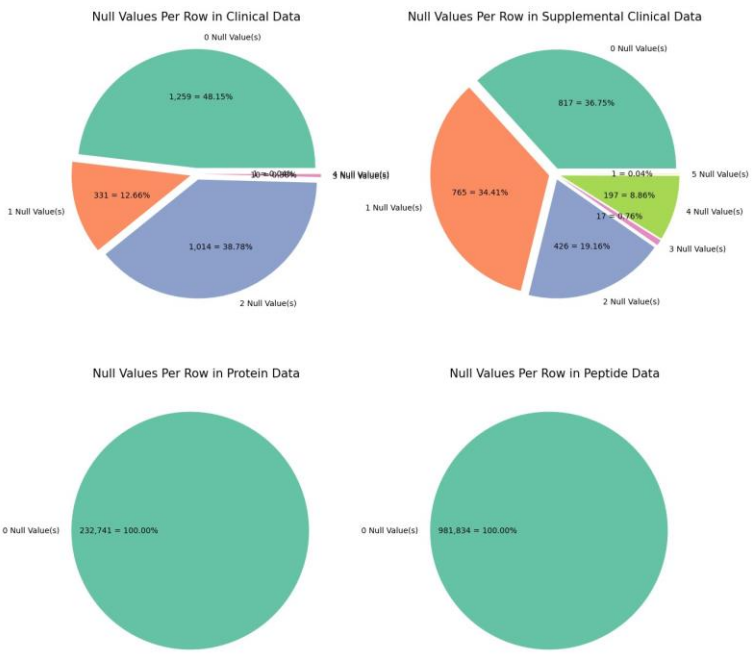
Parkinson's Disease Progression Prediction

Methods

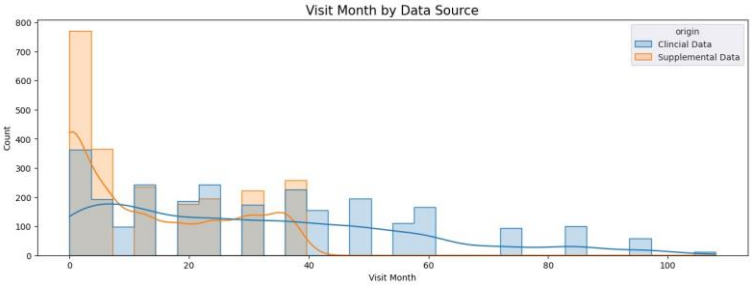
Exploratory Data Analysis and Preprocessing

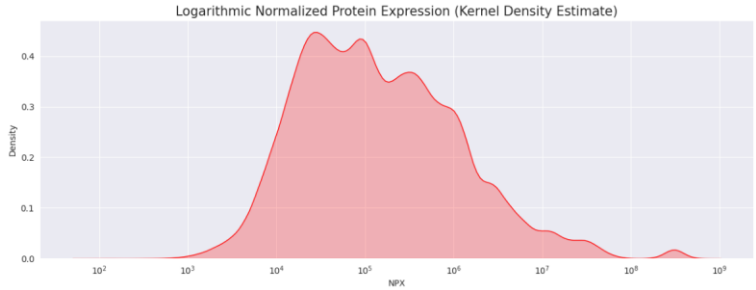
In the initial phase of exploratory data analysis, we observe that the datasets are small in size, thus not causing significant memory pressure. The analysis involves four datasets: train_clinical_data, train_peptides, train_proteins, and supplemental_clinical_data. Train_clinical_data consists of 2,615 rows and 248 unique patient_id values. Train_peptides contains 981,834 rows, 248 unique patient_id values, 227 unique UniProt values, and 968 unique Peptide values. Train_proteins includes 232,741 rows, 248 unique patient_id values, and 227 unique UniProt values. Supplemental_clinical_data comprises 2,223 rows and 771 unique patient_id values. In total, we have 4,838 unique visits, 1,019 unique patients, and 18 unique month values. The datasets contain a mix of categorical and continuous features, with train_peptides and train_proteins both joining the train_clinical_data dataset based on visit_id.

