Piecing the Puzzle: Predicting Parkinson's Disease Through User and Keyboard Data

Project Team 12: Jane Li, Vicki Lu, Michael Shu, Chris Suh, Justin Suh

What is Parkinson's Disease?

Every year, over 200,000 cases of Parkinson's Disease are reported in the United States alone. Parkinson's disease is a disorder of the nervous system in which nerve cell damage causes dopamine levels in the body to drop to insufficient levels. The beginning of Parkinson's Disease is often accompanied with little or no symptoms. You may begin to keep your arms rigid when you walk. Your voice may be a little softer than usual. However, as Parkinson's Disease progresses, it becomes debilitating. Symptoms become so intense that even walking becomes too difficult a task.

Background

Currently, there is no cure for Parkinson's. However, early diagnosis may lead to early intervention which slows its progression. We will use data from a previous study including keyboard data: flight time, latency, and hold time as well as user data: age and gender. The main reason for using this data set is that the original authors claimed 97% accuracy for their prediction model, yet they fail to give a detailed explanation of how they chose their model or how to repeat their findings. Thus, we will bring clarity to this mystery by testing several models against the data: linear regression, logistic regression, SVM, and Naive Bayes. We will then explain where and why these models succeed and fail.

Using this data, we hope to in the future develop a free-to-use prediction tool available as a Chrome extension which logs keyboard use. The idea is that this extension could be installed on loved ones' computers and alert caregivers when Parkinson's is predicted. It is important to note that this is not a diagnosis, but rather a suggestion for the caretaker to take the loved one to the doctor to receive an official diagnosis. This will lead to earlier diagnosis and slowed progression, **giving users more precious time to spend with their families.**

Determining Relevant Factors

The Kaggle data we started with had a lot of information per person, so we took out factors that we wouldn't know if the person themselves didn't know whether they had Parkinson's or not since our predictive model is trying to predict the presence of Parkinson's. These factors mostly included medication and Parkinson's symptom information: 'user_id','hand', 'tremors', 'diagnosis_year', 'updrs', 'impact', 'levadopa', 'da', 'maob', 'other', 'hand_L', 'hand_R', 'hand_S', 'impact_mild', 'impact_medium', 'impact_severe','impact_unknown', 'sided'.

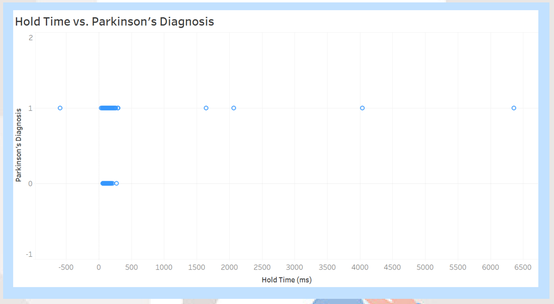
We were left with the following factors:

- gender: Male/Female
- birth_year: Year of birth
- hold_time: Time between press and release for curreny key (ms)
- latency: Time between pressing the previous key and pressing current key (ms)
- flight_time: Time between release of previous key and press of current key (ms)

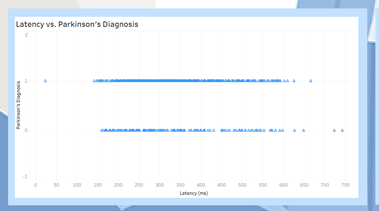
For the remaining factors, "gender, birth_year, hold_time, latency, flight_time", we created different visualizations (based on whether it was categorical or continuous) to examine their correlations with Parkinson's.

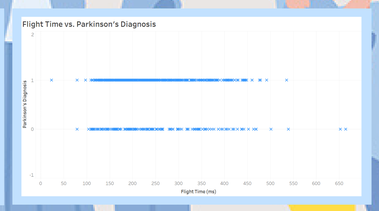
Additionally, if we look at the average values among the no Parkinson's (0) and yes Parkinson's (1) groups, we can see a significant difference in only the average **hold times**.

	Hold time	Latency	Flight time
No PD	83.766	339.461	245.277
Yes Parkinson's	204.363	334.834	238.356
Difference in Avg Times	120.597	4.627	6.921



We can see from these graphs that **there seems to be a correlation between the hold_time of a user and whether or not they have Parkinson's** because most of the lower range of hold times correspond to no Parkinson's, while higher range corresponds to has Parkinson's.





The same is not true for latency and flight time, where both users with Parkinson's and users without show a similar range of hold times.

Data Cleaning

The data that were used for this analysis had to be cleaned and formatted **to ensure that our models worked, and worked well.** The raw data had two components: user data and observation data, where the former had information about the user (eg. date of birth, whether they had parkinson's, what relevant drugs they used, etc.), and the latter had observations from the user's interactions with the keyboard (eg. latency, hold time, direction, etc.). These two components could be joined on the user id, however the size of the dataset made it infeasible to run our models on the joined dataset, as our queries either ran forever or crashed. We got around this bottleneck by iterating over each user, and averaging the observation data by user, per hand, and then running the models on these data.

