Detecting Parkinson's Disease from Typing Behaviour

Kyle Imrie
CMPT 733
krimrie@sfu.ca

Jessica Moloney
CMPT 733
moloney@sfu.ca

Summary

The project prepared covers the topic of predicting Parkinson's Disease using data collected from everyday typing activities. This topic and dataset were chosen based on an Australian study with the same research problem. After preprocessing, summary statistics of the keylog files are used as features for the machine learning pipeline. Multiple individual models were then trained using bagging to reduce variability, as well as cross-validation, using the Scikit-learn library. Those results were fed into an ensemble model that aggregated predictions. An F-score of 0.83 and a recall of 1.00 were achieved on our best model.

Motivation

Parkinson's Disease is the second most common neurodegenerative disease after Alzheimer's. Around 1 in 500 of the whole population and 1 in 50 of the elderly suffer from it.[1] No cure currently exists and there are no ways to slow the progression of the disease once developed.

An important concern surrounding Parkinson's Disease is that no lab test exists to detect its presence. Currently, a diagnosis is made by consulting with the patient's medical history, performing a neurological/behavioural examination, and observation with a movement specialist. Due to the lack of a solid and concrete marker for the disease, there exists a misdiagnosis rate of up to 25% among general non-specialist practitioners[2].

As a result of this difficulty, people are often properly diagnosed years after the disease has progressed and significant damage has been done. This is regrettable because, while reversing the disease isn't possible, treating the disease quickly can help to suppress the symptoms and to maintain quality of life for the patient. It is for all these reasons that we were drawn to a study that we found on Kaggle.

The Data

This study, conducted by Warwick R. Adams out of the Charles Sturt University in Australia, had 217 participants install a keylogger software called "Tappy" onto their computer[3]. "Tappy" recorded the keystroke timings of the participants while also maintaining privacy by only recording which side of the keyboard that was pressed, rather than the specific key. Using these data

---

[1] Tysnes, OB. & Storstein, A., "Epidemiology of Parkinson's Disease," (2017): 124: 901.

[2] A. Elbaz et al., "Epidemiology of Parkinson's Disease," (2016): 172:1.

[3] Adams W.R, "High-accuracy detection of early Parkinson's Disease using multiple characteristics of finger movement while typing". (2017)

points, the study created a machine learning pipeline that could detect whether or not a person had Parkinson's purely from typing behaviour.

The data included in the study were individual keylog files, which were associated with the unique ID of the participant and the date which the file was created. The keylog file itself consists of lines of data, each one corresponding to a single keypress and denoted with a timestamp. Each keypress records the side of the keyboard that the key belonged to and the direction from where the keystroke came from (right-hand side key to left, left-hand side key to left, and so on). In addition to these pieces of information are the three numerical variables which were of primary interest:

Table 1: Dataset Variables

Variable	**Description**
Hold Time	Time (ms) between press and release of current key
Latency Time	Time (ms) between press of previous key and press of current key
Flight Time	Time (ms) between release of previous key and press of current key

Using these variables, the original study obtained extremely positive results of 100% accuracy and 100% AUC (Area Under the Curve). Efforts were focused on creating our own models to achieve our own results rather than recreating the success of the original article written on the study.

Problem Statement

The question that we sought to answer was **"Can we use everyday typing behaviour to detect Parkinson's Disease?"**. An effective and accurate solution to this question would solve the issues with late diagnoses mentioned in the previous section. In addition, a typing-based detector would be a non-invasive and hassle-free solution to keeping track of one's health and the progression of Parkinson's Disease. This method would help those seek the needed help earlier which can help maintain quality of life.

This problem presented many challenges. As stated in the Methodology section below, the variables recorded by the "Tappy" software were either highly correlated or lack a well-defined decision boundary for classification, linear or otherwise. These issues made it more difficult than just feeding premade features into a machine learning model.

