Report—Parkinson's Disease Telemonitoring

Yichun Qu
Electrical and Computer Engineering
10/21/2024
GitHub: https://github.com/quyichun/Brown_Data1030_Yichun.git

A. Introduction

Affecting millions of people worldwide, Parkinson's disease has no definitive cure, making early diagnosis and effective management critical to improving patients' quality of life. Continuous monitoring of symptoms, such as through clinical measures like the UPDRS, is essential for tracking disease progression and tailoring treatment plans to individual needs.

The objective of this regression problem is to predict the total Unified Parkinson's Disease Rating Scale (Total_UPDRS) scores. The dataset consists of 5,875 voice recordings from 42 patients with early-stage Parkinson's disease, collected over a six-month trial using a telemonitoring device for remote symptom monitoring. Each recording includes 16 biomedical voice measurements and patient details such as age, gender, and time since baseline recruitment. The dataset had no missing values. The recordings were automatically captured in the patients' homes. This prediction offers a non-invasive and accessible means to assess disease progression. By leveraging voice data, it aims to improve patient care through early intervention and personalized treatment adjustments, providing a convenient and effective solution for managing Parkinson's disease.

In the study by Tsanas et al. (2010)[1], the authors developed a telemonitoring system to accurately predict the progression of Parkinson's disease using noninvasive speech tests. The results showcased a robust correlation between predicted and actual UPDRS scores, highlighting the potential of speech-based biomarkers as a reliable and cost-effective method for remote monitoring of Parkinson's disease.

B. EDA

To gain insights into the target variable, we visualized the distribution of total_UPDRS scores, enabling a better understanding of its characteristics and supporting decision-making in model development. The histogram Fig. 1 indicates that total_UPDRS scores follow a slightly right-skewed distribution, with the majority of values concentrated between 20 and 40. There are

fewer instances of extremely low or high scores, which may impact model predictions for outlier cases.

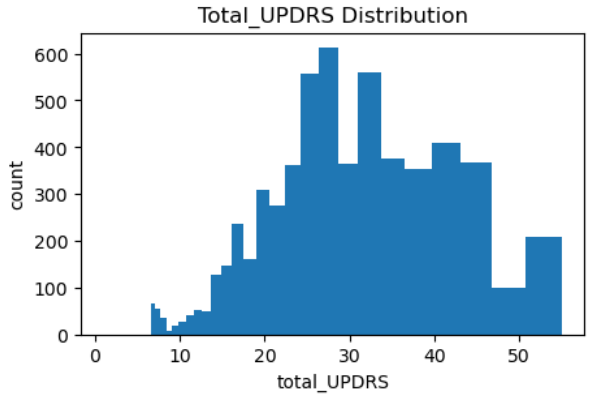

Fig.1. Distribution of the target variable "Total_UPDRS".

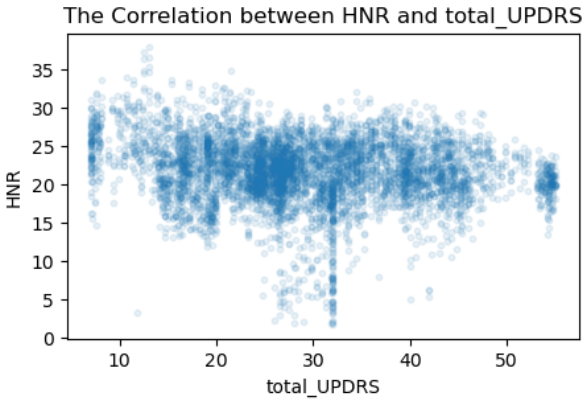

Fig.3. Relationship Between HNR and Total_UPDRS.

