AMP-Parkinson's Disease Progression Prediction

Using protein and peptide data measurements from Parkinson's Disease patients to predict progression of the disease

George Mason University

Department of College of Engineering and Computing

Fraol Abebe, Mitch Breeden, John Gullette, Jason Pinto, Ryan Wadsworth

Table of Contents

1. Introduction

Parkinson's disease is a neurodegenerative disorder that primarily affects the central nervous system. It is estimated that around 90,000 individuals in the United States are diagnosed with Parkinson's disease each year, according to the Parkinson's Foundation (Statistics | Parkinson's Foundation, n.d.). The focus of this report is to predict the progression of Parkinson's disease using a measurement called MDS-UPDR (Movement Disorder Society-Unified Parkinson's Disease Rating) and identifying factors associated with the progression of parkinson's disease using protein abundance data.

2. Sources of the datasets

The data sets come from the Accelerating Medicines Partnership Parkinson's disease (AMP-PD) program. This program is a partnership between the NIH and several private organizations to study Parkinson's disease. The data comes from anonymized patient doctors visits where they measure the protein and peptide levels as well as the advancement of parkinson's disease symptoms. AMP-PD published the data to Kaggle as part of a competition on the platform.

3. Data Description: Predictors and Response Variables

The predictors are over 1,200 different proteins and peptides abundance levels, as well as the patient ID, visit ID, medication usage information (which was a binary value), and the number of months since the patient's first visit. The response variables are Unified Parkinson's Disease Rating Scale (UPDRS) scores, which are comprehensive assessments of both motor and non-motor symptoms associated with Parkinson's. For this data, there were 4 different UPDRS scores: UPDRS_1, UPDRS_2, UPDRS_3, and UPDRS_4. The exact symptom measured by each response is unknown.

The data originally came in 3 different files: peptides, proteins, and clinical data. The proteins file contained all of the protein data for each patient's visit; the peptides file contained all of the peptides data for each patient's visit; and the clinical data contained information regarding the patient's UPDRS scores and the binary result of their medication usage.

4. Data Pre-processing: Transformation of Raw Data for Centering, Scaling and Removal of Near Zero Variance for Predictive Modeling

In order to be prepared for modeling, the data from the 3 different files needed to first be combined. At first, the protein and peptide files contained a separate row for each protein and peptide count from each visit. Therefore, we performed a pivot on the visit ID for each of those two tables in order to turn all of the different rows into columns for each visit. Then, those two pivot tables were merged with the clinical data based on the visit ID. This created a single dataframe where each row was a separate clinical visit and each row had 1,203 columns, one for each protein and peptide count as well as the columns for the clinical visit information.

After creating a single merged dataframe for the data, we then performed significant data cleaning.

The first issue we discovered was the prevalence of null values in the protein and peptide data. It turned out that not every clinical visit was properly measured for every protein and peptide. In total, some protein and peptide columns had upwards of 200 null values out of the 713 total rows. Most of the columns and most of the rows had significant null values, therefore we decided to impute the null values rather than eliminate rows and columns. For imputation, we used a K-nearest neighbors algorithm to determine the 3 rows that were most similar to the one with missing values and impute the values based on the average of those rows.

In addition to the null values in the protein and peptides data, we also found that there were significant numbers of null values in the UPDRS-4 scores. This was more problematic because the UPDRS-4 scores are one of the values we were trying to predict. In total, roughly half of the UPDRS-4 scores were missing. Because the direct request from the AMP-PD was to predict all four of the UPDRS scores, instead of not training a model on UPDRS-4, we decided to impute the values using the same K Nearest Neighbors algorithm as for the proteins and peptides.

After we imputed the missing values, we prepared the data by creating two different training and test sets for two different approaches. The first set was for training a model strictly off protein and peptide data. Therefore, the data was randomly split 70/30 into training and test sets. The second set, however, took advantage of the time series aspect of the data. Each patient's visit was timed with a number of months since the patient's first visit. Therefore that data was split 80/20 where all of the training data was each patient's first 80% of their visits and the test data was the last 20% of their visits. Additionally, the time series data included columns for the UPDRS scores from the patient's first initial visit as predictors; essentially establishing a baseline for the UPDRS scores afterwards.

Each training set was then used to train a standard scaler, which would center and scale the data. This was then applied to each test data set. Following centering and scaling, we then performed Principal Component Analysis for dimensionality reduction. Dimensionality Reduction was seen as a crucial step to produce a more efficient model, considering that the data originally had 1,200 columns.