In addition, once we had the reduced dataset, we needed to clean it, both to deal with missing values, and to ensure that we were maximizing our model performance. For the missing values, we imputed the mean, and **we also normalized each of the factors** by subtracting the mean and dividing by the standard deviation so that all of the factors would be of similar magnitude and none would dominate on any particular model.

In cleaning our data, we used a combination of SQL queries and the pandas data analysis library.

Predictive Modeling

Four predictive models were used: **linear regression**, **logistic regression**, **support vector machine**, and **naive bayes**. We decided to use **5-fold cross validation** in order to train and test our models. Because 5-fold cross validation randomly splits up the data into different "folds" and runs each "fold" as a test set while all the other "folds" are the training set, we can both check for accuracy of the model as well as check to make sure the model is not overfitting the data. For each "fold", we calculated the testing accuracy of the trained model. We then averaged the test accuracy for each "fold" to get our final accuracies for each models.

Logistic regression is an extension of linear regression and is a strong predictive model when there is a binary response variable. It is, however, strongly impacted by outliers, there must be no outliers in the dataset to be accurate. **The dataset contains outliers in some predictor variables (hold time), so these had to be removed before building the model.** The logistic regression is a good probabilistic model, and it can avoid overfitting the data Additionally, logistic regression is easy to interpret as the output of the model can be interpreted as the odds that a given user has Parkinsons. Thus, the logistic regression is a strong model, and is useful in predicting PD .

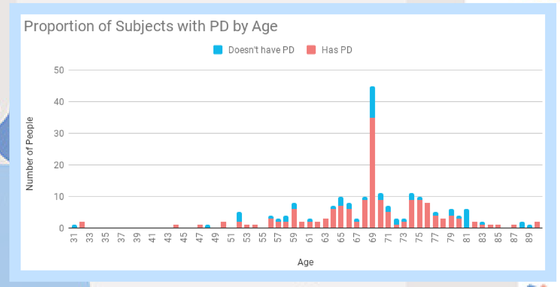
Support Vector Machine (SVM) does not require linear decision boundaries, which is not restrictive and can provide more accurate models. Additionally, **SVM is strong against overfitting**. However, SVM is very sensitive to noise and data points that have predictors far from their predictive class. When parsing the data set, there are many points that are ambiguous, with predictors that seem to suggest the unexpected prediction. Some patients were diagnosed with PD , but exhibited predictors of hold time, latency time, and flight time that are more representative of those without PD than those with PD. Data points like these have the potential to decrease the accuracy of the predictive model.

Naive bayes assumes **conditional independence**, or the assumption that all predictors are independent from each other. This is not true for our dataset, as hold time, latency time, and flight time are all dependent on one another and related to typing patterns. This could potentially decrease the accuracy of the predictive model, but given the simplicity of the model, the model performs surprisingly well in the real world. With that said, this model is still expected to perform worse than more complex models such as SVM.

Results

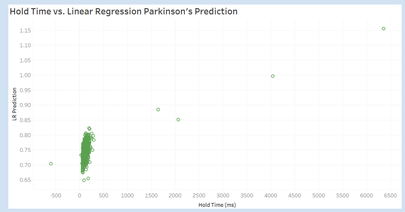
	Linear Regression	Logistic Regression	Support Vector Machine	Naive Bayes
Average Score on K-Fold Test	-0.02406	0.74560	0.75203	0.38625

The highest scoring model was the Support Vector Machine, with Logistic Regression scoring close behind. Linear Regression had the worst score by far of -0.02406. A negative score is valid as it is possible that the model performs worse than a constant model which always predicts the expected value.
However, **accuracy is not always indicative of a better model**. To craft a less naive conclusion of the best model as well as explore why each model was successful or unsuccessful, we delved further into the data, visualizing various factors against the predicted Parkinson's value. One of these factors was hold time. In creating these visualizations, we were able to compare with the original visualization to make the following observations.

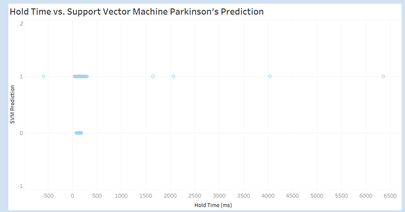


This column distribution of users in each age group with Parkinson's shows how there are greater numbers of people with Parkinson's concentrating around the age range of 70. This makes sense as the older one gets, the higher the chance of developing Parkinson's due to more neuronal loss. The average age onset of Parkinson's is 60 with the disease affecting over 1% of the population over age 60. **We therefore chose to use age as a predictive factor since it correlates with PD.**

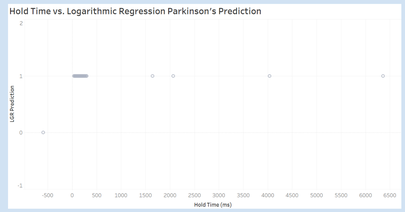
The Models



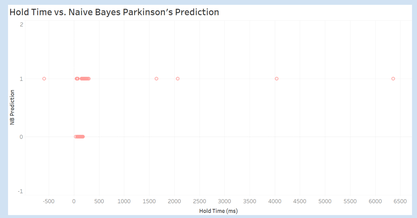
We can see that **linear regression** does not do very well, predicting >50% chance of Parkinson's for all data values.



