

# Predicting Parkinson's Disease Progression

By: Samuel Flusche, Ruben Valdovinos, Eric Vandament

The objective of this project was to make predictions of patients' Parkinson's Disease progression. Within this article, we will cover background information, about our data, Submissions on Kaggle, Conclusion, and moving forward.

The first question we had when starting on this project was what is Parkinson's disease? According to AMP-PD “Parkinson’s disease (PD) is a chronic and progressive neurological disease that is marked by tremors in the resting muscles, rigidity, slowness of movement, impaired balance and a shuffling gait, In addition, many people with PD develop non-motor symptoms such as behavioral changes and cognitive impairment.” As you can see PD is a very serious disease, and to build off this there is no cure. However, they have found ways through therapy and treatments to relieve symptoms.

The second question we needed to answer before starting our project is what is a Kaggle competition? Kaggle is a website that holds datasets and also hosts competitions. Anyone can host a Kaggle competition. The goals of these competitions are to make predictions using models. However, these predictions must be formatted in a certain way, which is chosen by the host. Once the host receives these predictions, they grade them and upload people's scores to a live leaderboard which is updated throughout the competition.

Now that we have an understanding of what PD is and what a Kaggle competition is, it's time to dive into our data. Our competition was being held by Accelerating Medicines Partnership® Parkinson's Disease (AMP®PD). Which is a Public-Private partnership made in order to help study PD. Their goal is to predict the severity of one's Parkinson's throughout its progression, this is what our predictions will be aimed towards. The way they created this dataset was by taking samples from thousands of patients with PD.

Within the dataset AMP-PD created there are certain variables that are important those being;

- **Udprs** - Rating scale for Parkinson's Disease symptoms, range 1-4, 4 being worst
- **Peptide** - Sequence of amino acids within the peptide
- **PeptideAbundance** - Frequency of the amino acid in the sample
- **UniProt** - The UniProt ID code for the associated protein
- **NPX** - Normalized protein expression
- **upd23b\_clinica\_state\_on\_medication** - Was the patient taking medication
- **visit\_month** - month patient came on visit

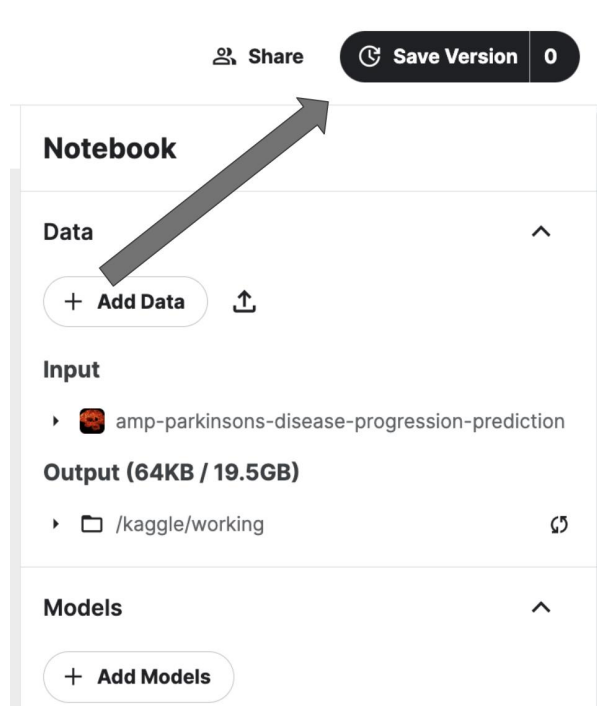
After we knew what our data looked like, it was time to figure out how to make a submission on Kaggle.

