Diagnosis from the keyboard

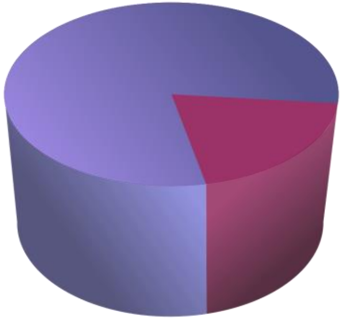
Can we detect early stages of Parkinson's disease from the typing pattern of people?

By Swee Loke and Daniel Silvestre

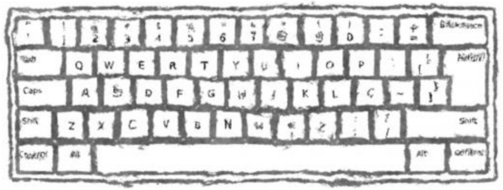
Aug 2019

Inspiration

While this is a difficult dataset to work with, there is a rich trove of information. It is a great set to **practice preprocessing**, attempt to replicate the results of the article, or do **your own analysis of keystroke data**.



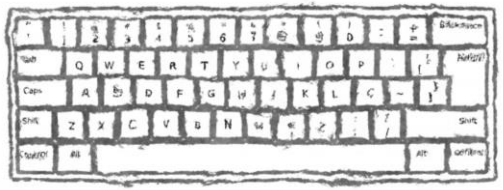
Source: https://www.kaggle.com/valkling/tappy-keystroke-data-with-parkinsons-patients



Tappy Files

Each file contains comma separated keystroke data for one month for a particular user. The filename comprises the 10 character code (matching the user details file) and the YYMM of the data. The fields are:

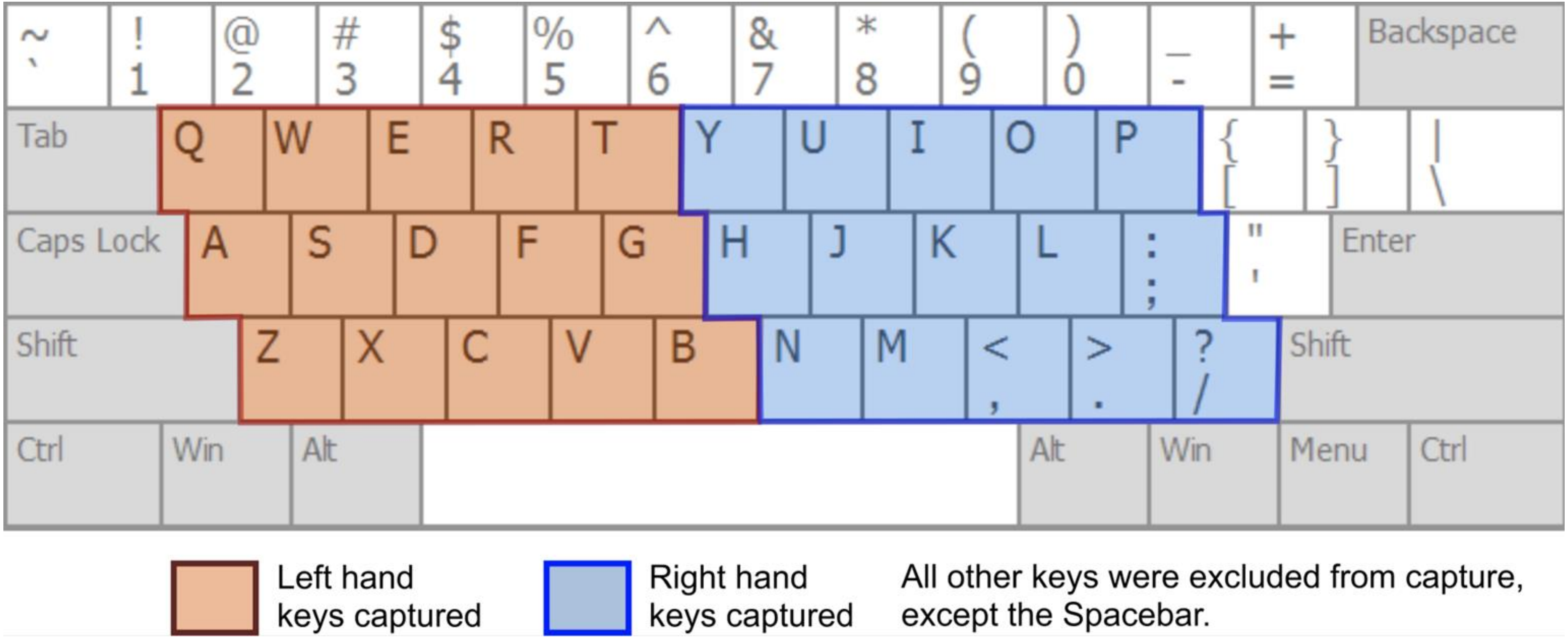
- **UserKey**: 10 character code for that user
- **Date**: YYMMDD
- **Timestamp**: HH:MM:SS.SSS
- **Hand**: L or R key pressed
- **Hold time**: Time between press and release for current key mmmm.m milliseconds
- **Direction**: Previous to current LL, LR, RL, RR (and S for a space key)
- **Latency time**: Time between pressing the previous key and pressing current key. Milliseconds
- **Flight time**: Time between release of previous key and press of current key. Milliseconds