Data Science Pipeline

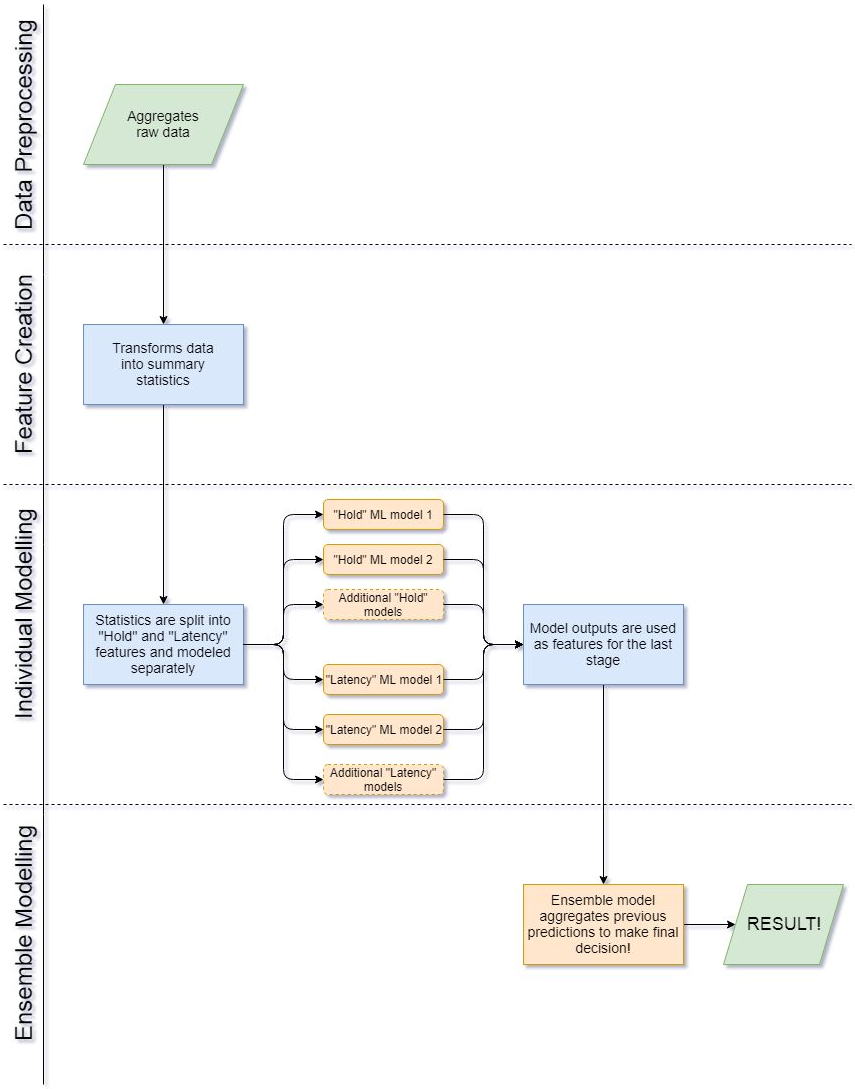
The layout of the Data Science Pipeline is fairly straightforward and can be separated into the following sections: *Data Preprocessing*, *Feature Creation*, *Individual Modelling*, and *Ensemble Modelling*. A visual representation of the pipeline may be found in Figure 1, below.

The first stage, *Data Preprocessing*, transforms the raw "Tappy" data from the study into readily usable data for the *Feature Creation* step. We first aggregate multiple "Tappy" files from each user into one, so that there exists one keylog file per user containing all their typing history. A second step is fixing and deleting corrupted keystrokes found from the original data set.

The *Feature Creation* stage involves the aggregation of user data and creation of variables that describe user behaviour. Understanding how a user *normally* behaves can give great insight into theirs and others' typing behaviour. The majority of features calculated from the preprocessed log

files are summary statistics of “Hold Time” and “Latency Time”, grouped by either which side the pressed key was on or by what direction the key was pressed in. Examples of these statistics are median, standard deviation, and skewness. Non-typing features that come from the participants themselves were also added, such as age and gender. These first two stages were completed using the Pandas dataframe library.

Figure 1: Data Science Pipeline



Next, the created features are fed into individual machine learning models in the *Individual Modelling* stage. In order to help with the curse of dimensionality, the study split the feature set into two groups: one group consisting of “Hold Time” features, and another group of “Latency Time” features. We implemented these same changes. This doubles the number of models fit and used for predictions, but it yields stronger results than fewer models trained on more features. Examples of the models used are Random Forest, Neural Networks, K Nearest Neighbours, and others, further noted in the appendix. We originally used the PySpark machine learning implementation of these models, but switched to Scikit-Learn when we decided to include more feature selection and cross-validation techniques that lacked a PySpark equivalent.

