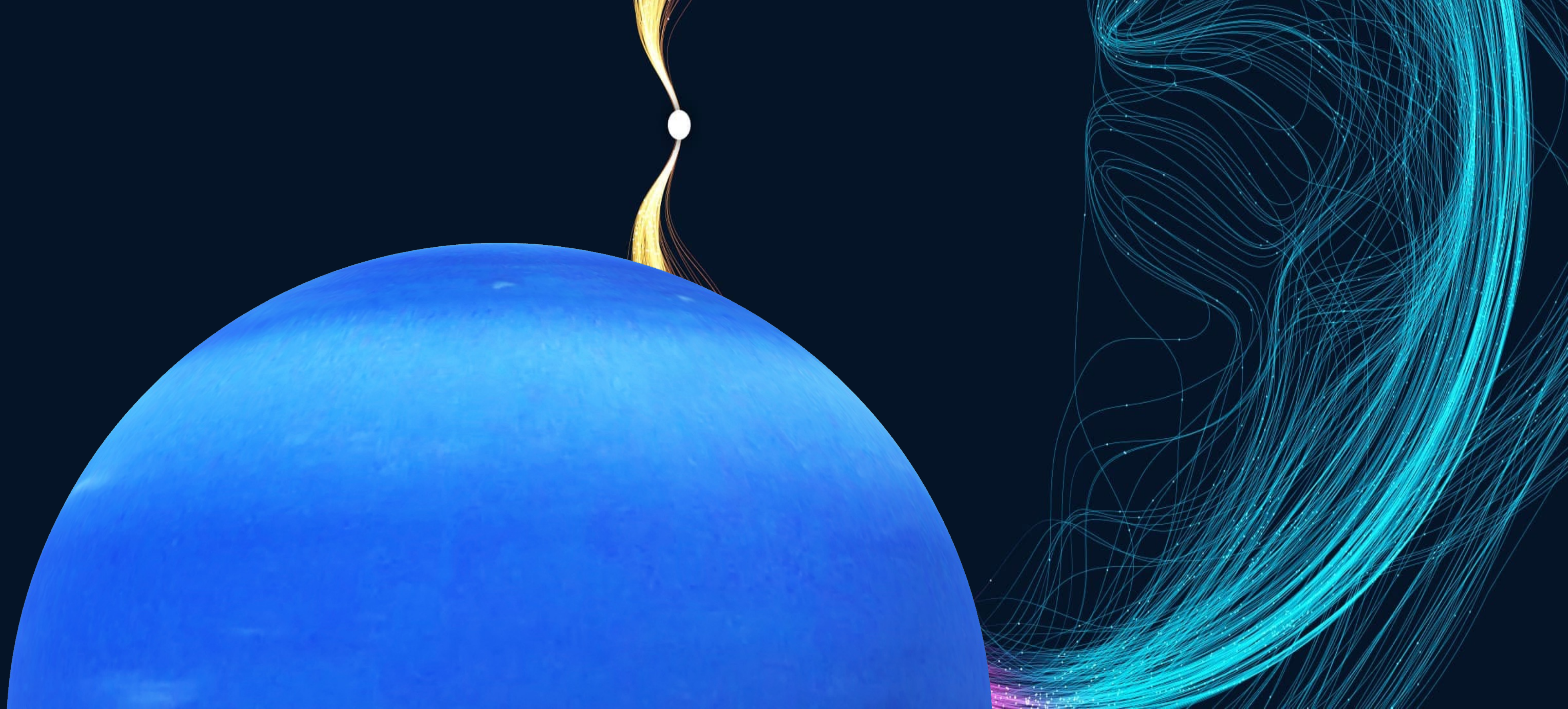
Detection of Parkinson's Disease



Detection of Parkin… isease

Par… se

Par… se

Introduction

What is Parkinson's Disease (PD)?

A progressive neurodegenerative disorder affecting motor function and quality of life.

Early detection is critical for effective management and improved patient outcomes.

Why Voice Analysis?

Vocal changes are among the earliest indicators of PD.

Machine learning enables us to analyze subtle voice features for predictive diagnostics.

Objective:

To develop a tool that predicts Parkinson’s disease based on voice recordings, using machine learning and a user-friendly GUI.