For the just protein and peptide models, we found that around 400 components explained 99% of the variance in the data, therefore we used 400 components.

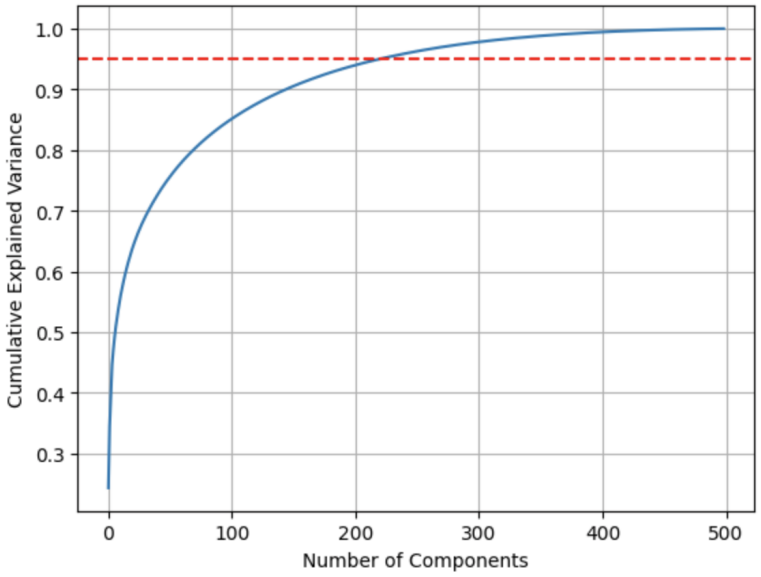

Figure 1: The scree plot for the protein and peptide data alone.

For the time series split data, we also found that 400 components explained 99% of the variance in the data, so therefore we also used 400 components.

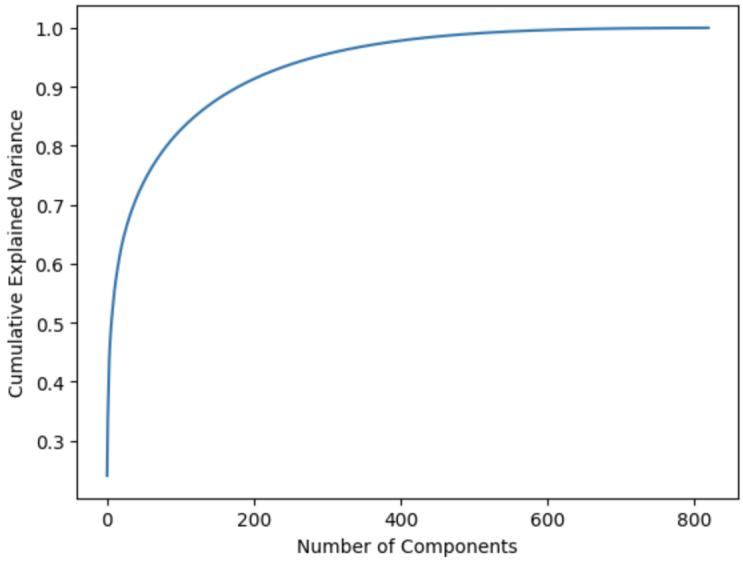

Figure 2: Scree plot for the time series split

5. The final cleaned-up data set that is used in predictive analysis: predictors and responses

In final, we used all 1,200 protein and peptide counts as our predictors. Additionally, for the time-split data, we used the patient's first visit UPDRS values and the number of months since the first visit as predictors as well. The response values were UPDRS-1, 2, 3, and 4 scores. Therefore, for each algorithm we trained 8 different models and compared their scores.

6. Predictive Modeling Techniques and Parameters Tuning Procedures

For both the protein and peptide alone models and the time-split models we used the same algorithms and parameters in cross-validation. The models we used were Ridge Regression, Lasso Regression, Extreme Gradient Boosting, Random Forest Regression, and Extra Trees Regression. We chose these models in order to have a combination of linear and non-linear models.

The parameters used for cross-validation in the Ridge and Lasso models were the following alpha values: 0.1, 0.5, 1, 5, 10, 100.

The parameters for Extreme Gradient Boosting, Random Forest Regression, and Extra Trees Regression were limited due to the time complexity of the algorithms. Even after reducing the dimensions to 400, it still took too long to perform proper cross-validation on all of the parameters we would have liked. Therefore, for each, instead, we simply used the default parameters on the Python packages Sklearn and XGBoost.