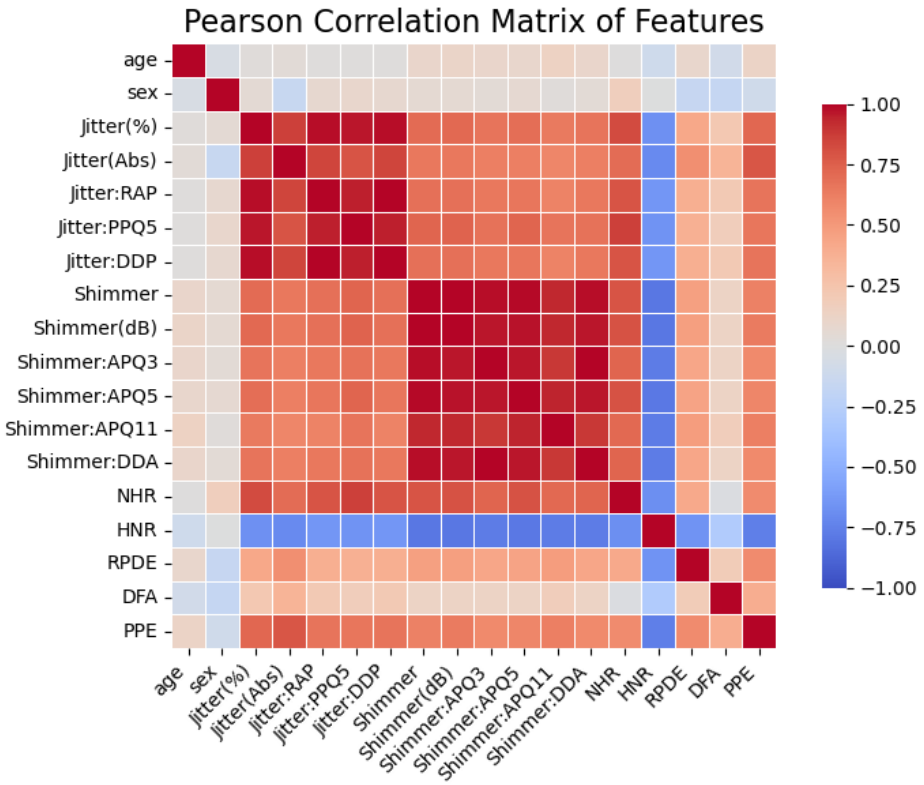

Fig.2. Pearson Correlation Matrix of Features.

Additionally, a Pearson correlation matrix was generated to analyze relationships among features. The heatmap Fig. 2 reveals strong correlations among several features, particularly

within the Jitter and Shimmer groups. These correlations suggest that the Jitter and Shimmer groups features could be dropped all but one to reduce data redundancy.

The relationship between the target variable and the ‘HNR’ feature was explored through visualization in Fig.3. The scatter plot shows a special trend between HNR and total_UPDRS, suggesting that HNR remains almost the same range as the Total_UPDRS scores increases. However, there is considerable dispersion, indicating potential non-linear relationships.

C. Methods

The dataset is split into training, validation, and test sets using group-based splitting to ensure that all recordings from the same patient remain in a single set (training/validation/test), preventing data leakage. Initially, the data is divided into a test set (20%) and an "other" set using GroupShuffleSplit. Subsequently, the "other" set is further split into training (60%) and validation (20%) sets using GroupKFold with 4-fold cross-validation, ensuring that patient groups remain distinct throughout.

The data preprocessing involves two main steps: One-Hot Encoding and Standardization. One-Hot Encoding is applied to the categorical feature 'sex' to convert it into binary columns representing each category (male or female). Standardization is then applied to continuous features such as 'age,' 'test_time,' and the biomedical voice measures to ensure that all features have a mean of 0 and a standard deviation of 1.