Upon examining null values, we find that peptide and protein data contain no null values. However, null values are present in the clinical and supplemental data. Single and double null values are most frequent in the updrs_4 and upd23b_clinical_state_on_medication features. Triple null values commonly occur in updrs_3, updrs_4, and upd23b_clinical_state_on_medication features for clinical data, while updrs_1, updrs_2, and upd23b_clinical_state_on_medication features for supplemental data. Quadruple null values appear mostly in updrs_1, updrs_2, updrs_4, and upd23b_clinical_state_on_medication features. Supplemental data has a higher number of rows with four null values than clinical data. Caution must be exercised when addressing null values, as it is unclear whether they signify missed assessments or can be assigned another value. For UPDRS assessments, setting the value to 0 may be incorrect as it indicates a "normal" result. For upd23b_clinical_state_on_medication, the impact of null values is undefined since the only valid settings are On or Off.

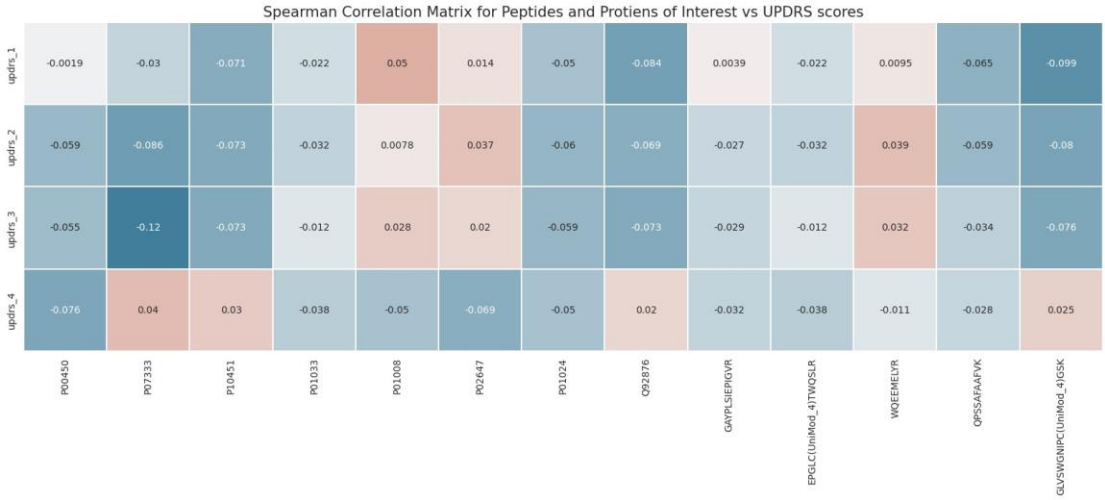


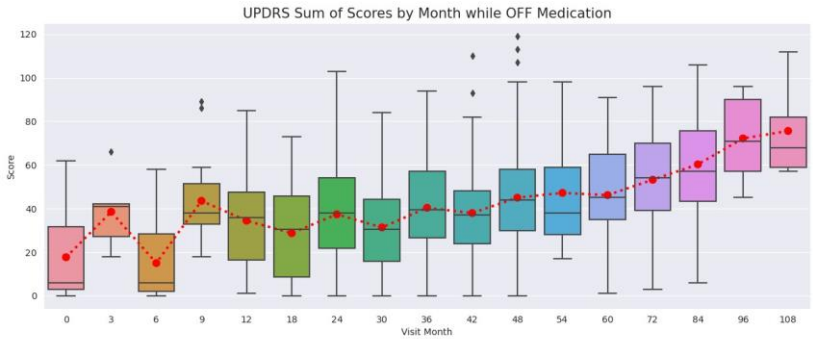
During the statistical analysis of continuous features, we found differences between clinical and supplemental data in terms of visit months, with supplemental data mainly covering 0 to 36 months and clinical data spanning 0 to 108 months. Kernel density estimates also confirmed these differences. UPDRS Part 1 and 4 scores showed similar distributions in both data sources, while UPDRS Part 2 and 3 scores had a higher proportion of 0-based scores in clinical data. Protein expression frequency values exhibited a wide range, and a logarithmic scale was employed to visualize kernel density estimates. The analysis revealed high variability in normalized protein expression. Peptide abundance also displayed a wide variation, with min, max, and standard deviation suggesting significant variability depending on the specific peptide. In summary, clinical and supplemental data have different month ranges, and both protein expression data and peptide abundance frequencies require further subgroup breakdowns to provide meaningful insights. The below graphs are few instances of the entire analysis.

