Predicting Parkinson's Disease Progression

Xiru Wei
Brown University Sociology Department
https://github.com/xiruwei/data1030projectparkinson.git

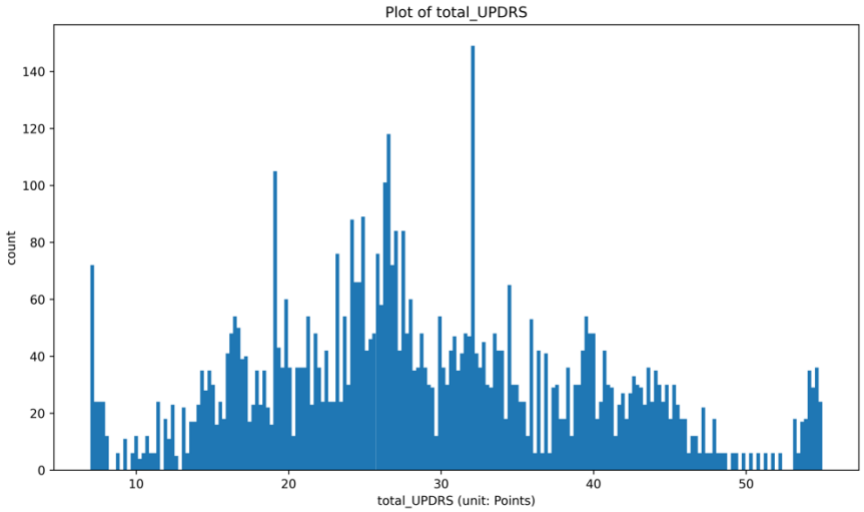
1. Introduction

1.1 Problem & Significance

Parkinson's disease affects 1 million people in North America. The current monitoring technique for the progression of Parkinson's disease requires the patients to visit specialists and undergo hours of physical and questionnaire-based examinations to attain a Total Unified Parkinson's Disease Rating Scale. This process is very burdensome and can be cumulatively expensive for patients since Parkinson's disease is a long-term disease.

Since about 70 to 90% of patients with Parkinson's disease have vocal impairment, if an accurate method linking voice recording features to the disease progression can be developed, telemonitoring Parkinson's Disease would be made possible. Telemonitoring Parkinson's Disease through patient's voices can provide a unique opportunity to elevate the burden for the patients and provide a timelier monitoring of the progression of the disease.

This report specifically investigates how to use speech signal features captured from voice recordings to calculate patients' total Parkinson's disease rating scale accurately. This problem is a regression-based problem since the target variable is the Total Unified Parkinson's Disease Rating Scale, which is continuous, as shown by Graph 1.



Graph1: Plot of Target Variable: total_UPDRS

1.2 Target Variable and Features

The data set I used comes from Tsanas et.al (2010), which I accessed through the UCI machine learning repository. The original clinical trial collected the voice recordings of the patients through a special voice recording machine *Intel AHTD.*

This raw data set contains 5875 rows and 22 features. The target variable in this data set is called "total_UPDRS" (Total Unified Parkinson's Disease Rating Scale), whose value ranges from 5 to 56, with a higher value meaning more severe disease progression.

The dataset is not IID there are in total 42 unique patients identifiable through their Subject#. Each patient has multiple rows of inputs containing their speech signal features and a corresponding Total Unified Parkinson's Disease Rating Scale score.

1.3 Previous Work

This data set have been used before. In Tsanas et.al's (2010) research, the best performing model in the research was the Classification and Regression Trees (CART) method. It outperformed the linear predictors (Least Squares (LS), Iteratively Reweighted LS (IRLS), and Least Absolute Shrinkage and Selection Operator (LASSO)) in terms of Mean Absolute Error (MAE). For the training data, CART had an MAE of 4.5 and for the testing data, it had an MAE of 5.8. This is in comparison to the LS and IRLS which had a testing MAE of 6.7 and LASSO with 6.8. Furthermore, for the training data regarding the linearly interpolated total UPDRS, CART produced an MAE of 6.0 and for the testing data, an MAE of 7.5, which was again better than the other models tested.

