1
gini = 0.0
samples = 1
value = [1, 0]
class = 0
gini = 0.488
samples = 19
value = [8, 11]
class = 1
gini = 0.247
samples = 118
value = [17, 101]
class = 1
gini = 0.0
samples = 3
value = [3, 0]
class = 0
gini = 0.4
samples = 365
value = [101.0, 264.0]
class = 1
gini = 0.0
samples = 3
value = [0, 3]
class = 1
gini = 0.355
samples = 13
value = [10, 3]
class = 0
gini = 0.077
samples = 75
value = [3, 72]
class = 1
gini = 0.0
samples = 1
value = [1, 0]
class = 0
gini = 0.408
samples = 7
value = [5, 2]
class = 0
gini = 0.26
samples = 26
value = [4, 22]
class = 1
gini = 0.0
samples = 8
value = [0, 8]
class = 1
gini = 0.0
samples = 1
value = [1, 0]
class = 0
gini = 0.444
samples = 3
value = [1, 2]
class = 1
gini = 0.074
samples = 156
value = [6, 150]
class = 1
gini = 0.0
samples = 5
value = [0, 5]
class = 1
gini = 0.444
samples = 3
value = [2, 1]
class = 0

Initial Model and Optimization

Initial Model

Compile, Train, and Evaluate Initial Model

```
# Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.
number_input_features = len(X_train.columns)
hidden_nodes_layer1= 25
hidden_nodes_layer2=10

nn = tf.keras.models.Sequential()

# First hidden layer
nn.add(
    tf.keras.layers.Dense(units=hidden_nodes_layer1, input_dim=number_input_features, activation="relu"))

# Second hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer2, activation="relu"))

# Output layer
nn.add(tf.keras.layers.Dense(units=1, activation="sigmoid"))

# Check the structure of the model
nn.summary()
```

In [9]:

```
# Evaluate the model using the test data
model_loss, model_accuracy = nn.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
17/17 - 0s - 18ms/step - accuracy: 0.8216 - loss: 0.4628
Loss: 0.4627760946750641, Accuracy: 0.8216318488121033
```

Accuracy: 0.8216

Optimized Model

Binning of data

In [4]:

```
#Change 'UPDRS' from integer to binned ranges
bins= [0, 25, 50, 75, 100, 125, 150, 175, 200]
labels = ['0-25',
          '25.1-50',
          '50.1-75',
          '75.1-100',
          '100.1-125',
          '125.1-150',
          '150.1-175',
          '175.1-200']

data_df['UPDRS_BINNED'] = pd.cut(data_df['UPDRS'], bins=bins, labels=lab

#Drop 'UPDRS' from the DataFrame so as not to be included in the feature
data_df.drop(["UPDRS"], axis=1, inplace=True)
data_df.head()
```

In [6]:

```
#Change 'MoCA' from integer to binned ranges
bins= [0, 5, 10, 15, 20, 25, 30]
labels = ['0-5',
          '5.1-10',
          '10.1-15',
          '15.1-20',
          '20.1-25',
          '25.1-30']

data_df['MoCA_BINNED'] = pd.cut(data_df['MoCA'], bins=bins, labels=label

#Drop 'MoCA' from the DataFrame so as not to be included in the features
data_df.drop(["MoCA"], axis=1, inplace=True)
data_df.head()
```

In [14]:

```
# Evaluate the model using the test data
model_loss, model_accuracy = nn.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
17/17 - 0s - 18ms/step - accuracy: 0.8824 - loss: 0.3852
Loss: 0.3852328062057495, Accuracy: 0.8823529481887817
```

Accuracy: 0.8823

Optimized Model without Diagnostic Tests

- Removed "UPDRS","MoCA" and "FunctionalAssessment" columns
- Remaining columns are all health markers or symptoms
- Initial accuracy of 0.6231
- Optimized with binning of 'AlcoholConsumption' column

```
In [11]:
#Drop columns with diagnostic testing scores
data_df.drop(["Diabetes"], axis=1, inplace=True)
data_df.columns

Out[11]: Index(['Age', 'Gender', 'Ethnicity', 'EducationLevel', 'BMI', 'Smoking',
       'AlcoholConsumption', 'PhysicalActivity', 'DietQuality', 'SleepQuali
ty',
       'FamilyHistoryParkinsons', 'TraumaticBrainInjury', 'Hypertension',
       'Depression', 'Stroke', 'SystolicBP', 'DiastolicBP', 'CholesterolTot
al',
       'CholesterolLDL', 'CholesterolHDL', 'CholesterolTriglycerides',
       'Tremor', 'Rigidity', 'Bradykinesia', 'PosturalInstability',
       'SpeechProblems', 'SleepDisorders', 'Constipation', 'Diagnosis'],
      dtype='object')
```

```
In [21]:
# Evaluate the model using the test data
model_loss, model_accuracy = nn.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")

17/17 - 0s - 18ms/step - accuracy: 0.6679 - loss: 0.7380
Loss: 0.7380037307739258, Accuracy: 0.6679316759109497
```

Conclusion

Model Selection & Optimization:

- Neural network model fit best due to high initial accuracy (0.8216) and optimization potential
- Binning the diagnostic data improved accuracy significantly (0.8823) and reduced loss (0.3852)

When Diagnostic Tests are Excluded:

- Using only health markers and symptoms achieved an accuracy of 0.6679 and suffered from higher loss (0.7380)
- Diagnostic testing is of critical importance in order to achieve high predictive accuracy

Broader Impact:

- Early detection tools utilizing machine learning can improve patient care and outcomes, assisting professionals where traditional diagnostic methods fall short
- Our model highlights how combining health data with diagnostic testing uncovers patterns not easily identifiable by human analysis

Further Recommendations

Building on Our Model to Advance Real-World Impact

- Continue adding to the dataset (only 2,105 patients currently) for more model comprehensiveness
- Investigating other factors that influence diagnosis
- Develop an interactive dashboard using Flask for healthcare professionals:
    - Display predictions
    - Feature importance
    - Confidence intervals