The machine learning pipeline begins with splitting the dataset into training and test sets, followed by cross-validation on the training set to ensure robust performance estimation. The pipeline integrates a preprocessor for feature transformation and a machine learning algorithm. The best model identified through GridSearchCV is then evaluated on the test set using Root Mean Squared Error (RMSE). RMSE is chosen as the performance evaluation metric because it provides an intuitive measure of the average error magnitude between predicted and actual values. To ensure robustness, the entire process, from data splitting to evaluation, is repeated 10 times with different random states. Results for each iteration, including test set RMSE scores, cross-validation RMSE scores, and the best models and parameters, are systematically recorded.

To solve the problem, I explored five machine learning algorithms: Linear Regression, Random Forest (RF), Support Vector Machine (SVM), XGBoost, and K-nearest neighbor classification (KNN). The machine learning models and parameters to tune are summarized in Table 1.

Table 1. Summary of Machine Learning Models and Tuned Parameters.

Models	Parameters Tuning
Linear Regression	None
Random Forest	max_depth: [5, 10, 15, 20]
Support Vector Machine	C: [0.1, 1, 10] epsilon: [0.1, 0.2, 0.5] kernel: 'rbf'
XGBoost	max_depth: [3, 5, 7, 9] learning_rate: [0.001, 0.05, 0.1, 0.2]
KNN	n_neighbors: [90, 100, 200, 300, 400] weights: ['uniform', 'distance']

Table of models and hyperparameter tuning ranges used for training and evaluation.

Each model was evaluated using RMSE, and uncertainties were measured by repeating the train-test split and model evaluation 10 times with different random seeds, accounting for the variability due to data splitting and non-deterministic methods like Random Forest. These steps were carefully designed to prevent overfitting, maintain model interpretability, and ensure reliable evaluation of each algorithm's performance.

D. Results

The test scores of all the machine learning models are summarized in Table 2. The SVM model was the most predictive, achieving the lowest RMSE of 4.0347, indicating its superior performance in predicting the target variable. Linear Regression followed closely with an RMSE of 4.1649, demonstrating strong predictive capability despite being a simpler model.

Table 2. Performance Comparison of Machine Learning Models

Models	Best Parameters	Mean RMSE	Standard Deviation	SD above Baseline
Linear Regression	None	0.5238	4.1649	6.5009
Random Forest	max_depth: 10	0.1675	6.2902	4.3833
Support Vector Machine	C: 0.1 epsilon: 0.5	0.2755	4.0347	6.6306
XGBoost	max_depth: 3 learning_rate: 0.05	0.9808	5.7218	4.9496
KNN	n_neighbors: 200 weights: uniform	0.4592	4.5218	6.1453

The table includes the mean RMSE, standard deviation, and the number of standard deviations above the baseline RMSE.

The baseline RMSE, calculated by using the mean of the target variable as the predictor, serves as a point of reference. The baseline RMSE (10.6994) is significantly higher than the RMSE scores of all the tested models, as demonstrated in the comparison Fig.4 and Fig.5. The models exhibit marked improvements, with RMSE values lower than the baseline. In terms of relative improvement, SVM and Linear Regression achieved the largest percentage reductions in RMSE compared to the baseline. These results demonstrate that the selected models significantly outperform the baseline, indicating robust and reliable predictions.

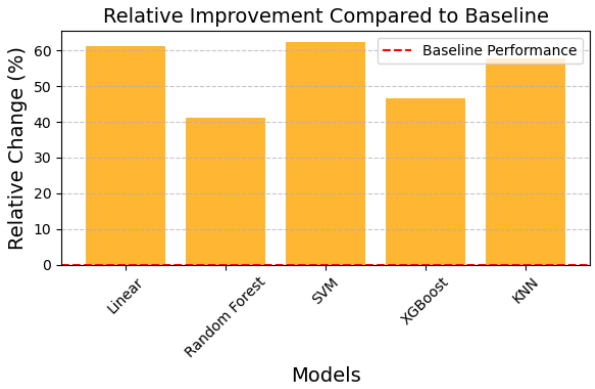

Fig.4. The relative improvement in performance of various models compared to the baseline, with Linear and SVM models achieving the highest percentage increases.

