Predicting Parkinson’s Disease Progression Using Machine Learning

(Group 11)

Goal and machine learning task:

In this study, the primary objective was to employ protein abundance data to predict the trajectory of Parkinson's disease, specifically by estimating the Unified Parkinson's Disease Rating Scale (UPDRS) scores. The investigation commenced with an exhaustive exploratory data analysis (EDA) to elucidate the underlying structures and patterns within the dataset, composed predominantly of mass spectrometry readings from cerebrospinal fluid (CSF) samples. This preliminary analysis was imperative for delineating the complex relationships and temporal dynamics present in the data, which includes a comprehensive array of peptides, proteins, and clinical metrics such as UPDRS scores and the effects of medication.

Subsequent to the EDA, the study identified several advanced modelling techniques as potentially efficacious for the predictive task at hand. The CatBoost algorithm was selected for its superior ability to handle datasets replete with missing values, a common challenge in empirical research. Support Vector Regression (SVR) was chosen for its aptitude in navigating the intricacies of high-dimensional spaces, a characteristic feature of the peptide and protein data. Finally, acknowledging the temporal dimension of the dataset, which encompasses longitudinal samples from patients over several years, Long Short-Term Memory (LSTM) networks were considered optimal for capturing the sequential dependencies critical for accurate UPDRS score prediction. Collectively, these methodologies underscore a multifaceted approach to modelling the progression of Parkinson's disease, leveraging both the static and dynamic features inherent in the dataset.

Contributions from group members:

Sanyam and Govind engaged in Exploratory Data Analysis (EDA) and data preprocessing activities, including the management of NULL values and the construction of datasets for the purpose of training models. Additionally, Govind conducted an examination of Linear Mixed Models to analyse both the random and fixed effects inherent within the data. Kashyap was responsible for the training and fitting of the Long Short-Term Memory (LSTM) model, alongside the optimization of hyperparameters to enhance model performance. Akshat engaged in the implementation of Support Vector Regression, utilising Bayesian optimization techniques for the fine-tuning of hyperparameters. Shreya focused on the application of CatBoost, encompassing the training, evaluation, and selection of hyperparameters for the model. Hetarth and Yashna dedicated efforts towards delineating the future scope of the project by examining various Kaggle notebooks for intriguing insights. Yashna focused on feature engineering and model fitting. She also concentrated on identifying and addressing the challenges associated with the project, undertaking a comprehensive literature review to explore potential avenues for improvement. Hetarth also worked on model fitting and worked on feature importance. A notable discovery from this exploration was the observation that data pertaining to proteins and peptides exhibit a lower signal strength in comparison to clinical data, a finding that was corroborated through the application of Linear Mixed Models (LMM). Ansh focused on the training of the LightGBM model, including the tuning of hyperparameters to optimise its performance.

Exploratory Data Analysis:

Upon examination, it has been observed that certain datasets exhibit missing entries. Specifically, the absence of values within the Unified Parkinson's Disease Rating Scale (UPDRS) columns is attributed to the non-uniform conduct of assessments across various visits. In the case of the 'upd23b_clinical_state_on_medication' column, a missing value signifies an ambiguity regarding the medication status of the patient.

An analysis of missing values reveals a distinct pattern: the 'updrs_1' column contains a single missing entry, the 'updrs_2' column has two, the 'updrs_3' column presents with twenty-five, and the 'updrs_4' column demonstrates a substantial count of 1,038 missing entries. Additionally, the 'upd23b_clinical_state_on_medication' column records 1,327 missing entries, indicative of unknown medication statuses for certain visits. Given that over 60 percent of the data within the 'updrs_4' and 'upd23b' columns are missing, a decision was made to exclude these columns from further analysis.

The data further reveals a linear decrement in the counts of visit months, signifying a reduction in patient follow-up visits over time. This might be due to a number of reasons - but it is also possible that this observation is directly related to the fatality rate of Parkinson’s Disease. This statement is only a speculation and is not within the scope of our analysis. A notable observation is the prevalence of zero values within the 'updrs_2', 'updrs_3', and 'updrs_4' columns, in contrast to a more evenly distributed range of values in 'updrs_1'.

Contrastingly, the dataset pertaining to protein data does not suffer from missing values, with protein counts for patients ranging between 190 and 1,900, which appears to correlate with the frequency of visits.