Support vector machine does better than Logarithmic Regression. However, it miscategorizes several points within the 0 to 500 range.



Logistic regression similarly categorizes the greater hold times correctly, but categorizes all hold times above 0 as indicative of Parkinson's.



Naive Bayes performs the best out of all the models, correctly categorizing every point. However, it is important to note that this is just for hold time. Our conclusion of the best model is considerative of other factors including hold time, latency, gender, and age.

Conclusion

SVM probabilistic model is the most accurate. However, there is **not a large enough difference** between the accuracy of SVM and the logistic regression probabilistic models to warrant choosing the SVM model based solely on accuracy. Furthermore, the logistic regression is easier to interpret than SVM. Thus, although SVM is the most accurate model tested, the **logistic regression model is ultimately is our choice of model** to use due to its ease of interpretation and high accuracy, which is important in the medical context where decisions must be made quickly.

The accuracy of our logistic regression model is only around 75%. This does not match the 97% accuracy found in the previous study of Adams (2017). However, as Adams (2017) does not explicitly state the factors considered in their data modeling, it is possible that they included medication and PD symptom factors in their model. To test this hypothesis, we ran our models with these variables included, and we were able to get an accuracy of 98% using SVM as our model. However, we decided to not use these factors as they are factors only existent after a diagnosis of PD.

While an accuracy of 75% might be considered adequate in other fields, **the medical field involves life-and-death decisions and cannot afford a high risk of misdiagnosis.** Misdiagnoses having potentially very serious ramifications. Any sort of medical diagnosis places an immense amount of financial, physical, and mental strain on the misdiagnosed individual, as well as all of their friends and family. Thus, the repercussions of a misdiagnosis can potentially severely impact one's well-being by creating unnecessary stress.

We acknowledge that the accuracy of our model renders it unable to be used as a diagnostic tool. However, **we may still use this tool to suggest users that should visit their doctors to receive an official diagnosis.** This accomplishes the goal of increasing early diagnosis of Parkinson's Disease, and thus succeeds in the goal of creating a tool that enables early treatment, prevents progression, and **ultimately gives people more time to spend with their loved ones.**

Piecing the Puzzle: Predicting Parkinson's Disease Through User and Keyboard Data

Project Team 12: Jane Li, Vicki Lu, Chris Suh, Justin Suh, Michael Shu

Background and Problem Statement

Every year, over 200,000 cases of Parkinson's Disease (PD) are reported in the United States alone.[1] Parkinson's disease is a disorder of the nervous system in which nerve cell damage causes dopamine levels in the body to drop to insufficient levels. The beginning of PD is often accompanied with little or no symptoms. You may begin to keep your arms rigid when you walk. Your voice may be a little softer than usual. However, as PD progresses, it becomes debilitating. Symptoms become so intense that even walking becomes too difficult a task.[1]

Currently, no cure exists for Parkinson's. However, early diagnosis may lead to early intervention which slows its progression. Thus our goal is to use existing data to create a tool that could aid in the early diagnosis of PD. We will use data from a previously conducted study, Adams (2017), which includes the following keyboard data: flight time, latency, and hold time as well as the following user data: age and gender.[2] The original author claimed 97% accuracy for their prediction model,[2] yet in the study, they do not provide a detailed explanation of the methods used in achieving this. We will attempt to bring clarity to this mystery by testing several models against the data (linear regression, logistic regression, SVM, and Naive Bayes), thoroughly explaining our choices along the way.

Data & Tools

Because PD affects the hand through hand tremors and a general slowing of movement, we can observe these trends in keyboard data. Collecting keyboard data is relative noninvasive and can be collected regularly, thus making it a good candidate for an ongoing PD prediction tool. The dataset comes from Adams (2017), a study in which keyboard data from over 200 subjects both with and without Parkinson's was recorded and analyzed.[2]

The original study has a number of shortcomings that we hope to address. The first shortcoming is a lack of clarity. They do not go into detail to describe the exact steps taken to achieve a predictive model with 97% accuracy. This has led many people to leave comments on the dataset asking for more explanation as they cannot achieve the same values as the authors of the study. Secondly, the original paper lacks explanation for their actions. Thus, it is not easy for readers to glean useful insights as there is no explanation as to why certain methods were chosen. Finally, the authors only used a subset of 53 subjects in their analysis. We believe that this number is too small to create an accurate predictor that will be used on several hundreds of people. As such, we have included as many subjects from the original dataset as possible.