Problem Statement

Challenges in Early Diagnosis:
- Subtle symptoms in early stages are often missed in traditional clinical assessments.
- Delayed diagnosis impacts treatment efficacy and quality of life.

Proposed Solution:
- Use voice recordings as a diagnostic tool.
- Leverage machine learning to analyze voice features and predict PD accurately.

Data and Features

Dataset: Parkinson's Data Set from Kaggle.
Includes 22 voice features (e.g., MDVP:Fo(Hz), Shimmer, PPE).
Target: status (0 = Healthy, 1 = Parkinson's).

Key Features:
Measures jitter, shimmer, and harmonic-to-noise ratios, which differ between healthy individuals and PD patients.

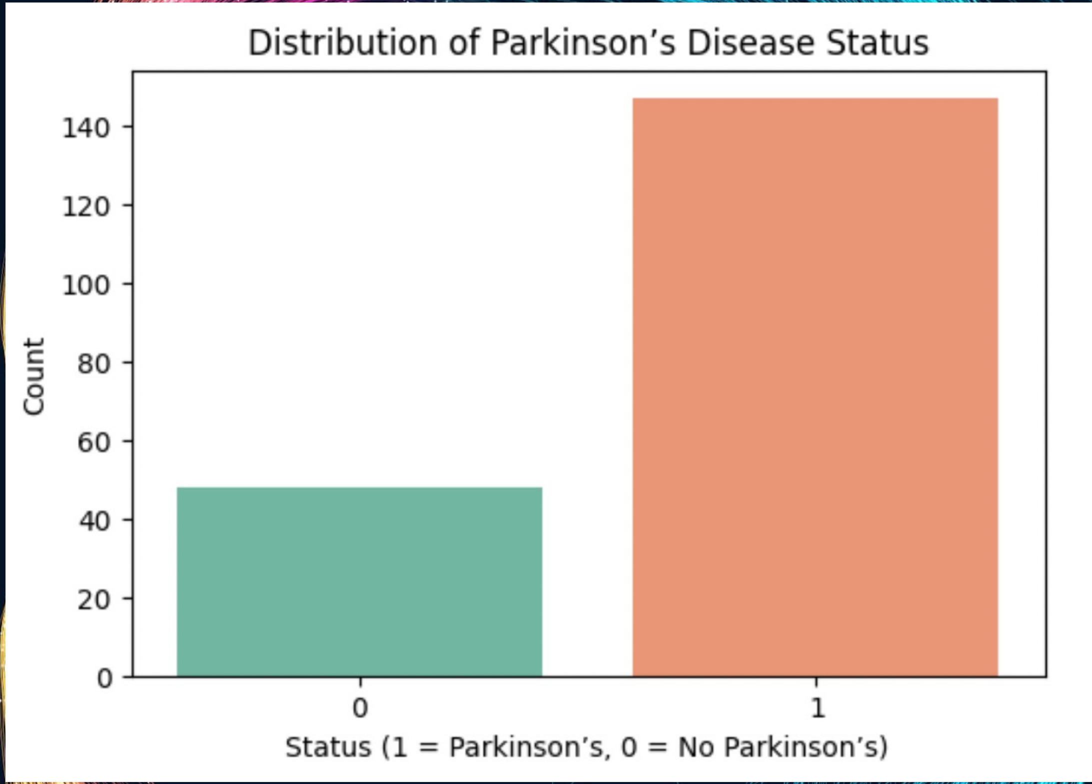
```
parkinsons_data = pd.read_csv('DataSet/parkinsons.data')
parkinsons_data.head()
```

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	...	Shin
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	...	
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	...	
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	...	
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	...	
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	...	

5 rows × 24 columns

Data Preprocessing

Insight: The dataset is imbalanced, with more patients diagnosed with Parkinson's disease (status = 1) than those without it (status = 0).