Tappy Files

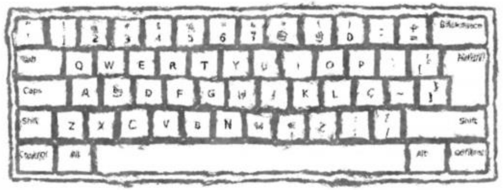
Each file contains comma separated keystroke data for one month for a particular user. The filename comprises the 10 character code (matching the user details file) and the YYMM of the data. The fields are:

- **UserKey**: 10 character code for that user
- **Date**: YYMMDD
- **Timestamp**: HH:MM:SS.SSS
- **Hand**: L or R key pressed
- **Hold time**: Time between press and release for current key mmmm.m milliseconds
- **Direction**: Previous to current LL, LR, RL, RR (and S for a space key)
- **Latency time**: Time between pressing the previous key and pressing current key. Milliseconds
- **Flight time**: Time between release of previous key and press of current key. Milliseconds



All other keys were excluded from capture, except the Spacebar.

Source: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0188226#sec008

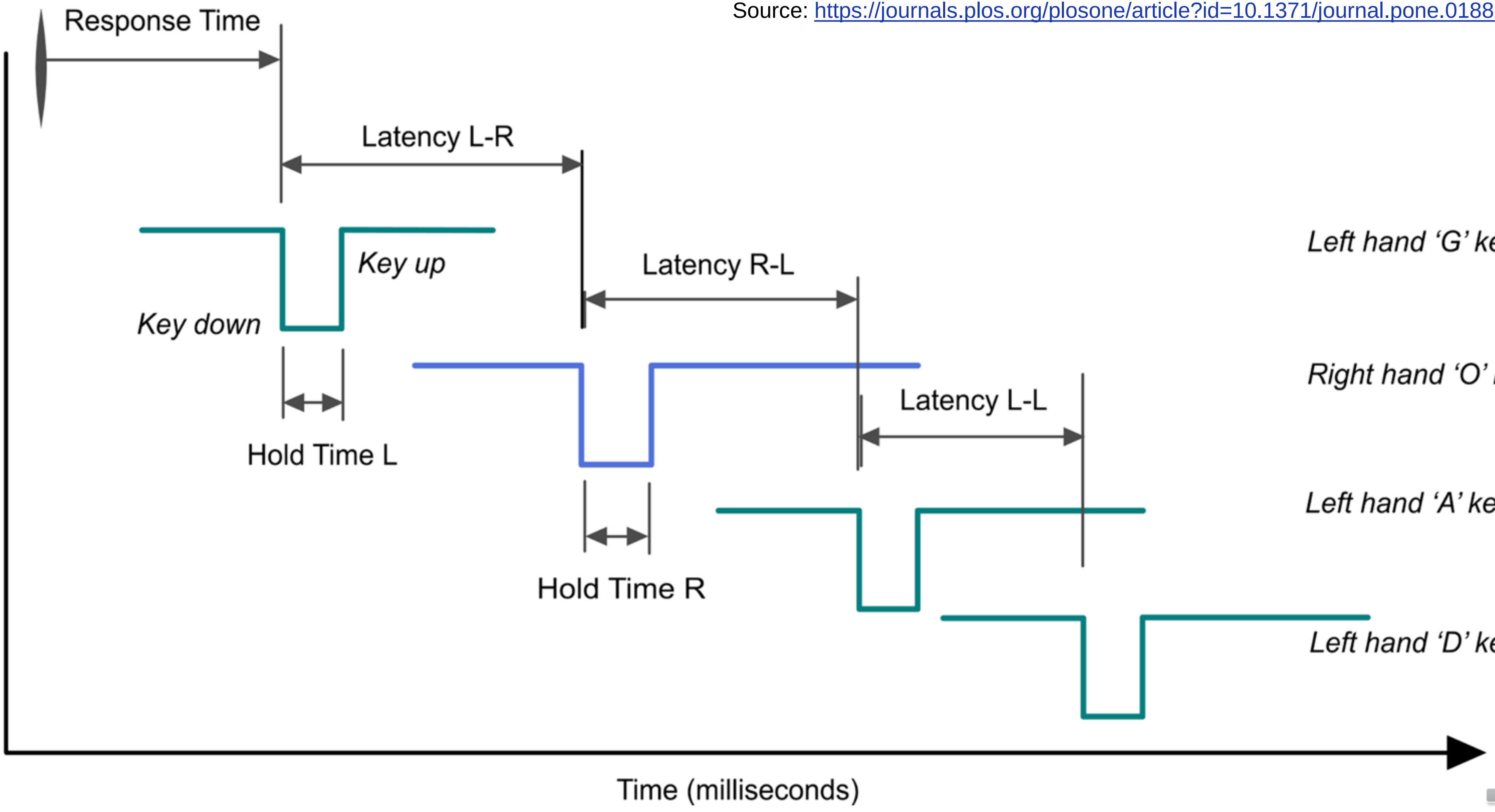


Tappy Files

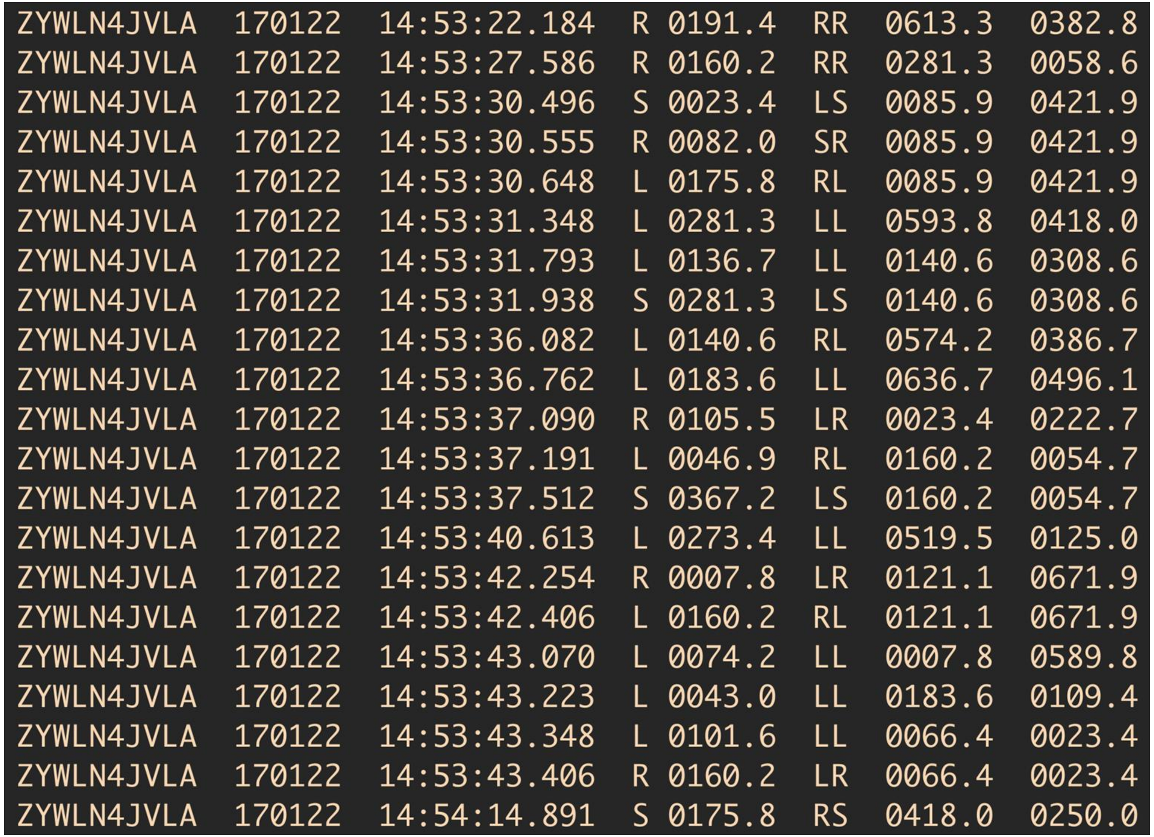
Each file contains comma separated keystroke data for one month for a particular user. The filename comprises the 10 character code (matching the user details file) and the YYMM of the data. The fields are:

- **UserKey**: 10 character code for that user
- **Date**: YYMMDD
- **Timestamp**: HH:MM:SS.SSS
- **Hand**: L or R key pressed
- **Hold time**: Time between press and release for current key mmmm.m milliseconds
- **Direction**: Previous to current LL, LR, RL, RR (and S for a space key)
- **Latency time**: Time between pressing the previous key and pressing current key. Milliseconds
- **Flight time**: Time between release of previous key and press of current key. Milliseconds