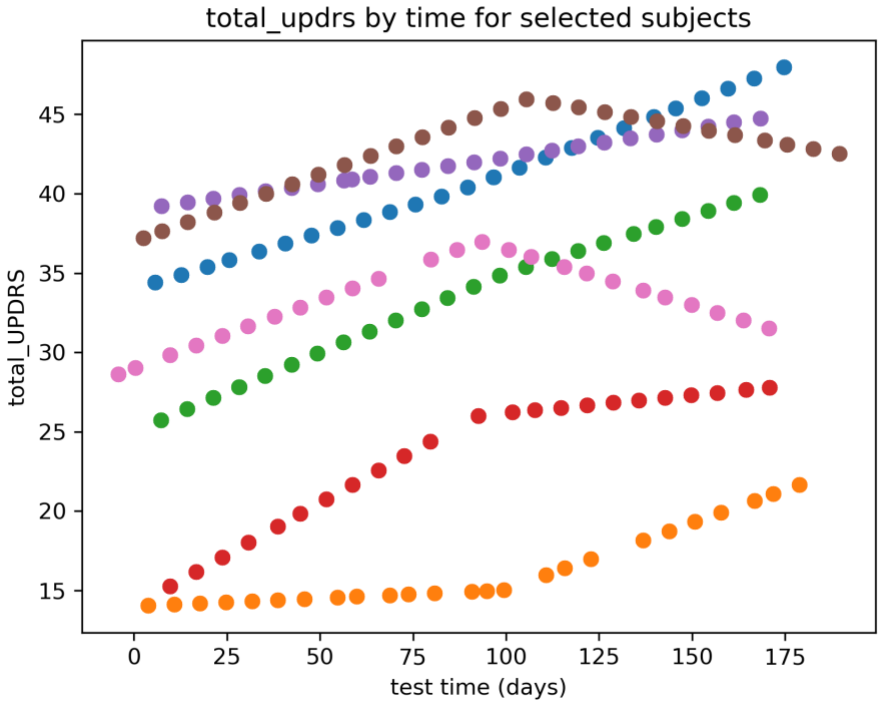
2. EDA

2.1 Missing Values

There are no missing values in the Dataset I use.

2.2 Target Variable

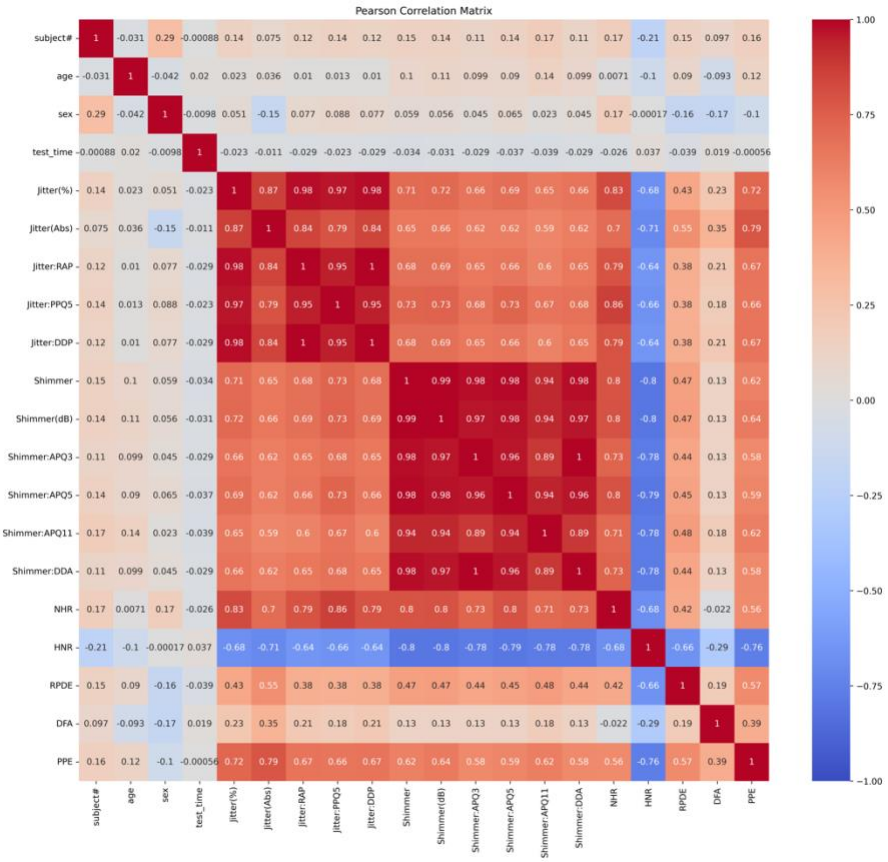
It is important to note that not all 5875 records of the Total UPDRS are physically measured in the clinics. For every subject, the scientists running the clinical trials only measured their UPDRS scores at the beginning, middle, and the end of the trial. The rest of the measurements are all imputed linearly based on test time. This imputation method is widely adopted in Parkinson's disease research. This relationship is shown in Graph 2. In implementation, the hope is that when inputting the patient's voice recording features, there will be an output of the total_UPDRS score. Therefore, I decided to drop "test_time" because it will not be something that should provide as much information in real implementation.