The peptide data similarly shows no deficiencies in entries. However, it is noteworthy that only select visits in the clinical dataset are accompanied by peptide entries. Peptide sequences are denoted using single-letter abbreviations for the twenty universally recognized amino acids, with glutamic acid being the most prevalent and tryptophan the least.

Algorithms used:

1. SVR with Bayesian Optimisation :

The main goal of SVR is to predict a continuous target variable, as opposed to SVM which is used for classifying discrete categories. SVR tries to fit the error within a certain threshold (epsilon). Data points that fall within the epsilon margin are considered as having been predicted correctly by the model, and no error is associated with these points.

Bayesian Optimisation : Bayesian Optimization is a strategy for optimization of functions that are expensive to evaluate. Bayesian Optimization works by constructing a surrogate probability model of the objective function. The surrogate is much cheaper to evaluate than the actual objective function and is typically a Gaussian Process (GP) for continuous functions, but can also be a Bayesian neural network, random forests, or other machine learning models that can provide a measure of uncertainty.

It uses an acquisition function to decide where to sample next. The acquisition function determines how to balance exploration (sampling where the model is uncertain) and exploitation (sampling where the model predicts high performance). The aim is to improve the model's understanding of the objective function with the least amount of samples possible.

2. Cat Boost :

CatBoost, is a gradient boosting algorithm that uses ordered boosting and seamlessly handles categorical features. Both techniques introduced by the algorithm aim to work against overfitting which is caused by data leakage.

Ordered Boosting: Ordered Boosting in CatBoost leverages the natural order of categorical variables to guide the split decision process, which aids in reducing overfitting by incorporating a form of regularisation based on category-wise target variable averages. This approach enhances model robustness and generalisation performance, particularly in datasets with categorical features.

Ordered Target Statistic: CatBoost uniquely handles categorical features using Ordered Target Statistics, where target statistics for each data point are calculated from a sequence set by random permutations of training data. This ensures predictions rely on historically observed data. A single permutation, however, might introduce variance in target statistics for earlier data points due to limited historical information. To counter this, CatBoost employs multiple random permutations during gradient boosting, distributing variance more evenly across data points. This method improves the reliability of target statistics, reduces overfitting risks, and results in more accurate models, significantly enhancing how categorical data is interpreted and boosting overall predictive performance.

3. Light GBM :

Light GBM, or Light Gradient Boosting Machine, is a highly efficient and scalable implementation of gradient boosting framework. Developed by Microsoft, it stands out for its ability to handle large-scale data and its speed in training models without compromising accuracy. What sets Light GBM apart from other machine learning algorithms is its unique approach to building trees. Unlike traditional gradient boosting methods that grow trees level-wise or breadth-first, Light GBM employs a leaf-wise, or depth-first, growth algorithm. This approach allows for more complex models by choosing the leaf with the highest loss to split on during the tree growth process, leading to faster learning and improved efficiency.

LightGBM uses the following in its mechanism which sets it apart from other Boosting Algorithms:

Gradient-based One-Side Sampling (GOSS) is a feature within Light GBM that improves model efficiency by prioritising data points with large gradients for training. This approach reduces the computational load by focusing on the most informative instances, enabling faster training times and lower memory consumption without losing accuracy. By selectively sampling instances, GOSS ensures Light GBM remains effective for large datasets, making it a smart choice for handling complex machine learning tasks.

Exclusive Feature Bundling (EFB) is another efficiency-boosting feature in Light GBM, designed to manage datasets with numerous sparse features. EFB bundles together mutually exclusive features, which rarely take non-zero values simultaneously, thus reducing the feature space. This reduction in dimensionality speeds up the computation and shrinks the model size, all while maintaining model performance. EFB complements GOSS, collectively streamlining Light GBM's operation and reinforcing its suitability for high-dimensional data scenarios.

4. Long Short Term Memory (LSTM):

Our decision to explore LSTM neural networks was driven by their adeptness in handling sequential data, crucial for uncovering temporal patterns within Parkinson's disease datasets. The sequence length of 3 was selected to strike a balance between data inclusivity and the capture of temporal relations, as 235 out of 248 patients had at least three recorded visits. This facilitated LSTM model training and testing, ensuring uniformity across sequences spanning three consecutive visits. A strategic partitioning approach ensured that patient IDs were independent between the training and testing datasets, maintaining an 80-20 split between the training and testing sets, respectively.