Source: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0188226



ZYWLN4JVLA_1701.txt



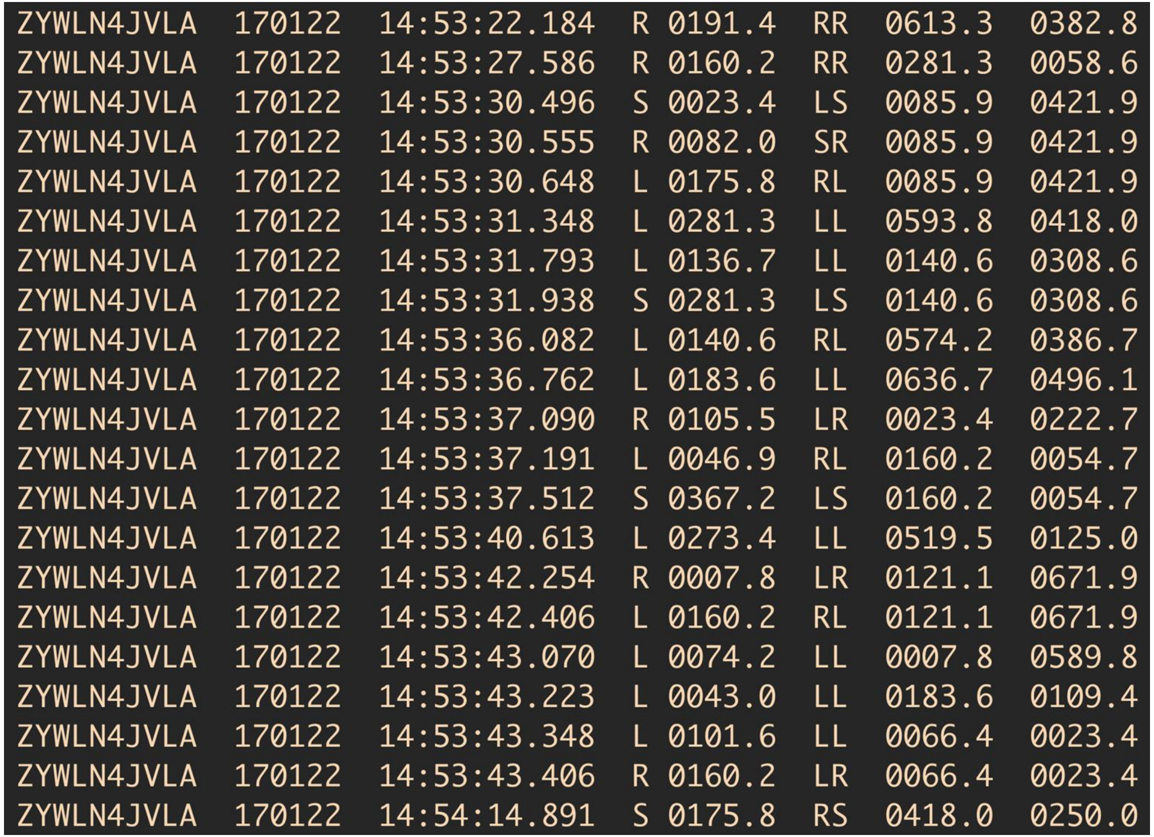
```
ZYWLN4JVLA  170122  14:53:22.184  R 0191.4  RR  0613.3  0382.8
ZYWLN4JVLA  170122  14:53:27.586  R 0160.2  RR  0281.3  0058.6
ZYWLN4JVLA  170122  14:53:30.496  S 0023.4  LS  0085.9  0421.9
ZYWLN4JVLA  170122  14:53:30.555  R 0082.0  SR  0085.9  0421.9
ZYWLN4JVLA  170122  14:53:30.648  L 0175.8  RL  0085.9  0421.9
ZYWLN4JVLA  170122  14:53:31.348  L 0281.3  LL  0593.8  0418.0
ZYWLN4JVLA  170122  14:53:31.793  L 0136.7  LL  0140.6  0308.6
ZYWLN4JVLA  170122  14:53:31.938  S 0281.3  LS  0140.6  0308.6
ZYWLN4JVLA  170122  14:53:36.082  L 0140.6  RL  0574.2  0386.7
ZYWLN4JVLA  170122  14:53:36.762  L 0183.6  LL  0636.7  0496.1
ZYWLN4JVLA  170122  14:53:37.090  R 0105.5  LR  0023.4  0222.7
ZYWLN4JVLA  170122  14:53:37.191  L 0046.9  RL  0160.2  0054.7
ZYWLN4JVLA  170122  14:53:37.512  S 0367.2  LS  0160.2  0054.7
ZYWLN4JVLA  170122  14:53:40.613  L 0273.4  LL  0519.5  0125.0
ZYWLN4JVLA  170122  14:53:42.254  R 0007.8  LR  0121.1  0671.9
ZYWLN4JVLA  170122  14:53:42.406  L 0160.2  RL  0121.1  0671.9
ZYWLN4JVLA  170122  14:53:43.070  L 0074.2  LL  0007.8  0589.8
ZYWLN4JVLA  170122  14:53:43.223  L 0043.0  LL  0183.6  0109.4
ZYWLN4JVLA  170122  14:53:43.348  L 0101.6  LL  0066.4  0023.4
ZYWLN4JVLA  170122  14:53:43.406  R 0160.2  LR  0066.4  0023.4
ZYWLN4JVLA  170122  14:54:14.891  S 0175.8  RS  0418.0  0250.0
```

> **One file for each subject in each month**

- Total of 622 files to load

- .....