Tools

- We created a collaborative GitLab repo where we shared code and files
- We used a Jupyter Notebook to write code to clean/analyze our data and create our predictive models
- We used Tableau and Google Sheets to explore and visualize our PD data
- We used Python packages to implement creation and analysis of our Naive Bayes, SVM, and regression models including: sklearn, pandas, numpy, scipy

The dataset contains the following data:

User Data	
user_id	10-character code for user identification
birth_year	Year of birth
gender	Male or female
parkinsons	Whether subject has PD [True/False]
tremors	Whether subject exhibits tremors symptom [True/False]
diagnosis_year	Year of PD diagnosis
sided	Whether exhibit exhibits sidedness of PD symptoms [Left/Right/None]
updrs	UPDRS score (method of scoring PD symptoms) [1-5]
impact	Severity of impact of PD symptoms [Mild/Medium/Severe]
levadopa	Whether subject is using Sinemet or similar medication [Yes/No]
da	Whether subject is using dopamine agonist [Yes/No]
maob	Whether subject is using an MOA-B inhibitor [Yes/No]
other	Whether subject is using other PD medications [Yes/No]

Keyboard Data	
user_id	10-character code for user identification
date	[YY-MM-DD]
timestamp	[HH:MM:SS.SSS]
hand	Hand used to press key [L/R]
direction	Direction between previous key to current key [LL/LR/RL/RR/S for space key]
hold_time	Time between press and release of key (ms)
latency	Time between pressing previous key and pressing current key (ms)
flight_time	Time between releasing previous key and pressing current key (ms)

Cleaning Data

The data that was used for this analysis was cleaned and formatted to ensure that our models work effectively. The raw data had two components, user data and observation data, as described above. These two components could be joined on the user id; however the size of the dataset made it infeasible to run our models on the joined dataset, as our queries either took too long to run or crashed. We bypassed this bottleneck by iterating over each user, averaging the observation data by user per hand, and then running the models on these data.

```
try:
    user_data = pd.read_sql(query, conn)
    user_data = user_data.groupby(['user_id', 'hand']).mean()
    all_data.append(user_data)
```

Additionally, once we reduced the dataset, we cleaned it, both to deal with missing values, and to ensure that we were maximizing our model performance. For the missing values, we imputed the mean and normalized each of the factors by subtracting the mean and dividing by the standard deviation, so that all of the factors would be of similar magnitude thus ensuring that none would dominate the predictions on any particular model.

```
# mean imputation
mean_birth = sum(int(r) for r in data['birth_year'] if (r != '----' and r != ' ------' and r != '')) /
             sum(1 for r in data['birth_year'] if (r != '----' and r != ' ------' and r != ''))
data['birth_year'] = [int(r) - mean_birth if r != '----' else 0 for r in data['birth_year']]
data['birth_year'] = data['birth_year'].astype('int32')
```

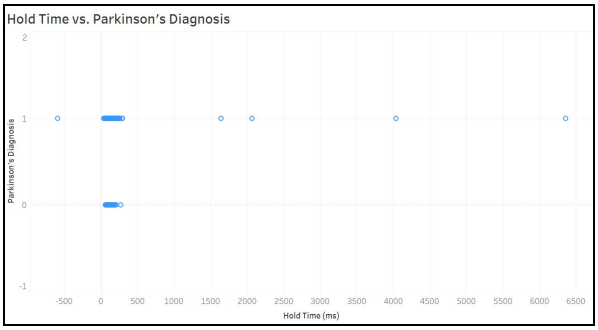
```
# normalizing non-binary variables
for to_norm in ['hold_time', 'latency', 'flight_time', 'birth_year', 'diagnosis_year']:
    m = sum(data[to_norm]) / len(data)
    s = np.std(data[to_norm])
    data[to_norm] = (data[to_norm] - m) / s
```

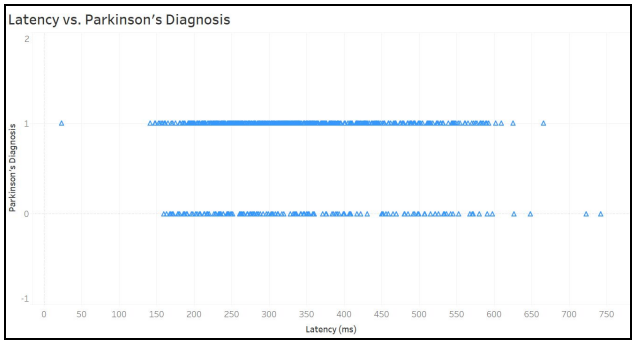
Determining Relevant Factors

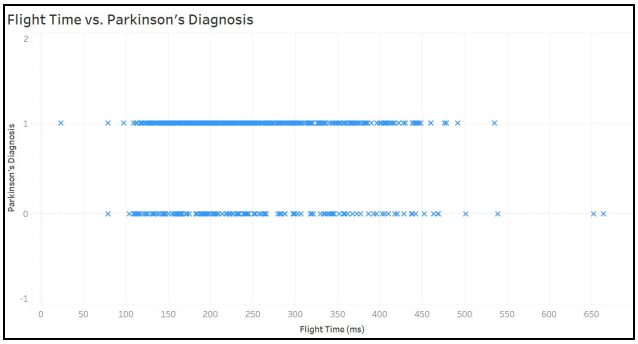
We wanted to evaluate and narrow down which factors would be relevant for our predictive models so we eliminated factors irrelevant to PD and then visualized correlations for the remaining ones. Since our model is meant for users who don't know whether or not they have PD, we wanted to train our prediction only on those factors that would be relevant for diagnosing. For example, a user would not know their UPDRS (unified PD rating scale) score if they didn't even know they had PD. Other factors like these inapplicable ones relate to medication and symptoms. The following are all the factors we dropped due to this reasoning: 'user_id','hand', 'tremors', 'diagnosis_year', 'updrs', 'impact', 'levadopa', 'da', 'maob', 'other', 'hand_L', 'hand_R', 'hand_S', 'impact_mild', 'impact_medium', 'impact_severe', 'impact_unknown', 'sided'.