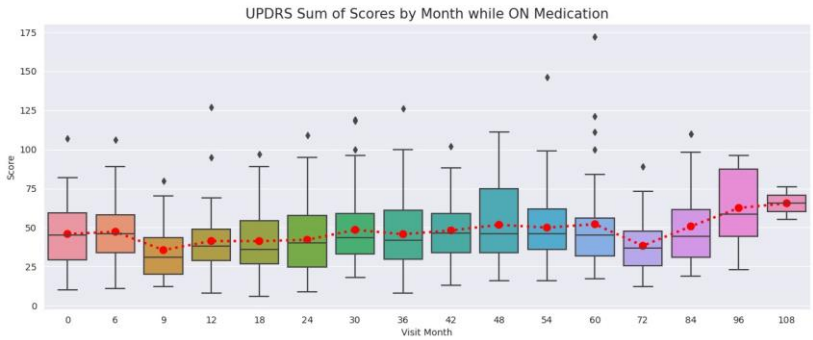




In the feature exploration section, we analyzed the impact of visit months on UPDRS scores and medication status, observing a large amount of variance and outliers for all UPDRS parts and visit months. UPDRS scores generally increased over time, indicating disease progression. The protein data showed stable amounts of protein expressions across each month category, with some proteins exhibiting significant increases or decreases across the months. The correlation matrix revealed weak correlations between some proteins and UPDRS scores, with missing values posing challenges for machine learning regression. Protein measurements existed for at most 40% of the visits, making it difficult to track trends. In terms of patients, protein data was lacking for nearly all patients at certain months. Finally, examining the research by Shi et al (2015), we found weak correlations between certain proteins and peptides and UPDRS scores. These observations suggest that no single protein or peptide in isolation had a clear correlation to UPDRS scores, but combinations of them might provide stronger signals. The below graphs are few instances of the entire analysis.