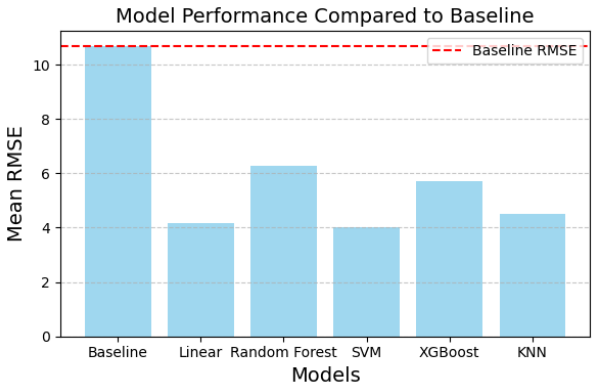

Fig.5. The mean RMSE of different models compared to the baseline, highlighting a significant reduction in RMSE across all models.

Global feature importance was evaluated using three metrics from the XGBoost model: Cover, weight, and Total_cover. The results, shown in Fig. 6, 7 and Fig. 8, reveal key patterns in feature importance, shedding light on which variables are driving model predictions. For Cover, the feature subject# emerged as the most important globally, indicating it contributes most frequently to improving splits in the model. When considering Total_cover, which accounts for the cumulative quality of splits, Shimmer(dB) stood out as the dominant feature. Other features, such as motor_UPDRS and PPE, also demonstrated high importance under this metric. Notably, while subject# showed high importance under Cover, it was less prominent in Total_cover, suggesting its contributions might be more about structural splits than improving predictive accuracy. ‘PPE’ tend to be in the top 5 regardless of the approach used.

Local feature importance was evaluated using SHAP values for example data points (Index 32 and Index 1131). The SHAP analysis, as Fig. 9 and Fig. 10 shown, provides insights into how individual features influenced the model's predictions for these data points. For Index 32, subject# and motor_UPDRS contributed the most to lowering the predicted value, while age, sex, and HNR had marginal positive influences. In contrast, for Index 1131, motor_UPDRS

significantly increased the predicted value, while age acted as a moderating factor, reducing the prediction. The consistent significance of motor_UPDRS across both data points underscores its pivotal role in the model.

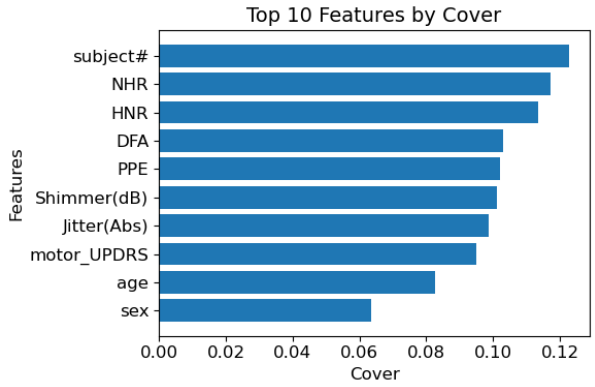

Fig. 6. Top 10 features ranked by **Cover** from the XGBoost model, showing the frequency with which features contribute to improving splits during training.

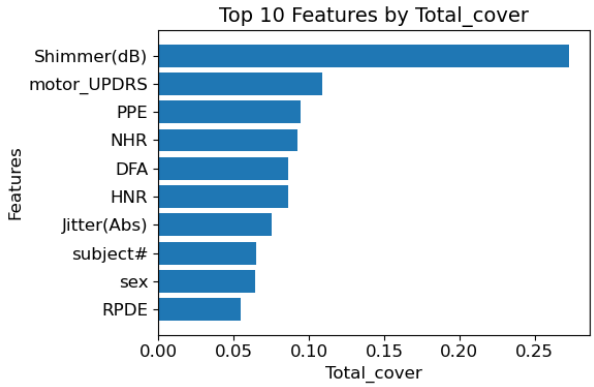

Fig. 7. Top 10 features ranked by **Total_Cover** from the XGBoost model.