The results of the *Individual Modelling* stage are fed into the *Ensemble Modelling* phase. The outputs of these models are used as features to train ensemble methods of classification. The two models we used were a "Mean Probability Classifier", which weighs the probabilistic outputs of the models, and a "Voting Classifier", which takes a straight majority vote of the individual model predictions. Because our individual models could not produce good enough determinations of Parkinson's by themselves, this final step was required. We coded the *Ensemble Modelling* stage using Python.

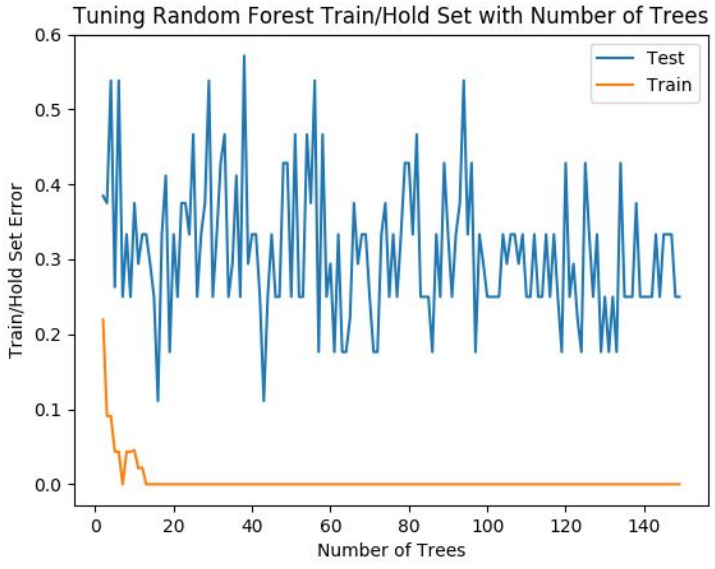
Methodology

The original article written for the study removed the participants from the study who were experiencing more than "Mild" self-reported symptoms of Parkinson's. In doing so, the models created on that dataset would be better able to catch the early signs of the disease rather than the more extreme tremors as the disease progresses. This filtering of data brings the total size of the dataset used down to 40 participants with Parkinson's and 32 without.

After preprocessing the raw dataset, exploratory data analysis was done on some of the basic summary statistics of the "Hold", "Latency", and "Flight" times in order to see if a simple pattern existed that could easily separate the Parkinson's Disease participants from the control group. The "Flight Time" variable was highly correlated with the other two variables, so it was discarded from further modelling efforts as it wouldn't provide much additional information to the model. Additionally, it was clear to see that no easy decision boundary could be found, linear or otherwise, to distinguish Parkinson's disease from the control group.

Another issue experienced during the modelling phase was that we could receive wildly varying prediction scores, likely due to the small training dataset. We would re-train our model with the same GridSearch cross-validated parameters and get completely different F-scores. For example, see how the test scores on the hold out set never quite settle down as the number of trees in the Random Forest increases (see Figure 2). To circumvent this issue, we implemented a bagging technique in order to increase the bias in the models while reducing the variance in its predictions. Once implemented, we found that the scores became much more stable.

Figure 2: Random Forest Tuning



Results and Future Work

High classification scores were achieved using the ensemble methods described above. With an F-score of 0.83 and a recall of 1.0, the Voting Classifier reaches the goal of having a strong overall predictive ability, while returning no false negatives. While we were unable to recreate the success of the model produced in the original study, we are proud of what we achieved given the difficulty of predicting with such variable data and such a small sample size.

What we envision as future steps for this project are twofold. We would have liked to implement a streaming version of the overall algorithm created, so that users could have a rolling prediction that is constantly updated as more keystrokes are recorded. Secondly, we would have liked to investigate the time-based nature of the data. Currently, the features we use eliminate all sense of time from the data, and while that was sufficient for the scope of our project, it would have been nice to use that data to add a temporal component to our predictions.