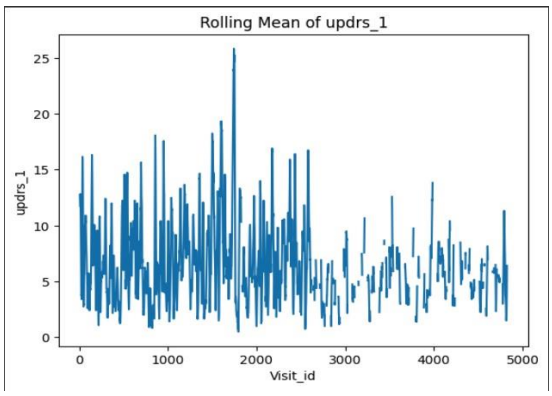


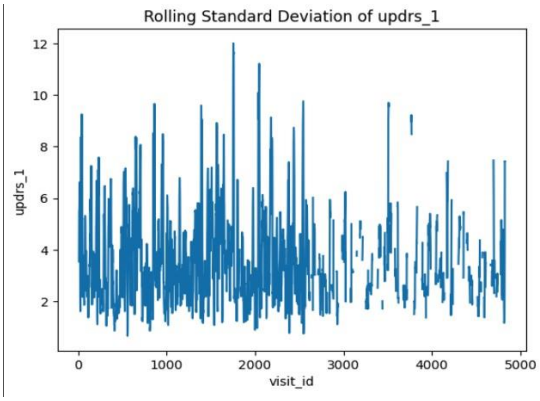




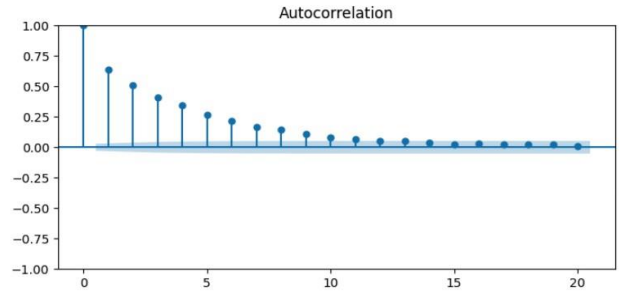
The dataset contains 4 individual time series data, named “updrs_1”, “updrs_2”, ”updrs_3” and “updrs_4”. For each of these time series, we have patient_id and visit_month. Combining pateint_id and visit_month gives us visit_id. To perform time series models on each of the individual time series data, we have to make sure that each of these series is stationary. Augmented Dickey-Fuller (ADF) is a statistical hypothesis test that checks for the presence of a unit root in the time series. A unit root indicates that the time series is non-stationary. If the p-value obtained from the ADF test is less than a significance level (e.g., 0.05), we can reject the null hypothesis of a unit root and conclude that the time series is stationary. Below is a screenshot of the ADF test for updrs_1, proving that the time series data is already stationary since p-value is less than 0.05. Test Statistic: -3.044229821894477 p-value: 0.030965294337072936.

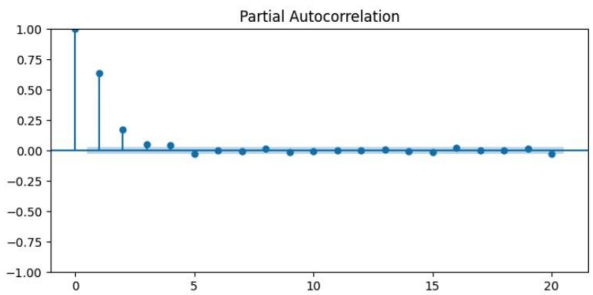
Further, rolling statistics like rolling mean and rolling standard deviation graph looks are calculated and plotted to visualize stationary data:





To successfully understand and build a time series model, it is necessary to identify the correlation structure between the observations of a time series. ACF (Auto-Correlation Function) shows the correlation between an observation and its lagged values. ACF helps in identifying the order of moving average (MA) terms in an ARIMA model. A sharp drop-off after a certain lag in an ACF plot suggests the presence of an MA term in the model. PACF (Partial Auto Correlation Function) plot shows the correlation between an observation and its lagged values after removing the effects on the intermediate lags. The PACF plots help in identifying the order of the autoregressive term (AR) in an ARIMA model. A sharp drop-off after a certain lag in PACF model suggests presence of AR term in the model.





Models

In our study, we employed various models to train on each of the UPDRS columns 1 through 4. These models included CatBoost, Linear Regression, Ridge Regression, Bayesian Regression, Automatic Relevance Determination Regression, Support Vector Regression, Decision Tree Regressor, Random Forest Regressor, K Nearest Neighbors Regressor, Stochastic Gradient Descent Regressor and ARIMA. By utilizing a range of different regression methods, we aimed to identify the most suitable model for predicting each UPDRS feature and assess their performance in relation to one another.