Graph 2: Total_UPDRS over test_time by Subjects

2.3 Features

As seen in Graph 3, many of the features in the dataset are very highly correlated. I decided to drop one feature from the two pairs of perfectly correlated features. Therefore, 'Jitter:RAP' and 'Shimmer: APQ3' are dropped.



Graph 3: Feature Correlation Plot

3. Methods

3.1 Splitting Strategy

Since the dataset is not IID, I split based on Subject#. I firstly set aside 20% of the dataset as test set, through Group Shuffle Split method, with Subject# as group ID. For the remaining 80% of the dataset, I employed GroupKFold to split in a way that 60% of the total dataset is in Train and 20% is in Validation set, and there are 4 folds. I looped through 10 random states to ensure the results are not caused by randomness in the Data splitting.

3.2 Data Processing

There are in total 22 variables in the data set. I set aside "total_UPDRS" as the target variable. In the rest of the 21 variables. 'Jitter:RAP', 'Shimmer: APQ3', and "test_time" are dropped for reasons described above. In addition, I dropped "Subject#" after splitting. "motor_UPDRS" is also dropped since it is a variable very similar to the total_UPDRS score and is not relevant to the purpose of this project, which is using voice recording features to predict disease progression.

Thus, after dropping these features, there are 16 features left. In these 16 features, only one is categorical: "sex." Since sex is already coded as 0 for female and 1 for male, there is no need to preprocess it. For the other 15 features, I applied MinMax feature transformation for "age" and StandardScaler for the other 14 continuous features.

Model	Hyperparameters
Ridge	ridge__alpha: [0.1, 1, 5, 10]
	ridge__max_iter: [1000]
Lasso	lasso__alpha: [0.1,1, 5, 10]
	lasso__max_iter: [1000]
KNNeigbor	kneighborsregressor__n_neighbors: [3, 5, 10,250,280,290,500,1000]
	kneighborsregressor__weights: [uniform]
	kneighborsregressor__p: [1, 2, 3]
XGBoost	xgbregressor__n_estimators: [1, 27, 100, 200],
	xgbregressor__max_depth: [1, 3, 5, 7, 10, 30],
	xgbregressor__learning_rate: [0.1 0.2]

Table 1: Model name and Hyperparameters

3.3 ML Pipelines & Hyperparameters

I trained 4 models on my data set, including linear regression with Lasso regularization, linear regression with Ridge regularization, KNN regression and XGB regression. I used GridSearch SV to tune hyperparameters, as summarized in Table 1.