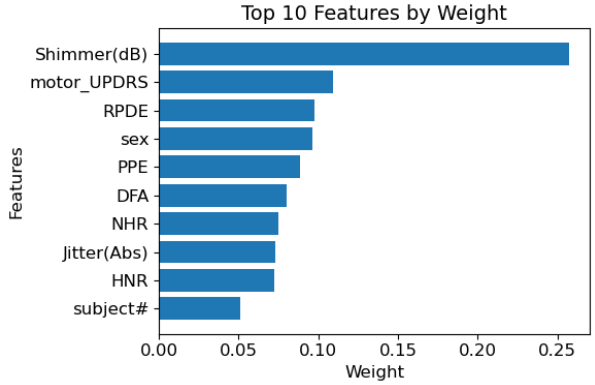

Fig. 8. Top 10 features ranked by **weight** from the XGBoost model.

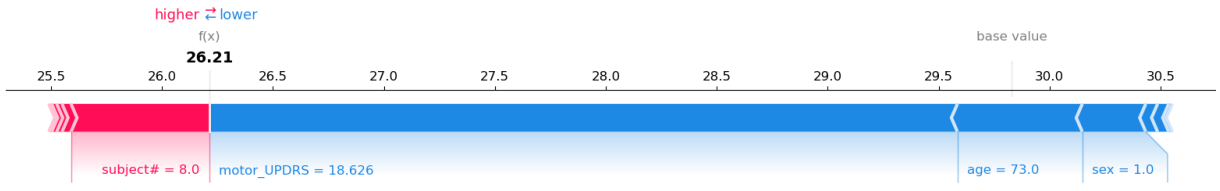

Fig. 9. SHAP Values of Data Point Index 32.

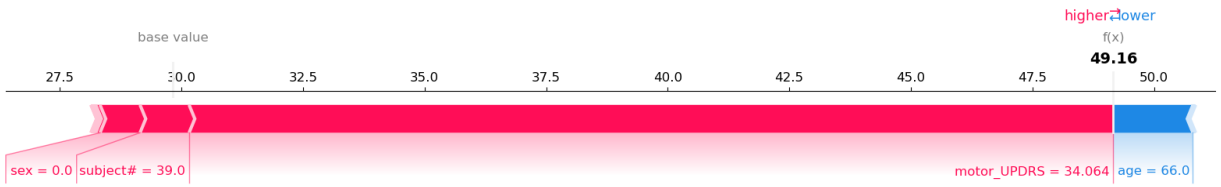

Fig. 10. SHAP Values of Data Point Index 1131.

Interestingly, the analysis revealed some unexpected insights. Despite its relatively low global importance under Total_cover, the demographic feature sex had noticeable local effects in specific cases. This suggests it interacts with other features to influence predictions, even if its overall contribution is minor. Similarly, while subject# ranked highly in global Cover metrics, its local effects were less pronounced for some instances, raising questions about its utility in broader predictive contexts versus individual cases.

E. Outlook

Linear Regression, while interpretable and computationally efficient, has limited flexibility in capturing complex or nonlinear relationships within the data. While feature engineering addressed some of these limitations, the overall model performance is still constrained by the linearity assumption. Specifically, Pearson Correlation (Scores are summarized in Table 3) was used for feature selection.

Table 3. Pearson Correlation Scores of Features.

Feature	**Correlation Coefficient**
motor_UPDRS	0.927768
subject#	0.678061
age	0.426433
Shimmer(dB)	0.314391
RPDE	0.310398
PPE	0.226326
Jitter(Abs)	0.203059
NHR	0.159914
DFA	-0.082357
sex	-0.255210
HNR	-0.293769

Table showing the correlation coefficients between features and the target variable. motor_UPDRS exhibits the strongest positive correlation (0.9278).