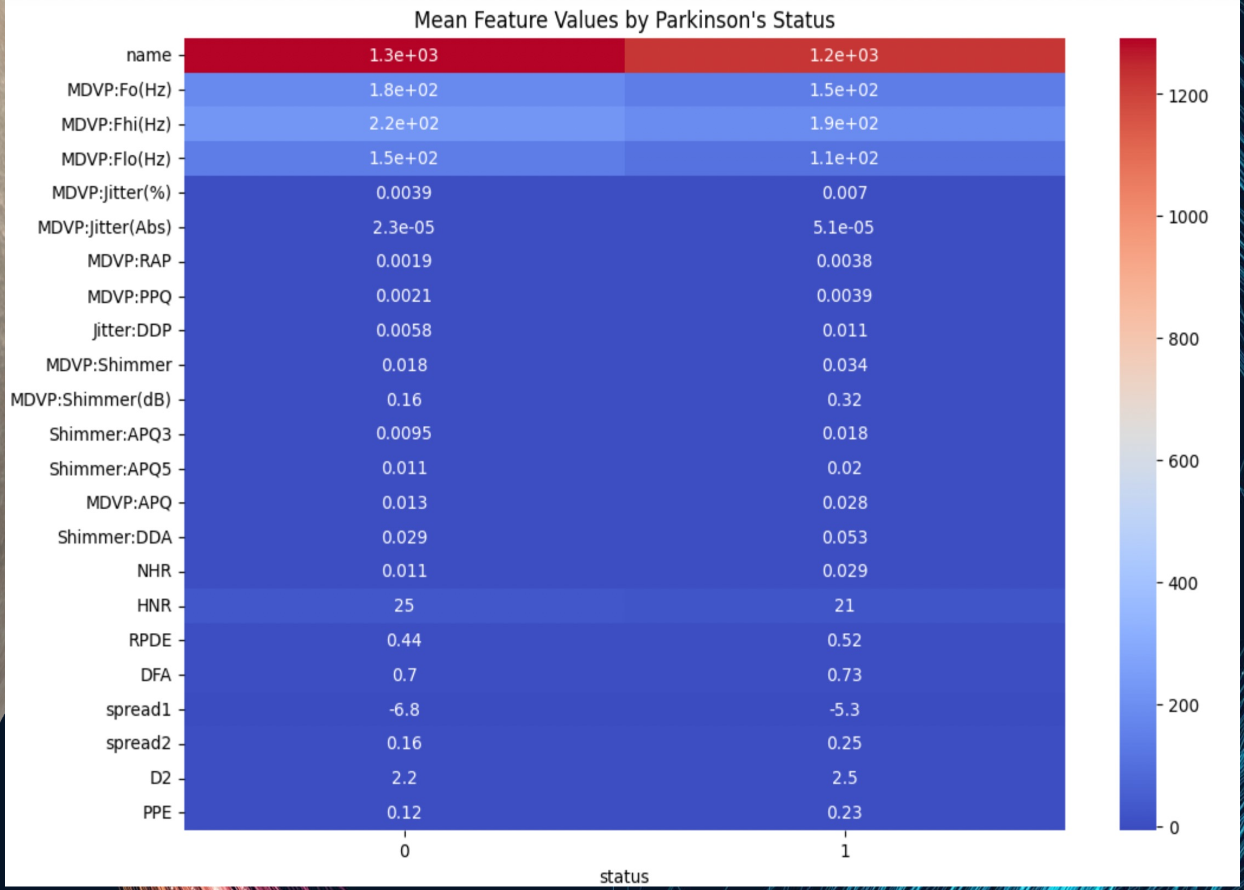
Data Preprocessing

Insight from the Heatmap:
The heatmap above shows the mean values of various features for both Parkinson's patients (status = 1) and non-Parkinson's patients (status = 0).

1. Distinctive Differences: Several features, such as MDVP:Fo(Hz), MDVP:Fhi(Hz), MDVP:Flo(Hz), and PPE, show noticeable differences in mean values between the two groups. This suggests that these features could be significant indicators in differentiating Parkinson's from non-Parkinson's individuals.

2. Higher Jitter and Shimmer Values: Parkinson's patients tend to have higher jitter and shimmer values, indicating more voice irregularities, which are characteristic of the disease.

3. Lower HNR Values: Harmonic-to-noise ratio (HNR) is lower in Parkinson's patients, suggesting that their voices have more noise relative to harmonic components, which can be another distinguishing factor.



Model Development

Model Evaluation and Selection:
Tested **Support Vector Machine (SVM)** with a linear kernel.

Impact of Imbalanced Data on Classification Report
When the target classes are highly imbalanced (e.g., more healthy patients than Parkinson's cases), models often predict the majority class more frequently, leading to misleading performance metrics:
High Accuracy but Poor Sensitivity:
The model correctly identifies most healthy individuals but struggles with Parkinson's cases.