For the remaining relevant factors, ('gender', 'birth_year', 'hold_time', 'latency', 'flight_time'), we created different visualizations to examine their correlations with PD. Because PD is a binary variable (0 = no PD and 1 = has PD), the visualizations were either comparing two categorical variables or comparing a categorical variable with a continuous one.

For the keyboard related data ('hold_time', 'latency', 'flight_time'), our graphs showed a distribution of points along two horizontal lines, which represented the distribution of keyboard times for users with PD and without PD. We can see that there seems to be a correlation between the 'hold_time'of a user and whether or not they have PD because most of the lower range of hold times correspond to no Parkinson's, while higher range corresponds to has Parkinson's. The same cannot be said about the correlation between PD with 'latency' nor with 'flight_time', where both users with PD and without PD show a similar range of hold times.



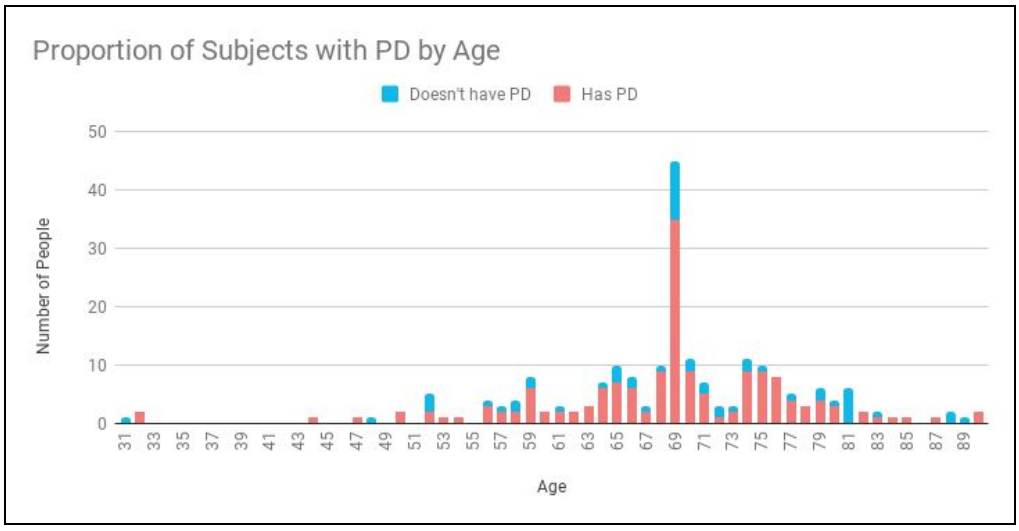




Additionally, if we look at the average values among the no PD (0) and yes PD (1) groups, we can see a significant difference in only the average 'hold_time'. Thus, we chose to drop 'latency' and 'flight_time' as well, only including 'hold_time' as a predictive factor since a higher value seems to correlates with PD.

	Hold time	Latency	Flight time
No PD	83.766	339.461	245.277
Yes Parkinson's	204.363	334.834	238.356
Difference in Avg Times	**120.597**	4.627	6.921

Lastly, we created a visualization for the age/birth year factor. Each column corresponds to an age of users, with a breakdown of how many of them do have PD versus don't have PD. We can see that overall we have a pretty elderly sample, but more importantly, that there are greater numbers of people with PD concentrating around the age of 70. This makes sense as the older one gets, the higher the chance of developing Parkinson's due to more neuronal loss. The average age of onset for PD is 60 with the disease affecting over 1% of the population over age 60[1]. We therefore chose to use age as a predictive factor since it correlates with PD.



Predictive Models - Testing and Training

Four predictive models were used: linear regression, logistic regression, Support Vector Machine (SVM), and Naive Bayes. We decided to use 5-fold cross validation in order to test and train our models. Because 5-fold cross validation randomly splits up the data into different "folds" and runs each "fold" as a test set while all the other "folds" are the training set, we can check for both accuracy of the model as well as check to make sure the model is not overfitting the data. For each "fold", we calculated the testing accuracy of the trained model. We then averaged the test accuracy for each "fold" to get our final accuracies for each models.

In using the predictive models, the strengths and weaknesses of each model were analyzed. When deciding to use a logistic regression model, a linear regression was initially considered. This model, however,

is not ideal when the response variable is a binary variable.[3] Given that our response (predicted) variable is whether a patient has PD (yes or no), a linear regression may not be the best model.