Interaction features were generated using PolynomialFeatures with a degree of 2. By creating interaction terms such as the product of age and motor_UPDRS, the model was able to account for subtle relationships between features that contribute significantly to the target prediction.

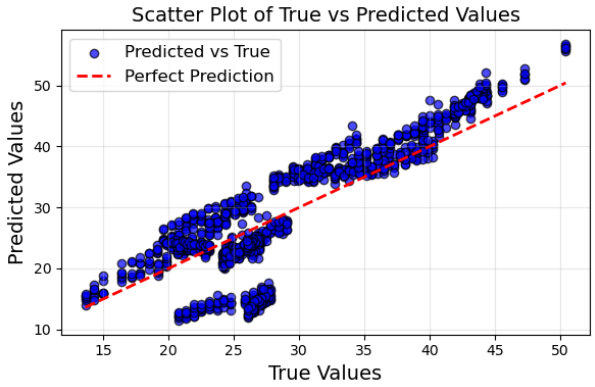

Fig.11. Scatter plot of true versus predicted values before feature engineering.

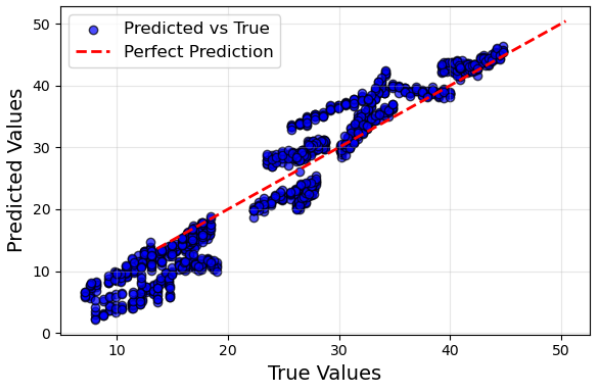

Fig.12. Scatter plot of true versus predicted values after feature engineering.

From the visual analysis, it is evident that the feature engineering process led to a noticeable improvement in model performance. In Fig.11, the predictions exhibit greater scatter around the perfect prediction line, indicating that the model struggled to capture the underlying relationships, particularly at higher values. In contrast, Fig.12 shows significantly reduced scatter. The improved alignment indicates that the model better captures the relationships between the selected features and the target variable, reducing errors and enhancing overall predictive accuracy.

Another significant area for improvement lies in the modeling techniques used. Experimenting with more sophisticated models like LightGBM or CatBoost, which handle categorical features efficiently, may yield better performance without extensive preprocessing. From a data perspective, additional features could be collected to improve model performance. For instance, domain-specific data such as clinical history, genetic markers, or environmental factors could provide more context and improve the model's ability to generalize.

F. Reference

[1] Tsanas A, Little MA, McSharry PE, Ramig LO. Accurate telemonitoring of Parkinson's disease progression by noninvasive speech tests. IEEE Trans Biomed Eng. 2010 Apr;57(4):884-93. doi: 10.1109/TBME.2009.2036000.

[2] Little MA, McSharry PE, Hunter EJ, Spielman J, Ramig LO. Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. IEEE Transactions on Biomedical Engineering. 2009 Apr;56(4):1015-22. doi: 10.1109/TBME.2008.2005954.

[2] Orozco, Juan Rafael & Noeth, Elmar & Vargas-Bonilla, J. & Arias-Londoño, Julian D.. (2013). Analysis of Speech from People with Parkinson's Disease through Nonlinear Dynamics. Lecture Notes in Artificial Intelligence. 7911. 112-119. 10.1007/978-3-642-38847-7_15.

[4] Chiaramonte R, Bonfiglio M. Acoustic analysis of voice in Parkinson's disease: a systematic review of voice disability and meta-analysis of studies. Rev Neurol. 2020 Jun 1;70(11):393-405. Spanish, English. doi: 10.33588/rn.7011.2019414. PMID: 32436206.