In parameter tuning, we explored variations in the number of layers (1, 2, 3) and units (64, 90, 128, 160) in the LSTM architecture. Additionally, we experimented with different numbers of epochs (50, 100, 150) to train the model. The activation functions for both input and output layers were varied between identity and sigmoid functions. We evaluated the performance with different optimizers, including SGD, Adam, and Nadam, while also adjusting batch sizes (20, 50) for training the model utilising MAE as the loss function.

Following iterative tuning, our LSTM model, employing MAE as the loss function, the identity input and output activation and the Adam optimizer, demonstrated superior performance. With a configuration of two LSTM layers (128 units and 64 units) and a dense output layer, it effectively predicted three severity scores per sequence. The LSTM model was trained over 100 epochs with a batch size of 20, effectively capturing temporal dynamics. This refined architecture emerged as the top performer through our iterative tuning process. For the optimal architecture of LSTM, the test MAE for UPDRS score 1 was 3.721, for UPDRS score 2 was 4.737, and for UPDRS score 3 was 11.452.

In LSTM modelling, challenges included optimising sequence length, fine-tuning hyperparameters, and selecting relevant features. Balancing sequence length required careful consideration to balance data inclusivity and temporal relation capture. Hyperparameter tuning involved thorough optimization efforts, requiring the selection and execution of various trial-and-error combinations to refine the model's performance. Understanding feature importance within the LSTM model was complex, presenting a significant challenge.

Results:

Hyperparameters:

The following table provides an overview of the best hyperparameters tuned to attain the presented results. It delineates the process undertaken to optimise the models, ensuring robust performance across various algorithms. Additionally, it underscores the significance of hyperparameter tuning in refining model behaviour and improving predictive accuracy.

Algorithm	Best performing model hyperparameters
SVR	'C': 42.99295429333525, 'epsilon': 0.4207143255561162, 'gamma': 0.0006286281438056463
Catboost	allow_const_label: True, bootstrap_type: 'MVS', colsample_bylevel: 0.01, depth : 5, iterations: 1000, l2_leaf_reg: 6, learning_rate: 0.015, objective: MAE, eval_metric: 'MAE',
LightGBM	'model__num_leaves': [31], 'model__learning_rate': [0.05], 'model__n_estimators': [500]
LSTM	Layer 1: 128 units Layer 2: 64 units Dense layer: 3 Epochs = 100 Batch Size = 20 Optimizer = Adam Loss = MAE I&O Activation **=** Identity

Based on our comprehensive analysis, LightGBM (LGBM) emerged as the top-performing model, showcasing superior performance attributed to its adept handling of high-dimensional data and feature collinearity. Its Exclusive Feature Bundling (EFB) capability proved invaluable in managing the complexities inherent in our dataset. Following closely, Long Short-Term Memory (LSTM) demonstrated competitive performance, notable for its relevance in capturing temporal dynamics within healthcare datasets, effectively modelling sequential patterns and temporal dependencies. Support Vector Regression (SVR) exhibited competence in capturing nonlinear relationships within the dataset, although it was outperformed by ensemble methods such as LightGBM and LSTM. Despite respectable performance, CatBoost (CB) ranked slightly lower compared to other models, falling short of LightGBM and LSTM, despite its ensemble nature and effective handling of categorical features. The results of these algorithms are shown below in Fig. 1 and Fig. 2.

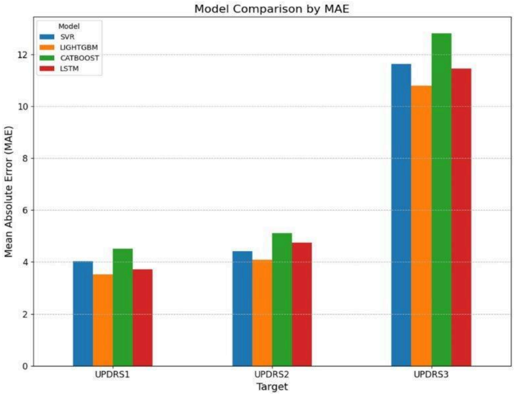
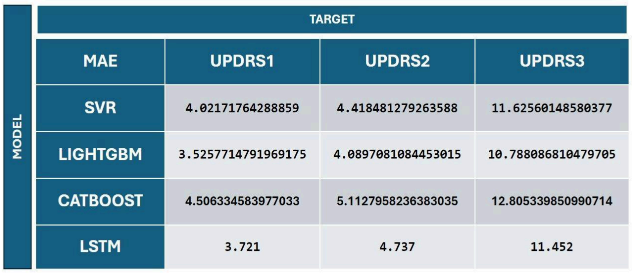