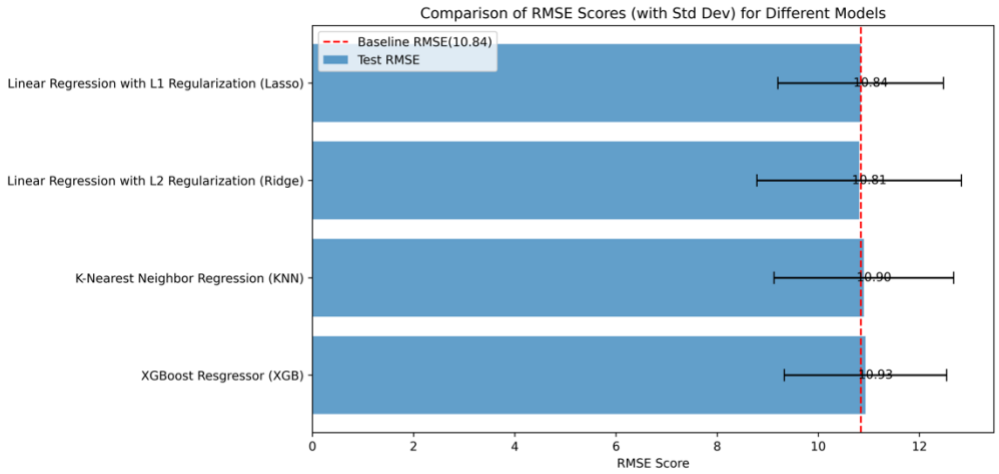
3.4 Evaluation Metrics

The evaluation metric I used is RMSE, which is the square root of the average of the squares of the errors. The error is the difference between the predicted values by the model and the actual values. RMSE gives a measure of the magnitude of the error made by the predictive model. A lower RMSE indicates a better fit to the data, meaning the model's predictions are closer to the actual values. It is commonly used in regression analysis.

4. Results:

4.1 RMSE Scores Compared to Baseline

As shown in Table 2 and Graph 4, the four models performed very similarly to the baseline model. XGBoost Regressor (XGB) has an RMSE score around 10.93 with a small standard deviation. K-Nearest Neighbors Regression (KNN) - RMSE just below 10.90, with a standard deviation slightly larger than the XGB. Ridge Regression (Ridge) - RMSE around 10.81 with a standard deviation comparable to the KNN. Lasso Regression (Lasso) - RMSE score is right at the baseline of 10.84, with a standard deviation that suggests some variability but is still relatively tight.



Graph 4: Summary of RMSE Scores of different models

Model	Mean Test RMSE	Std Dev
Lasso	10.84	1.64
Ridge	10.81	2.02
KNN	10.9	1.79
XGBoost	10.93	1.6
Baseline	10.84	1.64

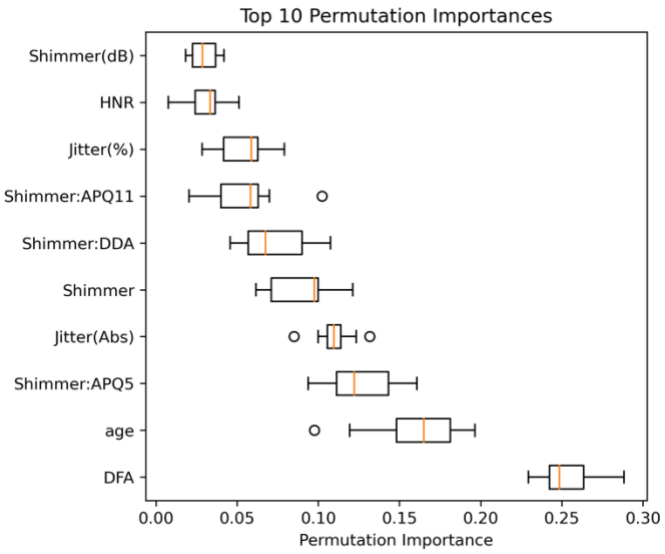
Table 2: Model Names, Mean Test RMSE, and Standard Deviation