A logistic regression, however, is an extension of linear regression and is a strong predictive model when there is a binary response variable.[3] It is, however, strongly impacted by outliers, so there must be no outliers in the data set for it to be accurate.[3] The dataset contains outliers in some predictor variables (hold time), so these had to be removed before building the model. The logistic regression is a good probabilistic model, and it can avoid overfitting the data.[4] Additionally, logistic regression is easy to interpret as the output of the model can be interpreted as the odds that a given user has PD. Thus, logistic regression is a strong model, and is useful in predicting PD.

The SVM probabilistic model does not require linear decision boundaries, which is not restrictive and can provide more accurate models.[4] Additionally, SVM is strong against overfitting.[4] However, SVM is very sensitive to noise and data points that have predictors far from their predictive class.[5] When parsing the data set, there are many points that are ambiguous, with predictors that seem to suggest the opposite of the diagnosis given. Patients such as patient *NMKZDOICAB* was diagnosed with PD, but exhibited predictors of hold time, latency time, and flight time that are more representative of those without PD than those with PD. Data points like these have the potential to decrease the accuracy of the predictive model.

Finally, Naive Bayes is widely-used probabilistic model that assumes conditional independence, or the assumption that all predictors are independent from each other.[4] This is not true for our dataset, as hold time, latency time, and flight time are all dependent on one another and related to typing patterns. This could potentially decrease the accuracy of the predictive model, but given the simplicity of the model, the model performs surprisingly well in the real world.[4] With that said, this model is still expected to perform worse than more complex models such as SVM.

Results

The accuracies of the models, when taking into account gender, birth year/age, and hold time, are -0.00617 for linear regression, 0.74550 for logistic regression, 0.75194 for SVM, and 0.39417 for Naive Bayes. A negative score for linear regression is valid as it is possible that the linear regression model performs worse than a constant model where the expected value is always predicted. Based on this low accuracy and the fact that our predictive variable is categorical rather than continuous, we opted to immediately throw out linear regression as a viable model. However, for the other models, we decided to keep them in the running for "best" model. While the accuracies of these are much lower than those found in Adams (2017), the paper does not specify the methods of their predictive model, so it is possible that the model in the paper uses the medication and PD symptom variables in their model.[2] To test this hypothesis, we ran our models with these variables included, and we were able to get an accuracy of 98.5% using SVM as our model. However, as previously stated, we ultimately decided not to include these variables in our final models as they are PD treatments and symptoms, which are conditional upon PD.

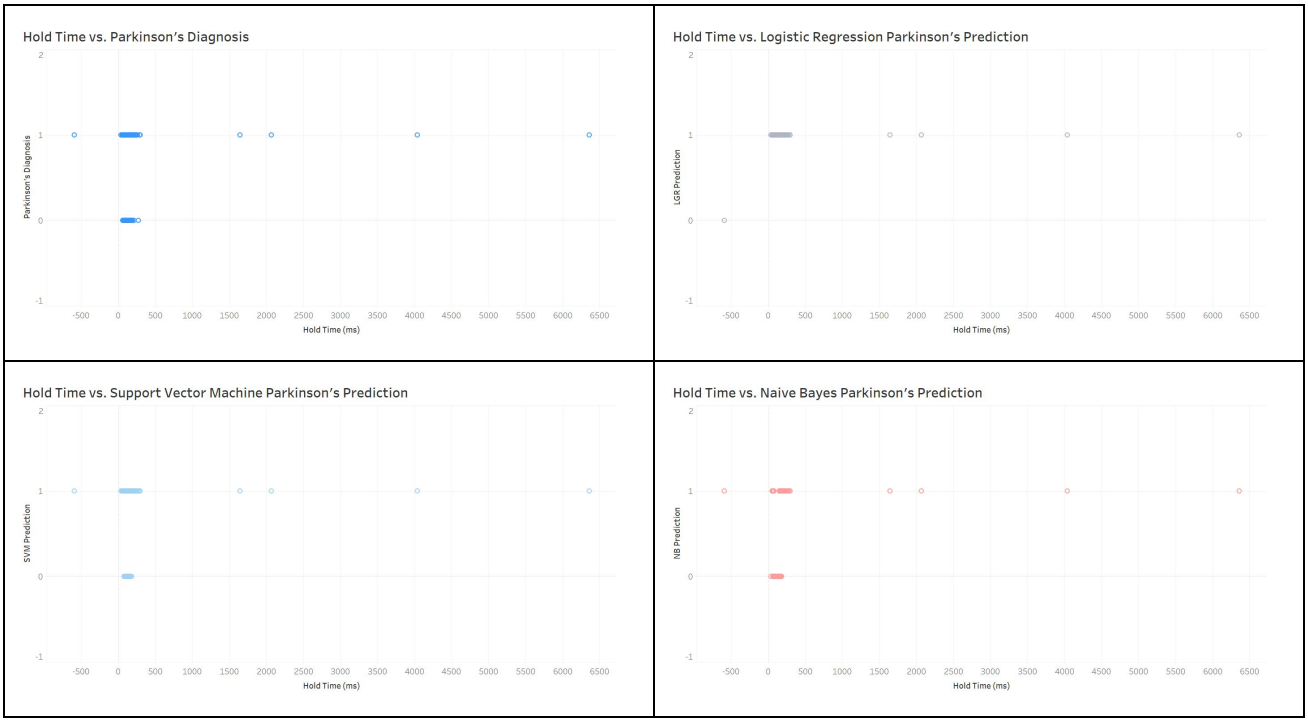
	Linear Regression	Logistic Regression	Support Vector Machine	Naive Bayes
Average Score on K-Fold Test	-0.00617	0.74550	0.75194	0.39417

While k-fold test scores can be used to evaluate the success of a model, it is a naive way to do so. Thus, we further delve into various aspects of the models before concluding on a "best" model for the data.