Fig. 1: Bar chart depicting model comparison by MAE



		TARGET		
	MAE	UPDRS1	UPDRS2	UPDRS3
MODEL	SVR	4.02171764288859	4.418481279263588	11.62560148580377
	LIGHTGBM	3.5257714791969175	4.0897081084453015	10.788086810479705
	CATBOOST	4.506334583977033	5.1127958236383035	12.805339850990714
	LSTM	3.721	4.737	11.452

Fig. 2: MAE for the algorithms used

In summary, LightGBM demonstrated superior performance among the evaluated models, attributed to its adept handling of high-dimensional data and feature collinearity. Following closely, LSTM showcased competitive performance, highlighting its relevance in capturing temporal dynamics within healthcare datasets. SVR exhibited competence in capturing nonlinear relationships, albeit slightly lower than ensemble methods like LightGBM and LSTM. CatBoost, while still respectable, ranked slightly lower in performance. This ranking provides valuable guidance for selecting optimal models tailored to specific healthcare data analysis tasks, with LightGBM and LSTM emerging as top contenders.

Challenges and Addressing Strategies:

Data Quality and Completeness: One significant challenge encountered was the prevalence of missing data within the dataset. To mitigate this issue, we explored various imputation techniques. These methods enabled us to fill in the missing values with reasonable estimates, thereby enhancing the dataset's completeness and improving the overall data quality.

High Dimensionality: The dataset presented a challenge typical in medical datasets, characterised by high dimensionality due to numerous protein/peptide features. In this context, LightGBM emerges as a powerful solution. Its Exclusive Feature Bundling (EFB) capability efficiently manages the high-dimensional space by bundling mutually exclusive features. This feature of LightGBM makes it particularly well-suited for handling the vast array of protein/peptide features present in our dataset. Furthermore, LightGBM's innate ability to handle features with high collinearity complements its efficacy in addressing the complexities inherent in biological datasets. This combined functionality ensures that LightGBM can effectively navigate the intricate relationships among features originating from similar sources, thereby enhancing the overall robustness and interpretability of our model.

High Collinearity and Feature Selection: Another challenge identified was the high collinearity among features, particularly arising from proteins originating from the same source. To tackle this issue, we implemented feature importance techniques. These methods facilitated the identification of crucial features (proteins), allowing us to mitigate collinearity effectively and enhance the model's predictive performance.

Temporal Dynamics: The dynamic nature of health data posed a significant challenge, as patient data could evolve over time, requiring the model to capture temporal dynamics accurately. To address this challenge, we adopted Long Short-Term Memory (LSTM) architecture. LSTM models are well-suited for analysing temporal data, enabling us to capture sequential patterns and temporal dependencies within the dataset effectively.

Hyperparameter Tuning: The process of hyperparameter tuning presented computational challenges, particularly with the exponential growth in configurations to evaluate during Grid Search. To overcome this, we employed Bayesian Optimization techniques. By intelligently selecting the next set of hyperparameters based on the current understanding of the objective function, Bayesian Optimization enabled us to optimise model performance efficiently while minimising computational resources.

Future Scope:

1. Research Scope:

Many of the Kaggle competition winners argue that the signal provided by the protein data is very low as compared to patient visit dates. Particularly [2] argues that using random numbers as a feature increases the CVScore by almost the same amount as Protein data does. This is surely an indication that the biological data is as good as random noise and putting time on constructing more features around them is futile and one should rather spend time on creating features around patient visit dates. In these sources, the authors also argue that adding Protein data leads to a Curse of Dimensionality-like problem. The authors argue that since there are just 248 patients and their data, one should just train 25 odd features and not 1195 features (by combining all protein features).