4.2 Global & local Feature Importance

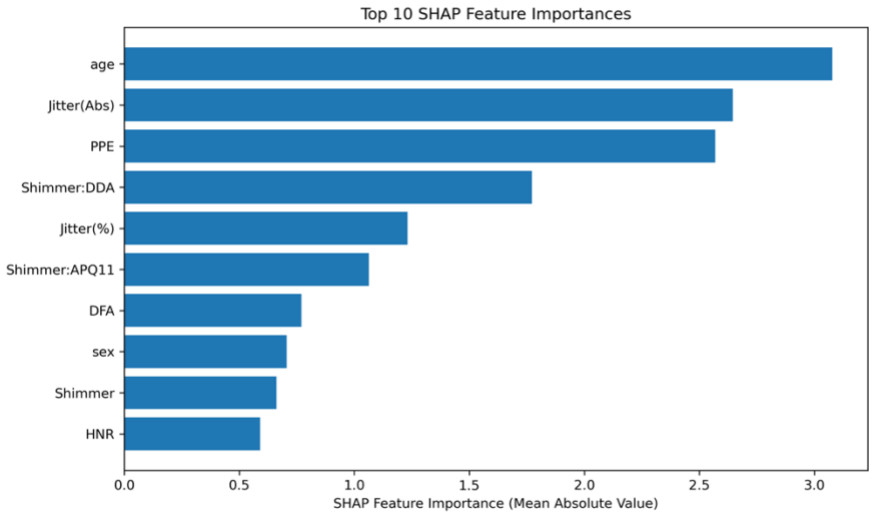
My models all performed similarly. However, for the purpose of interpretability, I chose Ridge Regression (Ridge) to calculate the global and local importance features. For global feature importance metrics, I calculated three metrics, including Permutation importance, SHAP Feature Importance, and Ridge Regression Coefficients (scaled). As shown in Graph 5, 6, 7, the features that emerged as most important for Permutation Importance metrics are “DFA”, “Age”, and “Shimmer: APQ5”; the features most important based on SHAP are “Age”, “Jitter(Abs)”, “PPE”; the features emerged as most important based of scaled Ridge coefficients are : “Jitter(Abs)”, “Jitter(%)”, and “PPE”. Across all three metrics, I found Age, Shimmer DDA, DFA, and HNR are the 4 features that appeared all three times in the top 10 of each of these metrics. (Summary shown in Table 3)

Feature	How many times it appeared in the top 10 of the three global feature importance metrics
age	**3**
shimmer dda	**3**
Dfa: detrended fluctuation analysis	**3**
Hnr: noise to harmonic ratio	**3**
ppe	2
jitter%	2
shimmer apq11	2
shimmer	2
jitter abs	2
shimmer apq5	2
jitter	1
sex	1
shimmerapq11	1
shimmer	1
jitter %	1
shimmer(db)	1

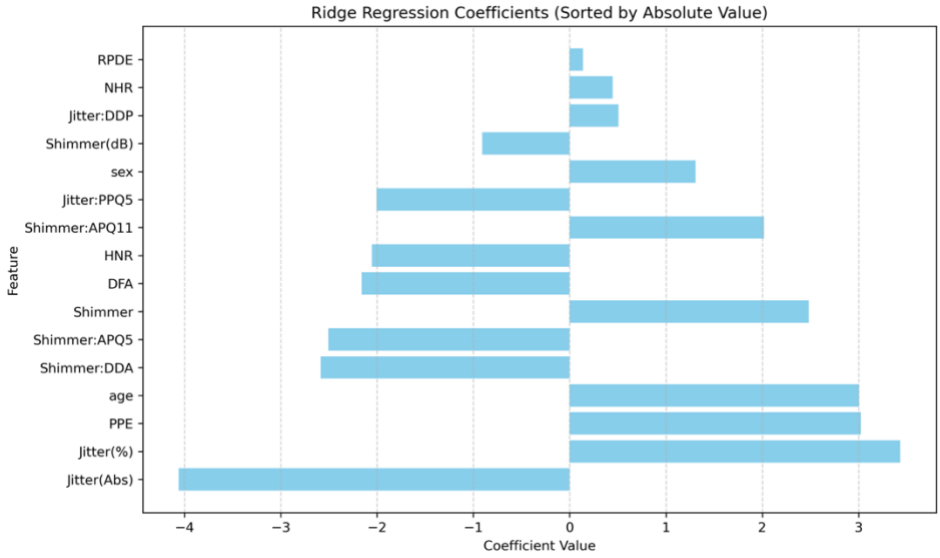
Table 3: Most important features by how many times they appeared in global feature importance metrics