Comparing Models to Real Diagnosis

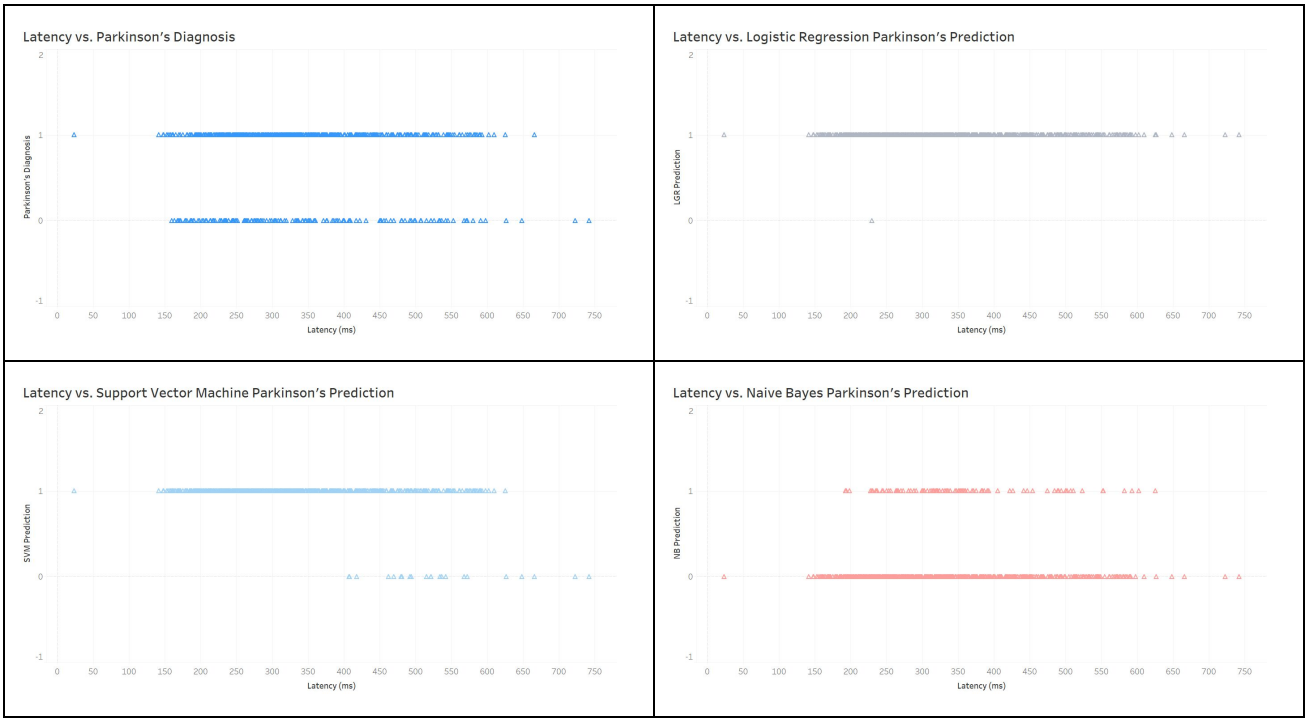
Hold Time

Based on these graphs, we can see that SVM and Naive Bayes models perform better than Logistic regression as they correctly predict PD in a greater number of points. Logistic regression mistakenly predicts PD in points with lower hold times. In fact, it appears that the logistic regression predicts PD for all hold times > 0.