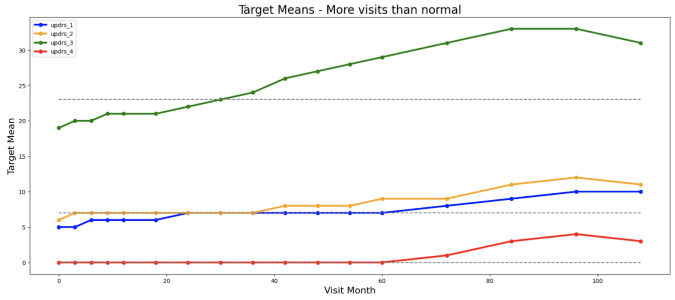


Fig. 3: Chart Describing increase in the Updrs scores for people having more visits than normal

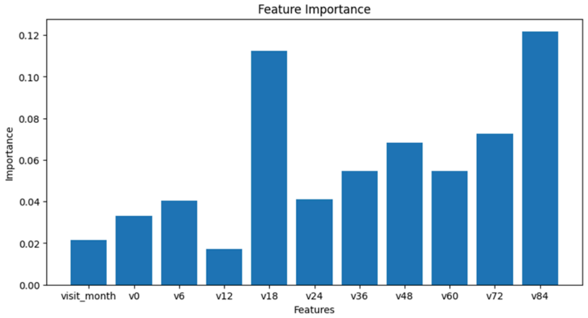


Fig. 4 Chart describing the temporal feature importance using a neural network feature perturbation method

Another insight is that there is a strong positive correlation between the frequency of doctor visits and scores (Fig. 3). Then we also used the Mixed Effects Model in order to find out the key features. Using a mixed effects model not only helps us figure out which features have a real impact on our outcome but also improves our dataset. This makes it easier for our machine learning model to learn from the data, ultimately helping us predict scores more accurately.The results from using Linear Mixed Models also confirm the idea that there is little signal in the proteins and peptides data. The only statistically significant effect of peptides is only on updrs_1 (p value: 0.028). For all other scores, there is no statistically significant effect from neither proteins nor peptides. Then we use feature engineering to capture the temporal information about the patient's visits, instead of protein data. We iterate over the DataFrame row by row and for each row (which represents a patient's visit) a variable v is tracked for the current visit month of that patient. For each possible visit month in the list [0,6,12,18,24,36,48,60,72,84], we check if the patient has visited the doctor in that month. If the patient has visited, the corresponding boolean feature (e.g., v6 for month 6) is set to 1. If the patient has not visited, the feature is set to 0.

We train and ANN model on top of it while tracking the progress of gradients concerning the inputs and the outputs to see the variable importance of each of the given parameters (Fig. 4)

[1] *AMP®-Parkinson's Disease Progression Prediction*. Kaggle. (n.d.). https://www.kaggle.com/competitions/amp-parkinsons-disease-progression-prediction/discussion/411505

[2] *AMP®-Parkinson's Disease Progression Prediction*. Kaggle. (n.d.-a). https://www.kaggle.com/competitions/amp-parkinsons-disease-progression-prediction/discussion/411398

2. Enhanced Data Collection: Future efforts should include collaborating with competition organisers to obtain missing data and expanding datasets with diverse variables to improve the models' predictive capabilities and account for a broader range of factors influencing Parkinson's disease progression.

3. Advanced Feature Engineering: There's a need for further feature engineering to capture changes in protein levels over time, explore protein interactions, and employ techniques like PCA to condense complex data into more potent predictive features.

4. Ensemble Approaches: Leveraging ensemble modelling can improve predictions by combining the strengths of various algorithms, reducing overfitting, and providing a multi-faceted view of the data.

5. Novel Data Transformation Techniques: Time series data could be transformed into visual formats to exploit CNNs and other deep learning methods, potentially revealing intricate patterns not discernible in the original format.

6. Cross-Domain Methodologies: Transforming time series data into other domains might provide fresh insights and innovative modelling techniques, offering new avenues for understanding and managing Parkinson's disease progression.

Appendix:

1. Catboost code
2. SVR code
3. LSTM code
4. LightGBM code
5. Future Feature Engineering
6. EDA+Statistical Analysis
7. Extensive Report