7. Evaluation of Predictive Model Performance: Results from Resampling and Testing Dataset

For analyzing the predictive performance of each model, we used the R-squared metric, Root Mean Squared Error (RMSE), and the Symmetric Mean Absolute Percentage Error (SMAPE) metric. The SMAPE was the metric of choice by the AMP-PD, therefore we included that in addition to the traditional regression metrics covered in this course. The scores were determined using the model on the testing data sets.

Below is the evaluation results of various regression models on all UPDRS using only the protein and peptides as predictors:

Model	UPDRS 1	UPDRS 2	UPDRS 3	UPDRS 4
Ridge				
SMAPE	84.57%	102.09%	88.60%	141.28%
R2	0.046	0.092	0.121	-0.127
RMSE	27.23	29.65	171.82	6.74
Lasso				

SMAPE	71.87%	94.06%	83.33%	125.66%
R2	0.036	0.055	0.125	0.004
RMSE	30.01	30.86	170.91	7.10
Extreme Gradient Boosting				
SMAPE	68.67%	100.24%	97.59%	132.69%
R2	0.108	0.079	0.049	-0.028
RMSE	25.92	38.39	252.40	7.33
Random Forest				
SMAPE	72.14%	98.87%	91.65%	127.21%
R2	0.025	0.048	0.153	0.047
RMSE	26.22	31.23	224.6	6.78
Extra Trees Regression				
SMAPE	84.57%	102.09%	88.64%	141.28%
R2	0.045	0.09	0.121	-0.127
RMSE	27.225	29.65	171.81	6.744

Table 1: Regression Model Performance on just Protein and Peptide data

As we can see from Table 1, the highest performing model is the Extra Trees Regression.

Next, we performed the same modeling test on data that included time features and was split based on time. The results are shown below in Table 2:

Model	**UPDRS 1**	**UPDRS 2**	**UPDRS 3**	**UPDRS 4**
Ridge				
SMAPE	61.79%	88.50%	78.36%	138.49%
R2	0.387	0.545	0.632	0.255

RMSE	21.04	21.66	102.09	6.48
Lasso				
SMAPE	61.53%	92.35%	80.65%	138.11%
R2	0.368	0.485	0.496	0.177
RMSE	21.7	24.48	139.76	7.15
Extreme Gradient Boosting				
SMAPE	66.83%	97.11%	89.59%	136.39%
R2	0.136	0.201	0.186	-0.013
RMSE	29.66	38.03	226.05	8.82
Random Forest				
SMAPE	66.02%	96.58%	86.01%	133.25%
R2	0.118	0.167	0.217	-0.034
RMSE	30.28	39.63	217.42	9.00

Table 2: Regression results for models trained with time-based features as well as proteins and peptides

For the time-split models, the highest performing model was the Ridge Regression, which differs from the results of the models trained on just proteins and peptides. Comparing the Ridge Regression model from the time-split data and the Extra Trees Regression from the protein and peptide data, we see that the time-split model significantly outperforms the protein and peptide alone model.

