# Final Project

Parkinson's disease is a progressive disorder of the central nervous system affecting movement and inducing tremors and stiffness. Nearly one million people in the U.S. are living with PD, and it is the second most common neurodegenerative disorder disease (Parkinson's Foundation, 2025). It affects dopamine-producing neurons in the brain and has five stages. Stage 1 has mild symptoms affecting one side of the body; stage 2 symptoms affecting both sides of the body; stage 3 is mid-stage with balance issues and slower movements; stage 4 has severe symptoms requiring assistance for walking; and Stage 5 is the most advanced stage with potential bedridden state and need for around-the-clock care. The chosen dataset comprises a range of biomedical voice measurements from 31 people; of those, 23 carry one of the five stages.

The model selected for my application, created based on the PD dataset, is to accurately detect the presence of Parkinson's disease in an individual. I used an XGBClassifier from XGBoost to combine multiple models and produce better predictions. XGBoost is an ensemble learning method that combines decision trees to create a strong predictor; it focuses on the errors from previous iterations to improve the model. It is an efficient approach for intricate problems with multiple factors contributing to the outcome, such as PD. The XGBClassifier allows me to classify data into different categories. XGBoost also includes L1(Lasso) and L2(Ridge) regularization to help prevent overfitting. It is known when dealing with medical data that the number of features might be larger than the number of samples.

The chosen project methodology was an ad hoc approach. It's a singular project that doesn't have any stakeholders or monitoring. In some form, I did follow five out of the six phases of CRISP-DM. I failed at the first phase: business understanding. However, that was due to a sudden desire for a project change. Phases two through six were thoroughly practiced except for deployment. Deployment wasn't planned very well. I had initially considered using Flask and Docker, but there were too many limitations. The next step to make this an operational product is to fully implement the tasks included in the deployment stage, such as planning the deployment, producing a final report, and planning for monitoring and maintenance.

There were a few discrepancies that I ran into that raised some alarms, such as the use of PCA for one model didn’t offer a high accuracy score as the model without. However, I surmised the cause to be due to the dimensionality reduction effect of PCA and information loss. Another alarming situation was how both models ended with the same accuracy score after hyperparameter tuning. Overall, I learned mainly about PD itself and which method works best for this particular set of data.