Precision:
Drops for the minority class (Parkinson’s). This means many false positives—predicting Parkinson’s when it’s not.

Recall (Sensitivity):
Significantly low for the minority class, leading to many missed cases of Parkinson’s.

F1-Score:
Harmed due to the imbalance between precision and recall, especially for the minority class.

Accuracy of the model: 87.18%

Classification Report:

	precision	recall	f1-score	support
0	0.71	0.62	0.67	8
1	0.91	0.94	0.92	31
accuracy			0.87	39
macro avg	0.81	0.78	0.79	39
weighted avg	0.87	0.87	0.87	39

Model Evaluation

Tuning the Model with Grid Search

To enhance our **Support Vector Machine (SVM)**, we utilized **GridSearchCV**, testing various combinations of hyperparameters like:

- C: Regularization strength

- gamma: Kernel coefficient

- kernel: Type of kernel (linear, RBF, etc.)

Despite finding the best parameters, the classification report was **not impressive**. Improvements in **F1-score**, **precision**, and **recall** were marginal, showing that tuning alone wasn't enough to overcome the effects of class imbalance.

```python
# Setting up the parameter grid
param_grid = {
    'C': [0.01, 0.1, 1, 10, 100],
    'gamma': [0.001, 0.01, 0.1, 1, 'scale', 'auto'],
    'kernel': ['linear', 'rbf']
}

# Setting up the GridSearchCV
grid_search = GridSearchCV(svm.SVC(), param_grid, scoring='accuracy', cv=5)

# Fitting the GridSearchCV
grid_search.fit(X_train, Y_train)

# Getting the best parameters and best estimator
best_params = grid_search.best_params_
best_model = grid_search.best_estimator_
```

```
Best Parameters: {'C': 100, 'gamma': 0.1, 'kernel': 'rbf'}

Accuracy of the tuned SVM model: 92.31%

Classification Report:
              precision    recall  f1-score   support

           0       0.73      1.00      0.84         8
           1       1.00      0.90      0.95        31

    accuracy                           0.92        39
   macro avg       0.86      0.95      0.90        39
weighted avg       0.94      0.92      0.93        39
```

Model Evaluation

What is SMOTE Doing?

Class imbalance was a significant challenge, so we applied **Synthetic Minority Oversampling Technique (SMOTE)** to tackle it.
SMOTE works by:
- Selecting a minority class sample.

- Finding its nearest neighbors in the feature space.

- Generating synthetic samples between the original sample and its neighbors.

- This method balances the dataset effectively, ensuring the model doesn't favor the majority class.

```python
# Splitting the data into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2, stratify=Y)

# Applying SMOTE to oversample the minority class
smote = SMOTE(random_state=2)
X_train, Y_train = smote.fit_resample(X_train, Y_train)

# Feature scaling
scaler = StandardScaler()

# Fitting the scaler with the training data
scaler.fit(X_train)

# Transforming the data
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

# Building the SVM model
model = svm.SVC( C=100, gamma=0.1, kernel='rbf')

# Training the model with the training data
model.fit(X_train, Y_train)

# Model Evaluation
# Predicting on the test data
Y_pred = model.predict(X_test)

# Perform k-fold cross-validation (k=5)
k = 5
cross_val_scores = cross_val_score(model, X_train, Y_train, cv=k)

# Calculate mean and standard deviation of cross-validation scores
cv_mean = cross_val_scores.mean()
cv_std = cross_val_scores.std()
```

Model Evaluation

Classification Report Results

1. Before SMOTE:

- Precision and recall for the Parkinson’s class were low.

- The F1-score was disappointing, reflecting the model’s struggles to detect true Parkinson’s cases.

2. After SMOTE:

- The classification report showed significant improvements.

- Recall improved, meaning fewer Parkinson’s cases were missed.

- F1-score balanced precision and recall, demonstrating better performance.