ZYWLN4JVLA_1701.txt



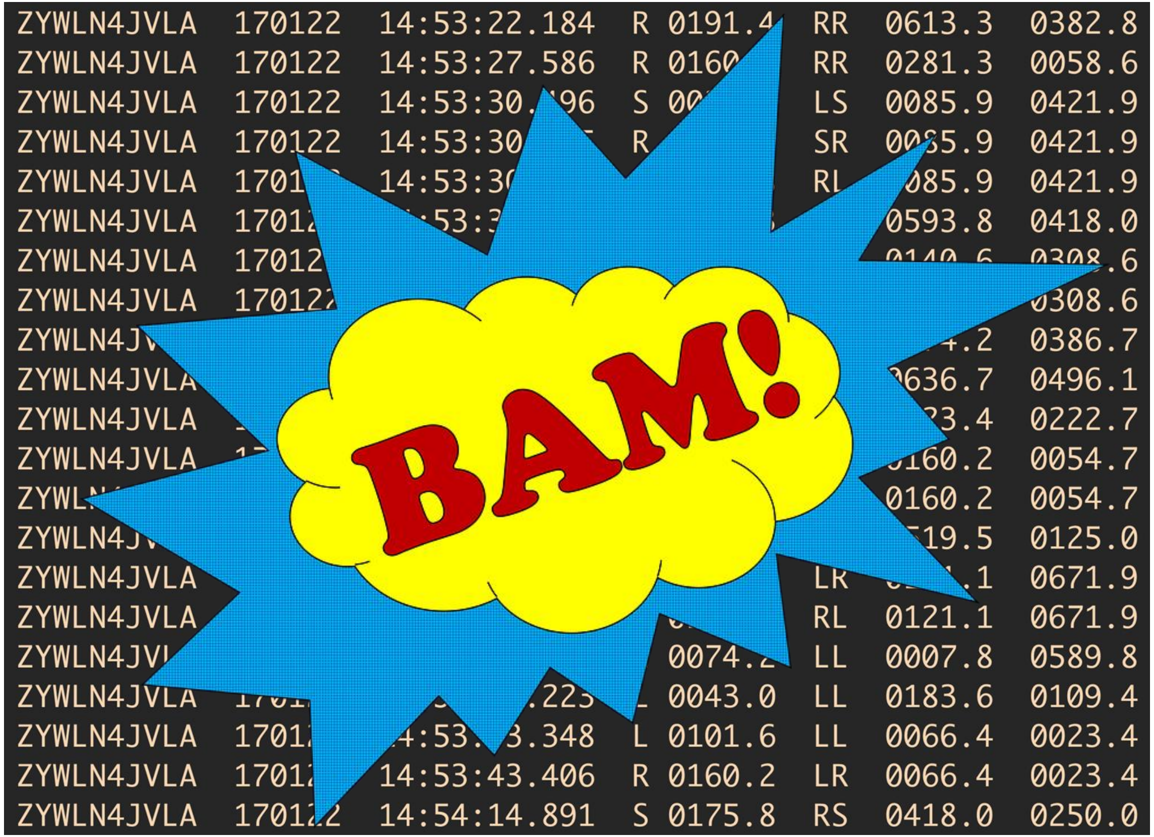
```
ZYWLN4JVLA  170122  14:53:22.184  R 0191.4  RR  0613.3  0382.8
ZYWLN4JVLA  170122  14:53:27.586  R 0160.2  RR  0281.3  0058.6
ZYWLN4JVLA  170122  14:53:30.496  S 0023.4  LS  0085.9  0421.9
ZYWLN4JVLA  170122  14:53:30.555  R 0082.0  SR  0085.9  0421.9
ZYWLN4JVLA  170122  14:53:30.648  L 0175.8  RL  0085.9  0421.9
ZYWLN4JVLA  170122  14:53:31.348  L 0281.3  LL  0593.8  0418.0
ZYWLN4JVLA  170122  14:53:31.793  L 0136.7  LL  0140.6  0308.6
ZYWLN4JVLA  170122  14:53:31.938  S 0281.3  LS  0140.6  0308.6
ZYWLN4JVLA  170122  14:53:36.082  L 0140.6  RL  0574.2  0386.7
ZYWLN4JVLA  170122  14:53:36.762  L 0183.6  LL  0636.7  0496.1
ZYWLN4JVLA  170122  14:53:37.090  R 0105.5  LR  0023.4  0222.7
ZYWLN4JVLA  170122  14:53:37.191  L 0046.9  RL  0160.2  0054.7
ZYWLN4JVLA  170122  14:53:37.512  S 0367.2  LS  0160.2  0054.7
ZYWLN4JVLA  170122  14:53:40.613  L 0273.4  LL  0519.5  0125.0
ZYWLN4JVLA  170122  14:53:42.254  R 0007.8  LR  0121.1  0671.9
ZYWLN4JVLA  170122  14:53:42.406  L 0160.2  RL  0121.1  0671.9
ZYWLN4JVLA  170122  14:53:43.070  L 0074.2  LL  0007.8  0589.8
ZYWLN4JVLA  170122  14:53:43.223  L 0043.0  LL  0183.6  0109.4
ZYWLN4JVLA  170122  14:53:43.348  L 0101.6  LL  0066.4  0023.4
ZYWLN4JVLA  170122  14:53:43.406  R 0160.2  LR  0066.4  0023.4
ZYWLN4JVLA  170122  14:54:14.891  S 0175.8  RS  0418.0  0250.0
```

- One file for each subject in each month

> **Total of 622 files to load**

- .....