References

A. Elbaz et al., "Epidemiology of Parkinson's Disease," *Revue Neurologique* 172, no.1 (January 2016): , Accessed April 04, 2018, https://www.sciencedirect.com/science/article/pii/S0035378715009224?via=ihub.

Adams WR (2017) High-accuracy detection of early Parkinson's Disease using multiple characteristics of finger movement while typing. PLoS ONE 12(11): e0188226. Accessed March 03, 2018. https://doi.org/10.1371/journal.pone.0188226

A.J. Hughes, S.E. Daniel, L. Kilford, A.J. Lees. "Accuracy of clinical diagnosis of idiopathic Parkinson's disease: a clinicopathological study of 100 cases" *J Neurol Neurosurg Psychiatry*, 55 (1992), pp. 181-184. Accessed March 25, 2018

Fortmann-Roe, Scott. "Understanding the Bias-Variance Tradeoff." June 2012. Accessed March 25. http://scott.fortmann-roe.com/docs/BiasVariance.html

Hastie, T., Tibshirani, R., & Friedman, J. H. (2001). *The Elements of Statistical Learning: Data mining, inference, and prediction.* 245-49. Accessed March 13, 2018.

Ng, Andrew. "Advice for Applying Machine Learning." Accessed March 26, 2018. http://cs229.stanford.edu/materials/ML-advice.pdf.

Tysnes, OB. & Storstein, A., "Epidemiology of Parkinson's Disease," *Revue Neurologique* 124, no.8 (February 2017): , Accessed April 01, 2018, https://doi-org.proxy.lib.sfu.ca/10.1007/s00702-017-1686-y

Appendix

Table A1: List of Models

Scikit-learn ML Model Used	Acronym for Following Tables
Support Vector Machine	SVM
Multilayer Perceptron	MLP
Logistic Regression	LR
Random Forest	RF
Nu-Support Vector Machine	Nu-SVM
Decision Tree	DT
K Nearest Neighbors	k-NN
Quadratic Discriminant Analysis	QDA

Table A2: Scores of Individual Models (Unbagged)

	Hold Feature Scores				Latency Feature Scores			
Models	**F-score**	**Acc.**	**Prec.**	**Recall**	**F-score**	**Acc.**	**Prec.**	**Recall**
SVM	0.66	0.60	0.60	0.75	0.70	0.53	0.53	1.00
MLP	0.70	0.67	0.67	0.75	0.72	0.60	0.57	1.00
LR	0.70	0.67	0.67	0.75	0.60	0.47	0.50	0.75
RF	0.70	0.67	0.67	0.75	0.60	0.47	0.50	0.75
Nu-SVM	0.62	0.60	0.62	0.62	0.44	0.33	0.40	0.50
DT	0.70	0.60	0.58	0.88	0.63	0.53	0.54	0.75
k-NN	0.70	0.67	0.67	0.75	0.63	0.53	0.54	0.75
QDA	0.67	0.60	0.60	0.75	0.60	0.47	0.50	0.75

Table A3: Scores of Individual Models (Bagged)

	Hold Feature Scores (Bagged)				Latency Feature Scores (Bagged)			
Models	**F-score**	**Acc.**	**Prec.**	**Recall**	**F-score**	**Acc.**	**Prec.**	**Recall**
SVM	0.63	0.53	0.54	0.75	0.60	0.47	0.50	0.75
MLP	0.70	0.67	0.67	0.75	0.63	0.53	0.54	0.75
LR	0.70	0.67	0.67	0.75	0.60	0.47	0.50	0.75
RF	0.70	0.67	0.67	0.75	0.63	0.53	0.54	0.75
Nu-SVM	0.67	0.60	0.60	0.75	0.57	0.40	0.46	0.75
DT	0.70	0.67	0.67	0.75	0.56	0.47	0.50	0.62
k-NN	0.70	0.67	0.67	0.75	0.53	0.40	0.45	0.62
QDA	0.63	0.53	0.54	0.75	0.60	0.47	0.50	0.75

Table A4: Scores of Ensemble Models

	Scores			
Ensemble Models	**F-score**	**Acc.**	**Prec.**	**Recall**
Mean Probability Classifier	0.84	0.82	0.84	0.84
Voting Classifier	0.83	0.77	0.72	1.00