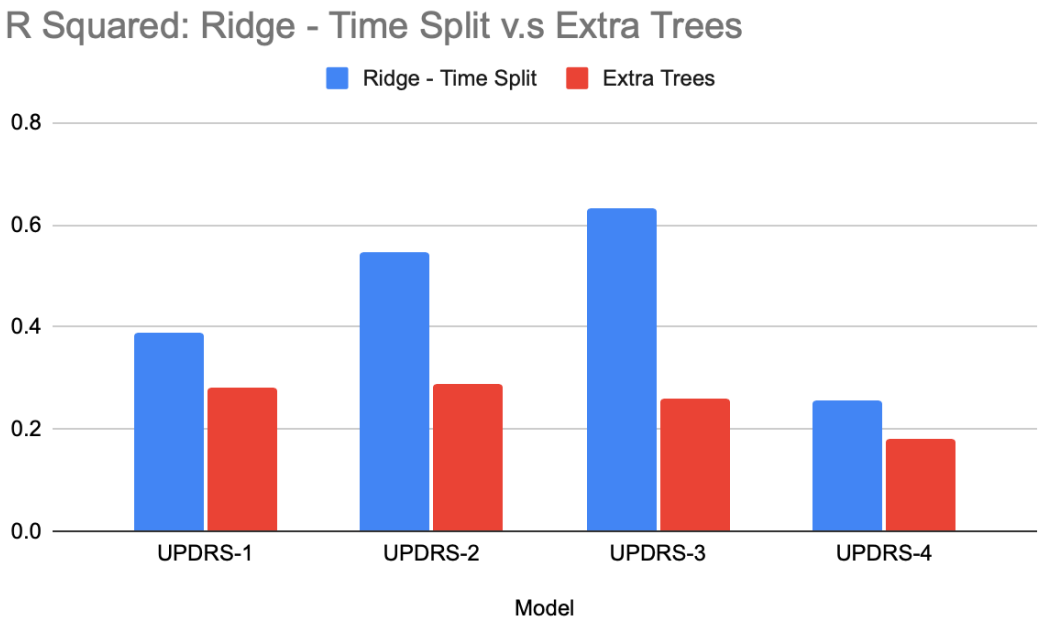

Figure 3: A comparison of the R squared value for the best models from the two different types of training

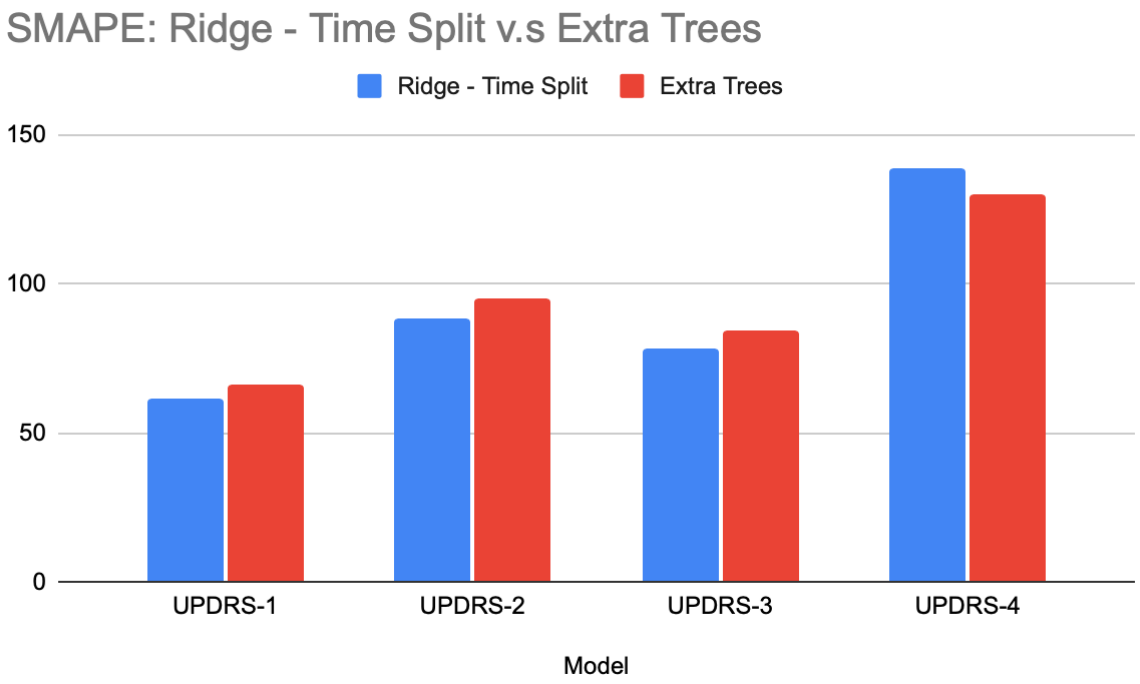

Figure 4: A comparison of the SMAPE value for the best models from the two different types of training

Based on the results observed in Tables 3 and 4, we can assess that the models that the time features greatly improve the predictability of the data using linear models.

8. Discussion of the predictors that are found to be important and whether these predictors agree with what a human expert would believe as important

From both the random forest and extra trees regression models, we found that the indicator variable for medication was the most significant predictor overall. This is a result that we expected but supports the efficacy of modern Parkinson's medication. When aggregated and averaged across both models and all UPDRS categories, there are about 10 proteins that have higher significance than the rest of the predictors. Below is a chart listing the top 15 important overall predictors minus the medication variable because its magnitude skews the scale of the graph.