When we began to work on Kaggle we had never submitted an entry to a competition before and it was quite difficult the first time around. So the first thing our group did was copy

someone's code just so we can see how they formatted their submission to Kaggle. We found out that for this particular competition, you needed to have a data frame with three different columns.

prediction...	# rating	# group_key
3342_0_updrs_1_plus_0_months	0	0
3342_0_updrs_1_plus_6_months	0	0
3342_0_updrs_1_plus_12_months	0	0
3342_0_updrs_1_plus_24_months	0	0
3342_0_updrs_2_plus_0_months	0	0
3342_0_updrs_2_plus_6_months	0	0
3342_0_updrs_2_plus_12_months	0	0
3342_0_updrs_2_plus_24_months	0	0
3342_0_updrs_3_plus_0_months	0	0
3342_0_updrs_3_plus_6_months	0	0

This is what the submission format looks like. To begin this process we needed to save a version of the notebook. On Kaggle there is a save version button on the top right of the screen that you click which looks like this:



After we clicked on the save button it brought us to a screen where we were given multiple options. The options were name, version type, and advanced settings. Version type allowed you to save your changes to your notebook or save your notebook for a submission. Advance settings allowed you to select how hard you wanted your computer to try and run the notebook.

×

Save version

VERSION NAME

Version 1

9 / 50

VERSION TYPE

✓ Save & Run All (Commit)

Run a fresh copy of your notebook and save the output

▼

Advanced Settings

Save

After understanding what we could do on this screen we then gave our notebook a version name (version #1) and selected Save & Run All. Then we clicked save. Once we did that we saw that our submission was running and being saved so we waited for a few minutes.

**Submission trend**

just now

Version #1

Running: just now

...

**Submission trend**

3 minutes

Interactive Session

Running: 3 minutes

...

**Linear model RV**

41 minutes

Interactive Session

Cancelled

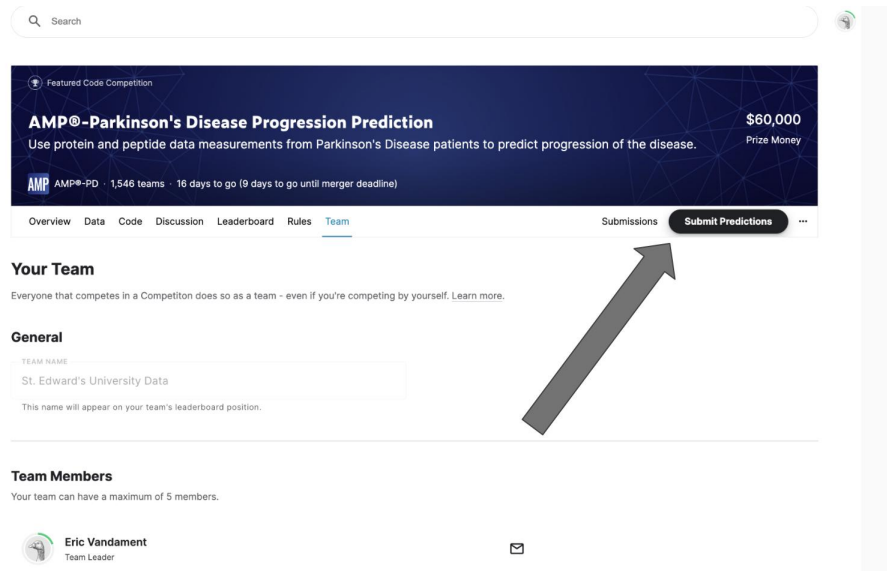
...

2

2 Active Events

▼

When version #1 completed running we then had to go back to the front page of the Kaggle competition and click submit predictions.



Search

Featured Code Competition

### AMP®-Parkinson's Disease Progression Prediction

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

**\$60,000**  
Prize Money

AMP AMP-PD · 1,546 teams · 16 days to go (9 days to go until merger deadline)

Overview Data Code Discussion Leaderboard Rules **Team** Submissions **Submit Predictions** ...

#### Your Team

Everyone that competes in a Competition does so as a team - even if you're competing by yourself. [Learn more.](#)

##### General


TEAM NAME


St. Edward's University Data

This name will appear on your team's leaderboard position.