ZYWLN4JVLA_1701.txt



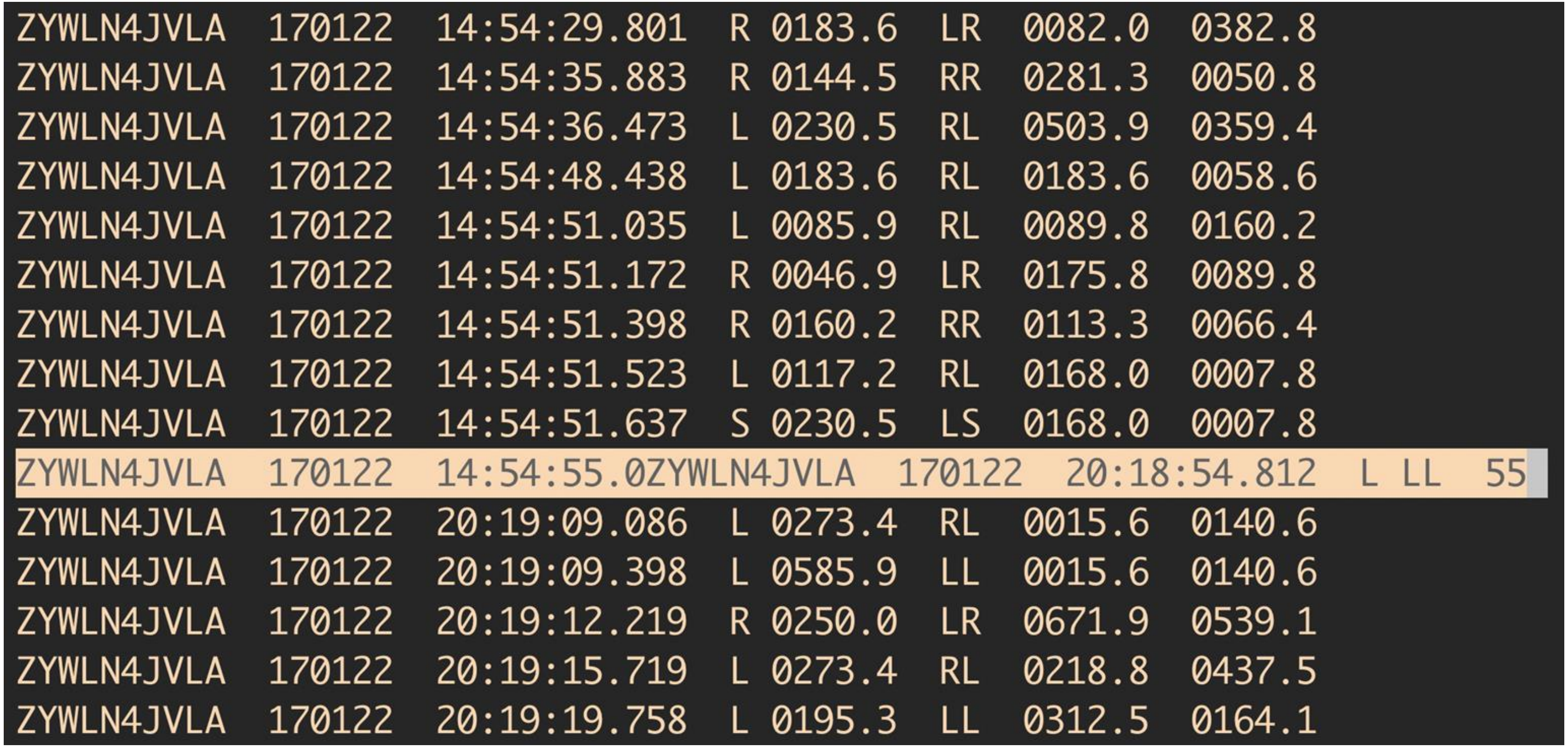
```
ZYWLN4JVLA  170122  14:53:22.184  R 0191.4  RR  0613.3  0382.8
ZYWLN4JVLA  170122  14:53:27.586  R 0160    RR  0281.3  0058.6
ZYWLN4JVLA  170122  14:53:30.             S 00      LS  0085.9  0421.9
ZYWLN4JVLA  170122  14:53:30        R         SR  00 5.9  0421.9
ZYWLN4JVLA  1701    14:53:30                  RL   085.9  0421.9
ZYWLN4JVLA  1701     :53:3                         0593.8  0418.0
ZYWLN4JVLA  17012                                   0140.6  0308.6
ZYWLN4JVLA  17012                                          0308.6
ZYWLN4J                                                 .2  0386.7
ZYWLN4JVLA                                         0636.7  0496.1
ZYWLN4JVLA                                            3.4  0222.7
ZYWLN4JVLA                                          160.2  0054.7
ZYWL                                               0160.2  0054.7
ZYWLN4J                                              19.5  0125.0
ZYWLN4JVLA                                    LR       .1  0671.9
ZYWLN4JVLA                                    RL  0121.1  0671.9
ZYWLN4JV                            0074.2   LL  0007.8  0589.8
ZYWLN4JVLA                   .223     0043.0  LL  0183.6  0109.4
ZYWLN4JVLA  1701      :53:  3.348  L 0101.6  LL  0066.4  0023.4
ZYWLN4JVLA  1701    14:53:43.406  R 0160.2  LR  0066.4  0023.4
ZYWLN4JVLA  170122  14:54:14.891  S 0175.8  RS  0418.0  0250.0
```