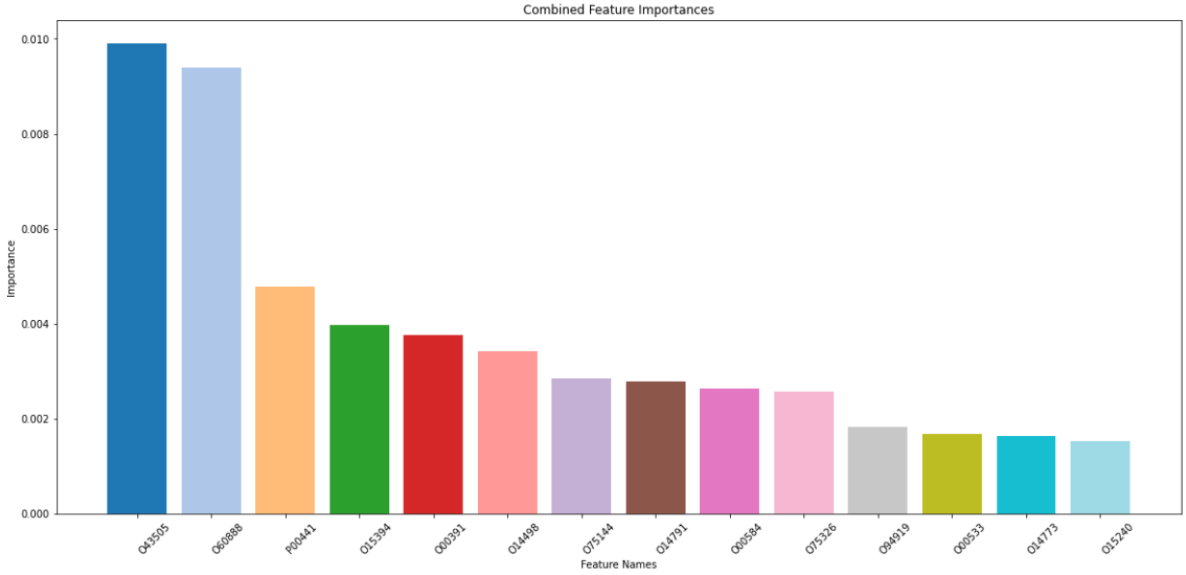

Figure 5: Scree plot for the top 15 important predictors

For the UPDRS 3 models, the peptide GYPGVQAPEDLEWER showed high significance and was the most significant predictor in the random forest model. In the UPDRS 4 models, the peptide AGLAASLAGPHSIVGR was the most significant variable in both the extra trees and random forest models. There are other peptides that show up near the top of the list, but this is an important finding and would be interesting to a human expert because it isolates specific peptides which appear to have an impact on the progression of Parkinson's disease symptoms.