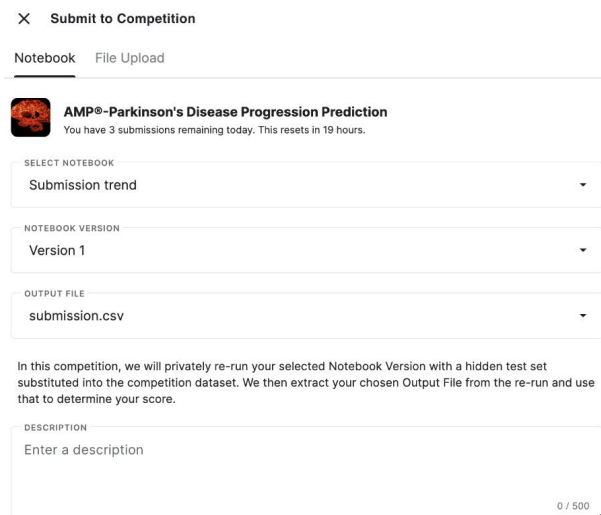
##### Team Members

Your team can have a maximum of 5 members.

 **Eric Vandament**  
Team Leader



Clicking on submitting predictions brought us to the following page. This page allowed us to specify which notebook we wanted to submit, what version of the notebook we wanted, and which output file we wanted from that notebook.



✕ **Submit to Competition**

Notebook File Upload

### AMP®-Parkinson's Disease Progression Prediction

You have 3 submissions remaining today. This resets in 19 hours.

SELECT NOTEBOOK

Submission trend

NOTEBOOK VERSION

Version 1

OUTPUT FILE


submission.csv

In this competition, we will privately re-run your selected Notebook Version with a hidden test set substituted into the competition dataset. We then extract your chosen Output File from the re-run and use that to determine your score.

DESCRIPTION

Enter a description

0 / 500

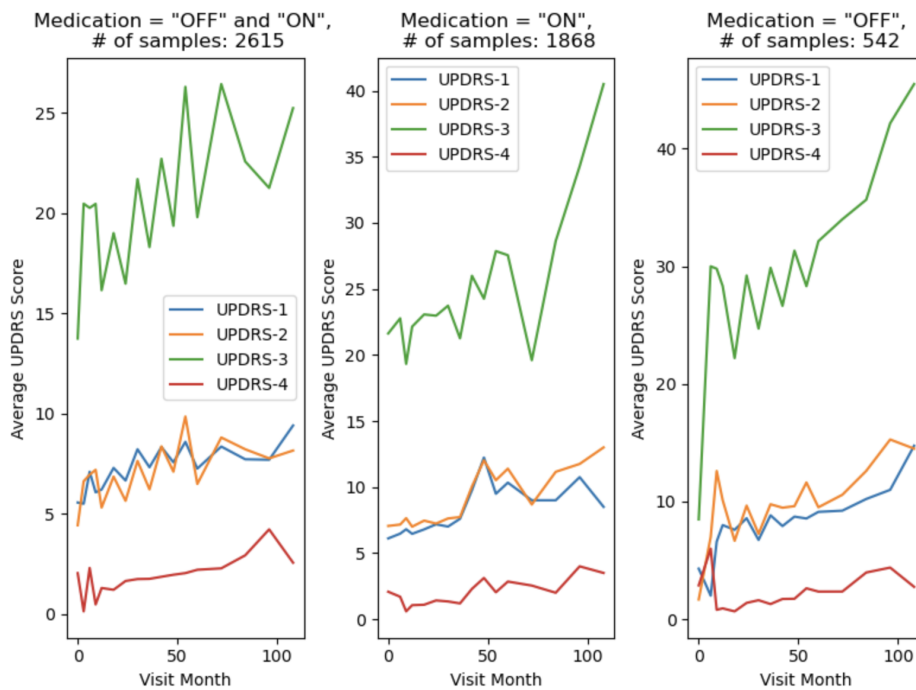


So we selected what we wanted to submit. This is very important because, for this Kaggle competition, we only get 5 submissions per day so if we submitted the wrong notebook it wastes a submission. Once we found out that we could waste submission we were very careful about selecting what was worth submitting. After completing all those steps we finally get our score of 56.1.