1. *CatBoost Model* : CatBoost, a gradient boosting algorithm, was selected for this project due to its capability of handling categorical features and missing data, as well as its strong performance and regularization. The algorithm uses target encoding to transform categorical features into numerical ones and has a mechanism to handle missing values. Additionally, CatBoost incorporates L2 regularization to prevent overfitting and provides feature importance

scores for interpretability. Overall, CatBoost was chosen for its robust predictive performance and ability to handle the complexities of the dataset while offering insights into important features for the prediction task.

2. *Linear Regression* : Linear regression is a statistical approach used to model the relationship between a dependent variable (also known as the response variable) and one or more independent variables (also known as explanatory variables or predictors). In the context of univariate time series data, the dependent variable is typically a sequence of observations over time, and the independent variable is time itself. In a univariate time series linear regression model, the goal is to find a linear equation that best describes the relationship between past observations and future observations. This linear equation can then be used to make predictions for future time points based on the historical values of the time series.

3. *Ridge Regression* : Ridge regression is a linear regression model that includes a penalty term to avoid overfitting. In the case of univariate time series data, ridge regression can be used to predict future values of the time series based on past values. The penalty term in the ridge regression model limits the complexity of the model and reduces the influence of individual data points, which can improve the accuracy of the predictions. The goal of the ridge regression model is to minimize the sum of squared errors between the predicted values and the actual values of the time series, while also minimizing the size of the coefficients in the regression equation.

4. *Bayesian Regression* : Bayesian regression is a statistical method for modeling relationships between variables in time series data, with the added benefit of quantifying uncertainty in the model parameters. It involves using Bayes' theorem to estimate the posterior distribution of the model parameters given the observed data, which provides a more complete picture of the uncertainty associated with the estimates. In contrast to classical regression methods, Bayesian regression allows for the incorporation of prior knowledge or assumptions about the model parameters, which can help to improve model performance and make the results more interpretable.

5. *Automatic Relevance Determination Regression* : The Automatic Relevance Determination (ARD) Regression model is a type of Bayesian regression model used for univariate time series data. It is a variant of the Ridge Regression model that introduces a prior distribution over the regression coefficients, allowing for Bayesian inference. The ARDRegression model estimates the posterior distribution of the regression coefficients, which enables us to estimate the uncertainty of the model's predictions. This model is useful when we have a large number of predictors and we want to automatically select the most important ones. It works by setting small or zero coefficients for irrelevant predictors, which effectively performs variable selection.

6. *Support Vector Regression* : Support Vector Regression (SVR) is a machine learning algorithm used for regression tasks that can be applied to univariate time series data. It works by mapping the data into a higher-dimensional space and finding a hyperplane that best fits the data while maximizing the margin between the hyperplane and the data points. SVR seeks to minimize the error between the predicted values and the actual values, subject to a regularization parameter that controls the trade-off between fitting the data and avoiding overfitting. The regularization parameter helps to prevent the model from becoming too complex, which can lead to overfitting and poor generalization to new data. SVR is often used in cases where the relationship between the predictor variables and the response variable is nonlinear and the data contains noise.

7. *Decision Tree Regressor* : The Decision Tree Regressor algorithm works by recursively partitioning the data into smaller and smaller subsets based on the values of the independent variables. At each node of the tree, the algorithm selects the variable and the threshold that result in the largest reduction in the variance of the dependent variable. The process continues until a stopping criterion is met, such as a maximum depth of the tree or a minimum number of samples per leaf. The resulting model can be used to make predictions for new values of the independent variable, based on the average of the dependent variable values in the corresponding leaf node of the tree. Decision Tree Regressor is a popular and powerful algorithm for modeling complex nonlinear relationships in time series data.

8. *Random Forest Regressor* : Random Forest Regressor is a type of ensemble machine learning model that uses multiple decision trees to make predictions on a given dataset. In the case of univariate time series data, it uses the historical values of the target variable (the time series) to predict its future values. Random Forest Regressor works by randomly selecting a subset of features and building decision trees on these subsets. The predictions made by each decision tree are then averaged to make the final prediction. This approach helps to reduce overfitting and increase the model's accuracy.