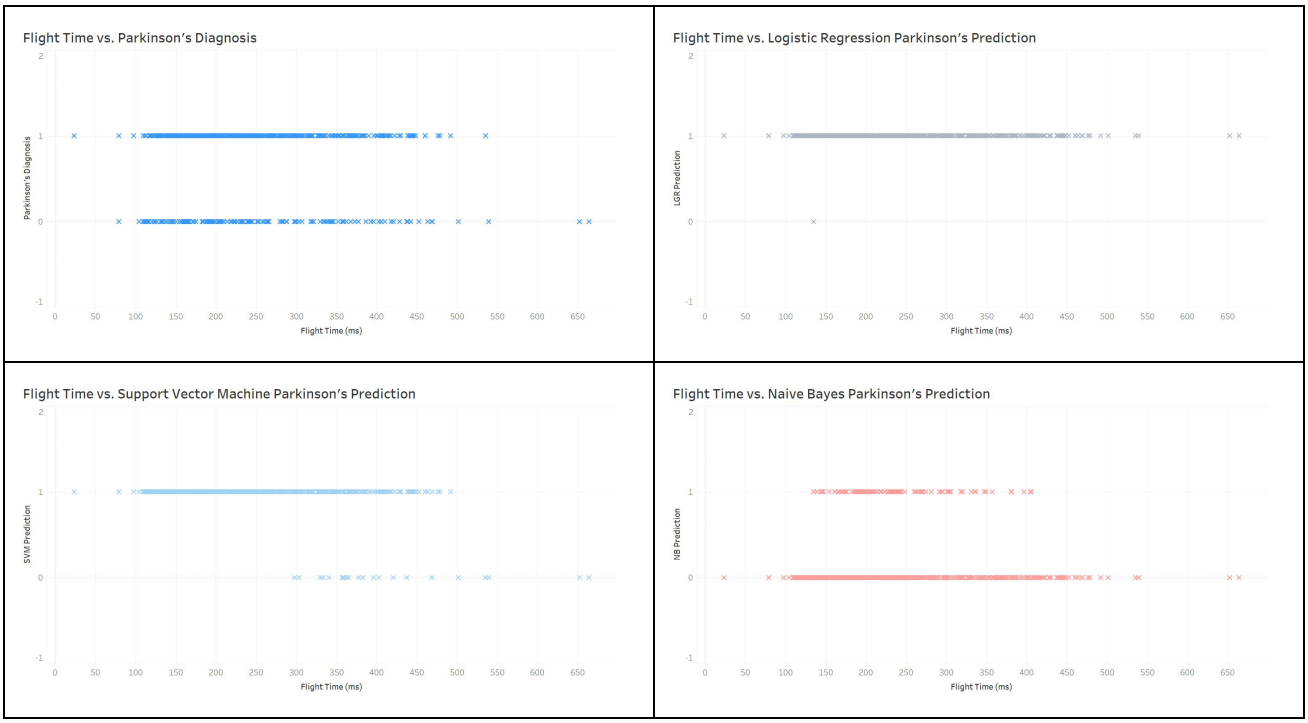
Latency

Here, logistic regression follows the same pattern of over predicting PD. SVM performs less ideally here, mispredicting lower latencies to be indicative of PD. Naive Bayes underpredicts PD.



Flight Time

Flight time follows similar patterns as latency. This is expected as these values measure almost the same thing. Logistic regression continues to overpredict, SVM mispredicts in the lower values, and Naive Bayes underpredicts overall.



True Positives

As an additional check, we calculated the true positive score by dividing the intersection of the number of subjects who have PD and were predicted by the model to have PD by the total number of subjects who have PD. Having a high true positive score is useful for our purposes as we aim to create a tool that will suggest to users when to visit their doctors for a professional diagnosis. The repercussions of over predicting PD is that users may have to spend some time in the doctor's office only to receive a negative diagnosis. However, this is much better than the alternative of users having PD but thinking that they do not have it because of the prediction of our model.

	SVM	LGR	NB
True Positive Count	465	469	89
Total # with PD	470	470	470
True Positive Score	**0.9893617021**	**0.9978723404**	**0.1893617021**

Conclusion

From the results, we conclude that the SVM is the "best" model out of the models chosen. In terms of k-fold test score, SVM had the highest score. SVM additionally had the second highest true positive score. Logistic regression, with a marginally lower accuracy score and a higher true positive score, was a close contender with SVM. However, even though logistic regression gives us a higher true positive score, we saw through visualizations that this is most likely because of how our logistic regression model over predicts PD. In fact, it almost always predicts PD which is not helpful as we do not want our prediction tool to tell everyone

that they have PD. Thus, we ultimately decided to proceed with the SVM model, the most well-balanced of the four, as our final model.

With our SVM model, we can be reasonably confident that it accurately predicts (~75% accuracy) whether a user has PD from their user information and keystroke data. However, the accuracy of our SVM model is only around 75%. So at this stage, we must acknowledge that the accuracy of our model renders it unable to be used as a diagnostic tool. However, we may still use this tool to suggest that users visit their doctors to receive an official diagnosis. This accomplishes the goal of increasing early diagnosis of Parkinson's Disease, and thus succeeds in the goal of creating a tool that enables early treatment.

Using our findings, we hope to in the future develop a free-to-use prediction tool available as a Chrome extension which logs keyboard use. The idea is that this extension could be installed on loved ones' computers and alert caregivers when PD is predicted. It is important to note that this is not a diagnosis, but rather a suggestion for the caretaker to take the loved one to the doctor to receive an official diagnosis. This will lead to earlier diagnosis and slowed progression, giving users more precious time to spend with their families.

References

1. The Parkinson's Foundation, https://parkinson.org/
2. Adams, W, "*High-accuracy detection of early Parkinson's Disease using multiple characteristics of finger movement while typing",* https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0188226, 2017
3. "Binary Logistic Regression", https://www.statisticssolutions.com/binary-logistic-regression/, 2019
4. "Modern Machine Learning Algorithms: Strengths and Weaknesses", https://elitedatascience.com/machine-learning-algorithms, 2017
5. Jesse Johnson, "Linear Separation and Support Vector Machines", https://shapeofdata.wordpress.com/2013/05/14/linear-separation-and-support-vector-machines/, 2013