The screenshot displays the Kaggle AMP-PD competition interface. At the top, there's a search bar and a banner with the competition title: "Use protein and peptide data measurements from Parkinson's Disease patients to predict progression of the disease." Below the banner, navigation tabs include Overview, Data, Code, Discussion, Leaderboard, Rules, and Team. The Submissions tab is active, showing a "Submit Predictions" button and a "0/2" submission count. The main section is titled "Submissions" and includes instructions: "Select up to 2 submissions that will count towards your final leaderboard score. If less than 2 are selected, Kaggle will automatically select from your best scoring submissions. Learn More". There are filters for "Auto-selection candidates" and tabs for "All", "Successful", "Selected", and "Errors". A table lists submissions with columns for "Submission and Description", "Public Score", and "Select". The first submission, "Submission trend - Version 1", is highlighted with a green checkmark, indicating it succeeded. A large grey arrow points from this submission to its public score of 56.1. The submission was made by "Ruben Valdovinos" 2m ago. The second submission, "Linear model DV - Version 1", is partially visible below.

Now that we understood how Kaggle submissions worked, we were able to start performing the actual analysis. This meant starting with a look at other people's codes.

Starting with other peoples' codes is the best way to go in a Kaggle competition. While they are called Kaggle competitions, they have a very big element of connectedness and interactivity between different teams, which is very nice. The first place we started was using other codes to first better understand the most efficient ways to format data and then to see what other modeling techniques were possible and available. The best score at the time of our analysis was 53, however, there were several publicly available codes that gave a score of 56.1, which was very close to that top score so that is where we started model-wise. By looking at the codes that gave that and similar scores that people had posted, we found an interesting trend. That being the fact that many of the codes did not even touch the protein and peptide data whatsoever. Which was very interesting for several reasons. Number one, the protein and peptide data and its effect on the updrs score was a main focus of why this competition even exists in the first place. Secondly, it was also quite interesting that it was possible to get a score so good without the protein and peptide data at all. Because without that data the models are based almost exclusively off of just historical trends.



The only dataframe of the three provided was the clinical data dataframe which included only the visit id, the four updrs scores, and then whether or not the person was on medication. As the graphs above show though, there is a pretty consistent trend present for each of the updrs scores over time, which is probably why the model was able to be so effective from looking only at the trends.

However, since this was not the full extent of the competition, we did want to experiment with using the protein and peptide data. To do this we merged the protein and peptide data with the clinical data and ran the same modeling on it. The 56.1 code used only a sklearn SGD Regressor, so we just ran that one. However, and unsurprisingly so, we found that the addition of the protein and peptide data did make the score less accurate overall. This makes sense because the data becomes significantly more complicated than just the clinical data and it makes it much more difficult to look at the trends over time.

We also experimented with other regressors and ensembling some of them, however, that still was not going to be enough to improve the score overall. Ultimately for someone to be successful in this project they will require significantly more advanced models using much more involved techniques than just regressors, which is where I believe that the winning teams will go in the future of this project.

The actual act of performing the analysis is also a little bit more complicated than we had realized due to the method of uploading submissions. We were limited to five submissions today

which did naturally put limitations on how much we could view our place in the contest as a whole. However, while submitting is a surefire way to find the score of ones predictions, it is not the only assess the accuracy. We were able to fine-tune models to test accuracy by splitting our training data into training and testing sets and testing them ourselves. However, it is still necessary to upload submissions overall and that can be complicated sometimes for this project. Uploading submissions requires the use of a custom api, which on paper is simple, but the more complicated the code and model, the more work that must go into creating the submissions. A big problem that we ran into is that failed submissions still count as a submission, and it will not tell you where the error in the submission lies either, which makes debugging rather difficult.

Overall though, while we were not able to make huge discoveries on Parkinson's and modeling of its severity, we were able to learn a lot about Kaggle competitions and the collaborative way that they function, which was valuable knowledge and the skills we gained from here are things that we can apply to many future endeavors.