9. *K Nearest Neighbors Regressor* : K Nearest Neighbors Regressor is a machine learning algorithm used for regression tasks on univariate time series data. It predicts the value of a new data point by looking at the k-nearest neighbors to that point in the training dataset, where the value of k is a hyperparameter set by the user. The predicted value for the new data point is then determined based on the average (or weighted average) of the values of the k-nearest neighbors. In the case of univariate time series data, the KNN Regressor algorithm can be used to predict the next value in a time series based on the k previous values.

10. *Stochastic Gradient Descent Regressor*: Stochastic Gradient Descent (SGD) Regressor is a linear regression model that is used for training large datasets. It is a type of iterative optimization algorithm that updates the coefficients of the model based on the error between the predicted and actual values. In each iteration, a random subset of the training data is used to update the model coefficients. The objective of SGD is to minimize the cost function of the linear regression model. This cost function measures the difference between the predicted and actual values, and the algorithm tries to find the coefficients that minimize this difference. The SGD Regressor is useful for univariate time series data as it can handle large amounts of data and can be adapted to different types of cost functions.

11. *ARIMA (Autoregressive Integrated Moving Average)* : ARIMA (Autoregressive Integrated Moving Average) is a popular time series forecasting model used to predict future values of a univariate time series. ARIMA models are based on the assumption that the time series is stationary, meaning that the mean, variance, and autocorrelation structure of the series remain constant over time. ARIMA models are denoted as ARIMA(p, d, q), where p, d, and q are the orders of the AR, I, and MA components, respectively. The p and q orders denote the number of lag terms in the AR and MA components, while the d order denotes the number of times the time series needs to be different to become stationary.

Performance Metrics

1. *SMAPE (Symmetric Mean Absolute Percentage Error)* : We used the SMAPE metric for evaluating the performance of the CatBoost model in this project. The main reason for this choice was that SMAPE is a metric that is less sensitive to outliers than other common metrics like mean squared error (MSE) or mean absolute error (MAE). In our dataset, we had some extreme values in the target variables, which could have had a significant impact on the model's performance if we used MSE or MAE. Additionally, SMAPE gives equal weight to overestimations and underestimations, which was desirable for this project since both types of errors are equally important in predicting Parkinson's disease severity. Overall, SMAPE was a suitable choice for our project due to its robustness to outliers and its ability to capture both overestimations and underestimations in the model's predictions.

2. *MSE (Mean Squared Error)* : We used Mean Squared Error (MSE) as the evaluation metric for the other regression models because it is a common and widely used metric for regression problems. It measures the average squared difference between the predicted and actual values, giving more weight to large errors. Using MSE as the evaluation metric allowed us to compare the performance of different models on the same dataset and choose the best-performing model. Additionally, MSE is easy to interpret and provides a good measure of the overall model performance.

3. MAE (Mean Absolute Error) : We used Mean Absolute Erroe (MAE) as the evaluation metric for the ARIMA model, as it provides a measure of the accuracy of the model's point forecasts. The main advantage of using MAE is that it penalizes large errors proportionally, and does not overly penalize outliers. By comparing the MAE of different ARIMA models or variations of the same model, we can determine which model is more accurate and select the one

that provides the best forecasting results for our time series data. MAE is a scale-dependent metric, which means that its value is dependent on the scale of the data. Therefore, it is important to use other evaluation metrics, such as Root Mean Squared Error (RMSE), in conjunction with MAE to get a complete picture of a model's performance.

4. RMSE (Root Mean Square Error) : RMSE (Root Mean Squared Error) is a commonly used evaluation metric for time series models like ARIMA. Unlike MAE, RMSE puts more weight on large errors and thus penalizes the model more severely for making large errors. RMSE is useful for ARIMA models because it helps us to understand how well the model is performing in terms of predicting the actual values of the time series. The smaller the RMSE, the better the model is at predicting the future values of the time series.