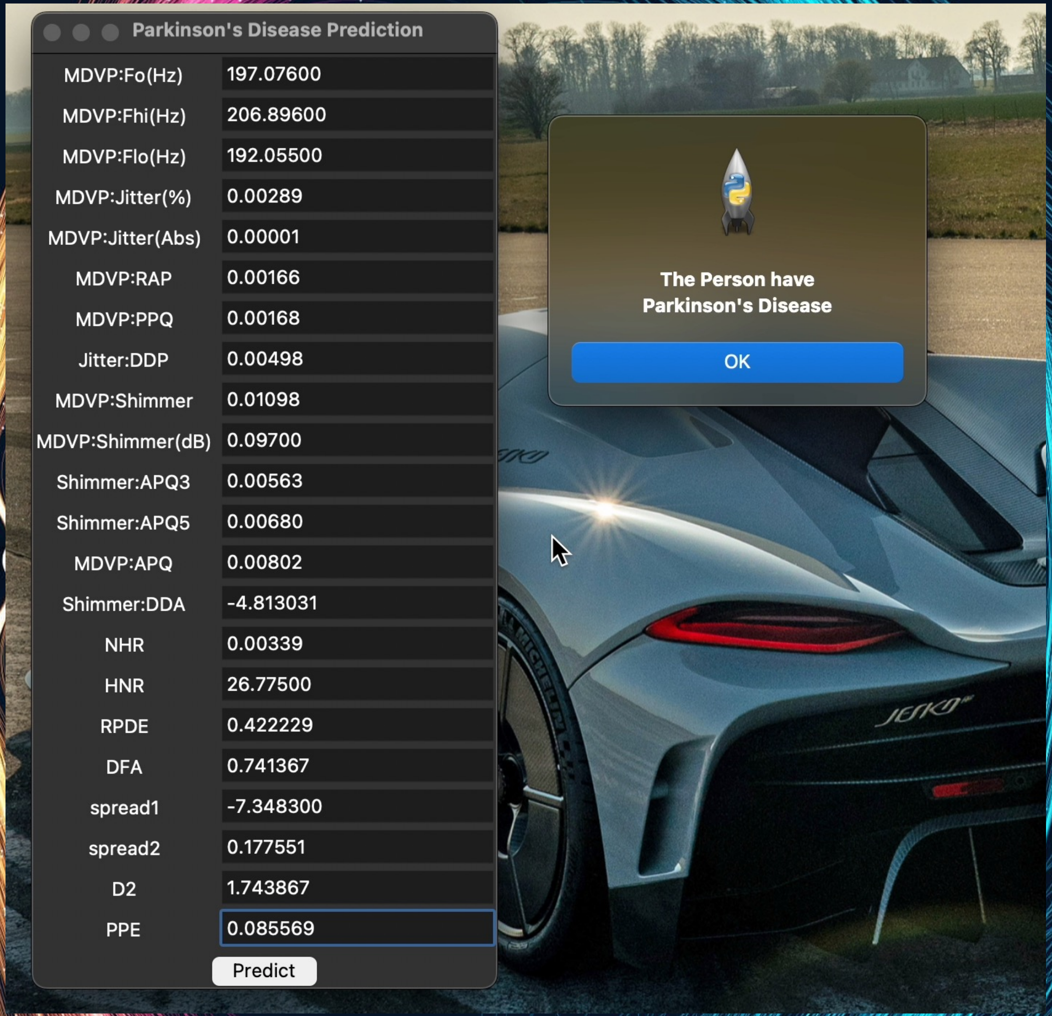
```
Accuracy of the model: 92.31%
Mean Cross-Validation Score (k=5): 96.61%
Standard Deviation of Cross-Validation Scores: 0.0216
Scores per Fold: [0.95833333 0.93617021 0.9787234  0.95744681 1.        ]

Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.80      0.84        10
           1       0.93      0.97      0.95        29

    accuracy                           0.92        39
   macro avg       0.91      0.88      0.90        39
weighted avg       0.92      0.92      0.92        39
```

GUI for Accessibility

Purpose:
Make predictions accessible to non-technical users through an interactive graphical user interface (GUI).

GUI Features:

Input: Users can enter vocal features manually or upload data.
Prediction: The model provides real-time Parkinson's predictions.
Visualization: Displays key metrics like jitter and shimmer.
User-Friendly: Simple design for easy navigation.



Results and Insights

Model Performance:

- Achieved high precision and recall for both classes.

- GUI simplifies interaction, increasing usability in non-clinical settings.

Strengths:

- Non-invasive and cost-effective.

- Easily extensible to other datasets and classifiers.

Thank You