- One file for each subject in each month

- Total of 622 files to load

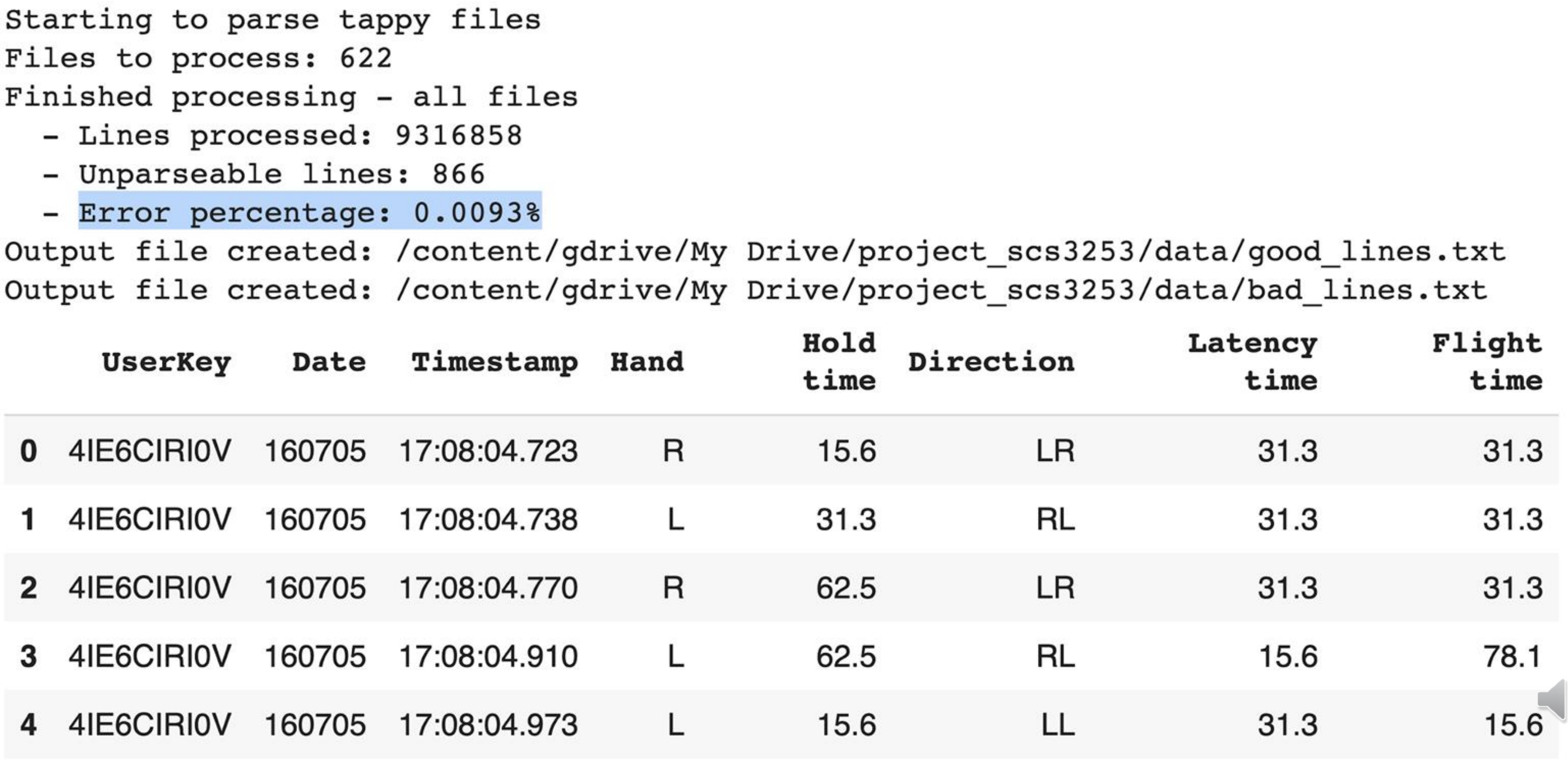
> We cannot directly load as a dataframe