Graph 5: Top 10 Permutation Importance Graph

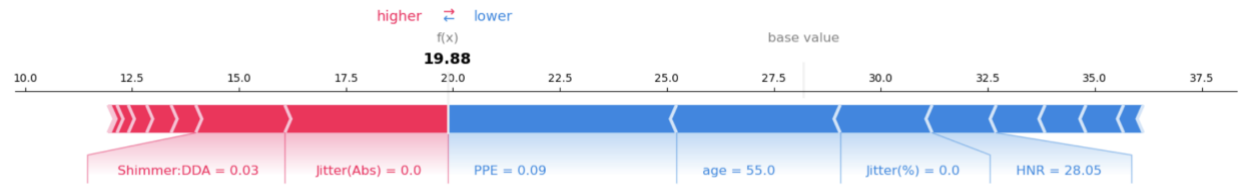


Graph 6: Top 10 SHAP Importance Graph

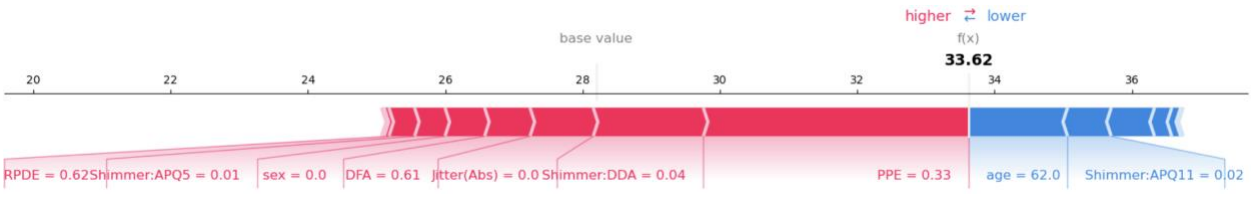


Graph 7: Standardized Ridge Regression Coefficient, sort by Absolute Values

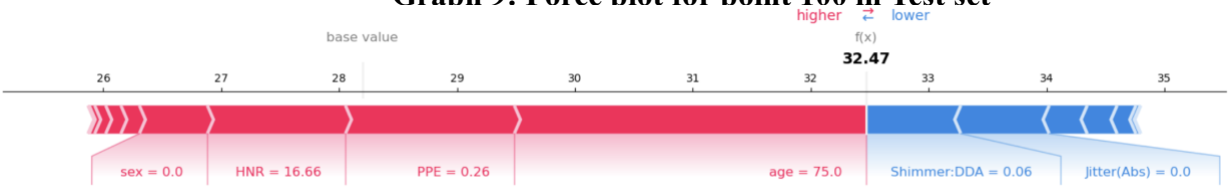
I also calculated the force plot for three points in my test set, as shown in Graph 8-10. The most important features for the three points mirrors some of the most important features discussed in Global & local feature importance, such as age, shimmer DDA and DFA etc.



Graph 8: Force plot for point 0 in Test set



Graph 9: Force plot for point 100 in Test set



Graph 10: Force plot for point 200 in Test set

The result is very intuitive since Parkinson's disease usually is not reversible, and the older the patients the more likely their disease is progressed (Tsanas, 2010)

5. Outlook

5.1 Improvement on the Current Model

Many features in my data set are highly correlated. I did not really employ any methods to address this issue, besides dropping some that are perfectly correlated. If I have more expertise and time, I would conduct more feature engineering and see if there is a way to improve the model performance.

5.2 Further Directions

Another direction I could take is to examine if the model's predictive performance for total_UPDRS differs based on whether the data point is imputed or actual measurements. Although it is common practice to assume that the total_UPDRS develop linearly in short time frames, if I have more data inputs that are actual measurement and a variable clearly indicating if the data point is imputed or actual measurement, I think an analysis on whether the model performance differ would be informative for implementation.

References:

Dataset: Tsanas,Athanasios and Little,Max. (2009). Parkinsons Telemonitoring. UCI Machine Learning Repository. https://doi.org/10.24432/C5ZS3N.

Tsanas, M. A. Little, P. E. McSharry, and L. O. Ramig, "Accurate Telemonitoring of Parkinson's Disease Progression by Noninvasive Speech Tests," in IEEE Transactions on Biomedical Engineering, vol. 57, no. 4, April 2010.