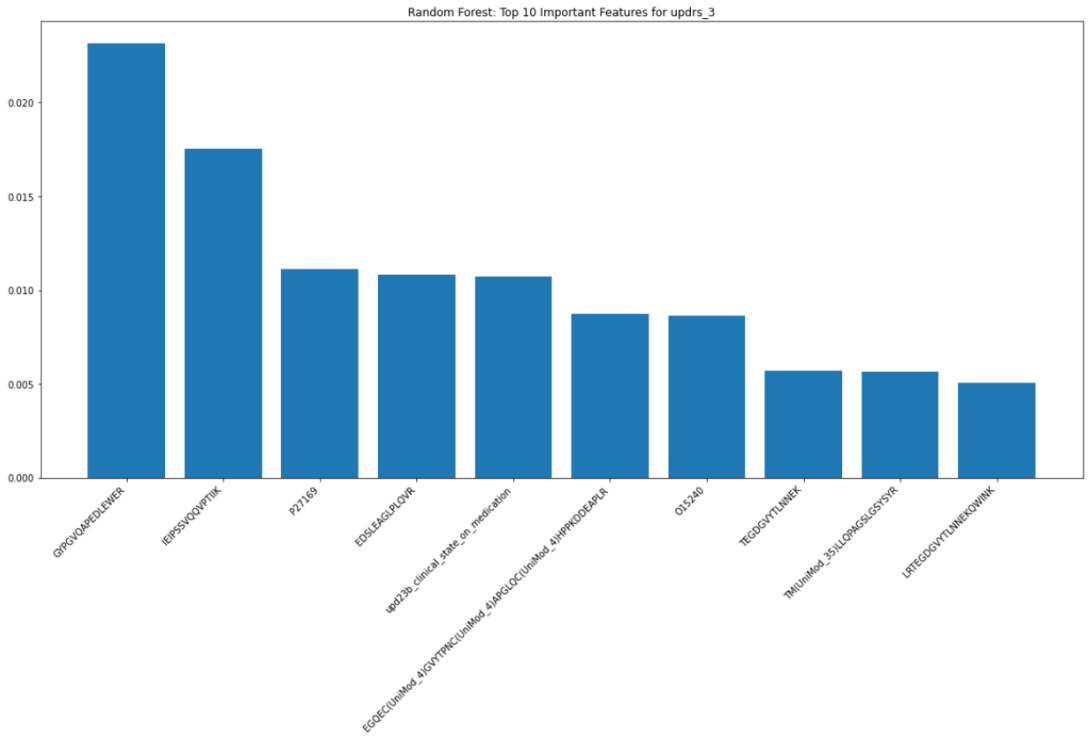

Figure 6: Random Forest: Top 10 Important Features for UPDRS_3

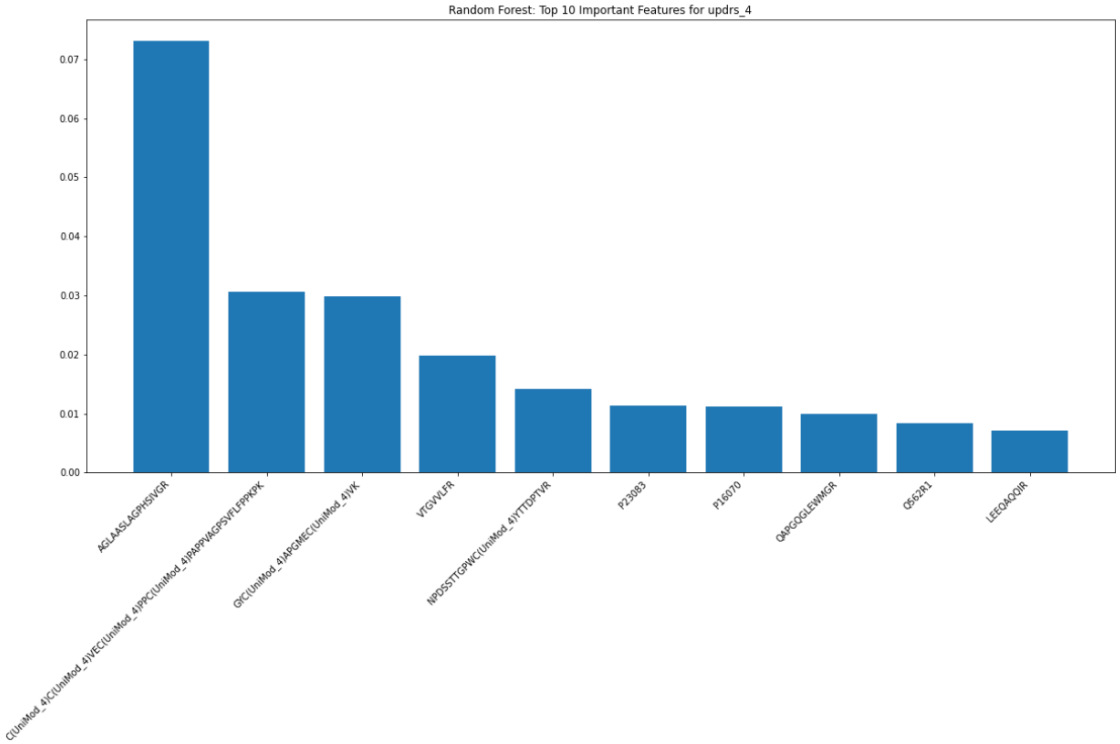

Figure 7: Random Forest: Top 10 Important Features for UPDRS_4

9. Detailed step-by-step instructions on how to run your codes with the data sets to reproduce your results. If your data sets are too large to upload, detailed instructions on where the data sets can be downloaded

Step by step instructions on where to obtain the datasets and how to run our code:

Step	Description
1	Go to the provided Google Colab link to access the code: https://colab.research.google.com/drive/1hZbMZ1qtvZedPOwWZYi_dM27G_V1ZA9S#scrollTo=9tt10UZJz973
2	Sign into your Google account if prompted.
3	Download the required dataset files from the Kaggle website using the provided link: https://www.kaggle.com/competitions/amp-parkinsons-disease-progression-prediction/data
4	In Google Colab, click on the folder icon on the left-hand side to upload all three CSV files
5	Once the necessary packages and datasets are loaded, run the code cells in the notebook by clicking Runtime.

Additionally, the data is contained in the zip file this report is submitted in.

Work Cited

AMP®-Parkinson's Disease Progression Prediction. (n.d.). Retrieved May 12, 2023, from https://kaggle.com/competitions/amp-parkinsons-disease-progression-prediction

Statistics | Parkinson's Foundation. (n.d.). Retrieved May 10, 2023, from https://www.parkinson.org/understanding-parkinsons/statistics