FTP or recording app is truncating...



```
ZYWLN4JVLA  170122  14:54:29.801  R 0183.6  LR  0082.0  0382.8
ZYWLN4JVLA  170122  14:54:35.883  R 0144.5  RR  0281.3  0050.8
ZYWLN4JVLA  170122  14:54:36.473  L 0230.5  RL  0503.9  0359.4
ZYWLN4JVLA  170122  14:54:48.438  L 0183.6  RL  0183.6  0058.6
ZYWLN4JVLA  170122  14:54:51.035  L 0085.9  RL  0089.8  0160.2
ZYWLN4JVLA  170122  14:54:51.172  R 0046.9  LR  0175.8  0089.8
ZYWLN4JVLA  170122  14:54:51.398  R 0160.2  RR  0113.3  0066.4
ZYWLN4JVLA  170122  14:54:51.523  L 0117.2  RL  0168.0  0007.8
ZYWLN4JVLA  170122  14:54:51.637  S 0230.5  LS  0168.0  0007.8
ZYWLN4JVLA  170122  14:54:55.0ZYWLN4JVLA  170122  20:18:54.812  L LL  55
ZYWLN4JVLA  170122  20:19:09.086  L 0273.4  RL  0015.6  0140.6
ZYWLN4JVLA  170122  20:19:09.398  L 0585.9  LL  0015.6  0140.6
ZYWLN4JVLA  170122  20:19:12.219  R 0250.0  LR  0671.9  0539.1
ZYWLN4JVLA  170122  20:19:15.719  L 0273.4  RL  0218.8  0437.5
ZYWLN4JVLA  170122  20:19:19.758  L 0195.3  LL  0312.5  0164.1
```

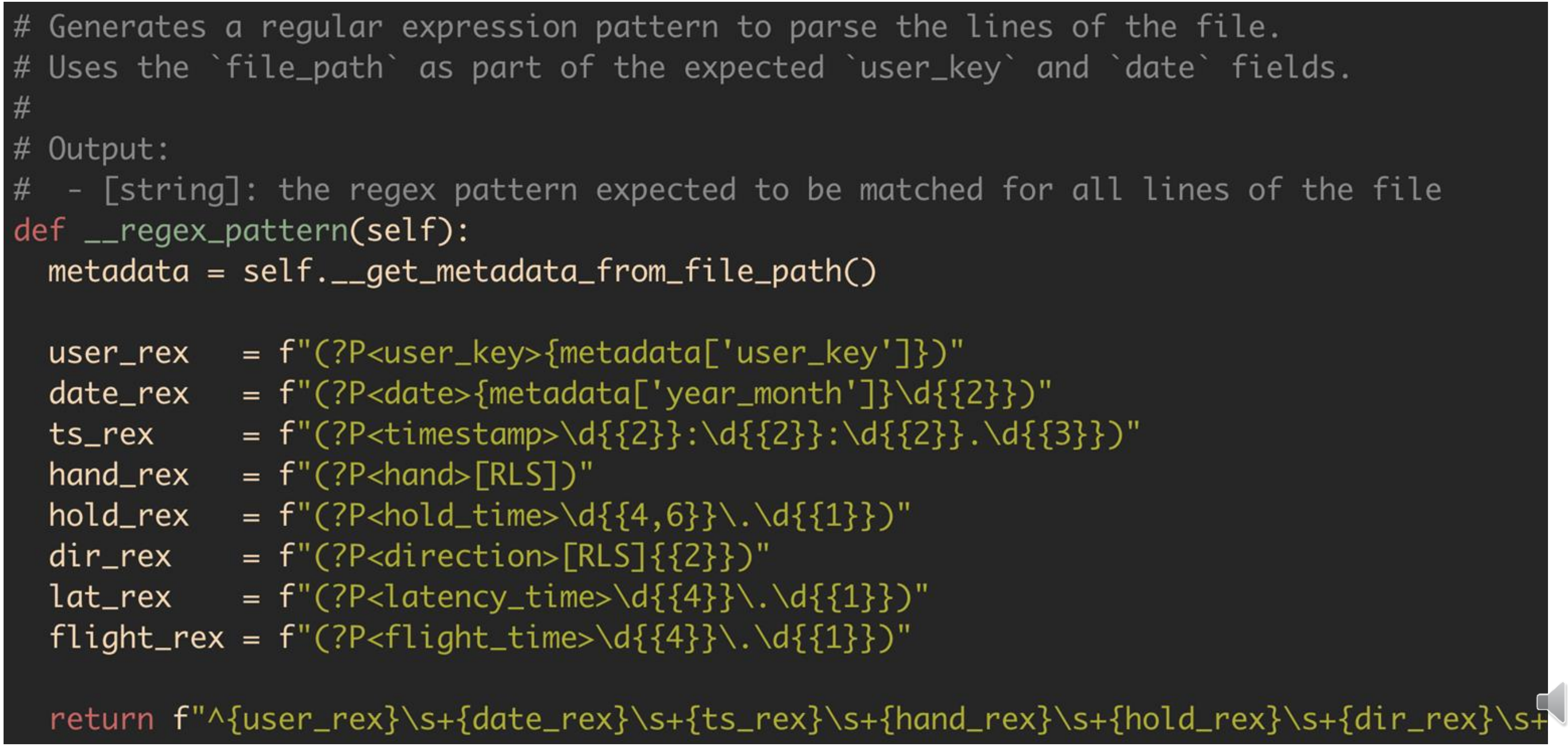
Regular expression for the win



```
Starting to parse tappy files
Files to process: 622
Finished processing - all files
  - Lines processed: 9316858
  - Unparseable lines: 866
  - Error percentage: 0.0093%
Output file created: /content/gdrive/My Drive/project_scs3253/data/good_lines.txt
Output file created: /content/gdrive/My Drive/project_scs3253/data/bad_lines.txt
```

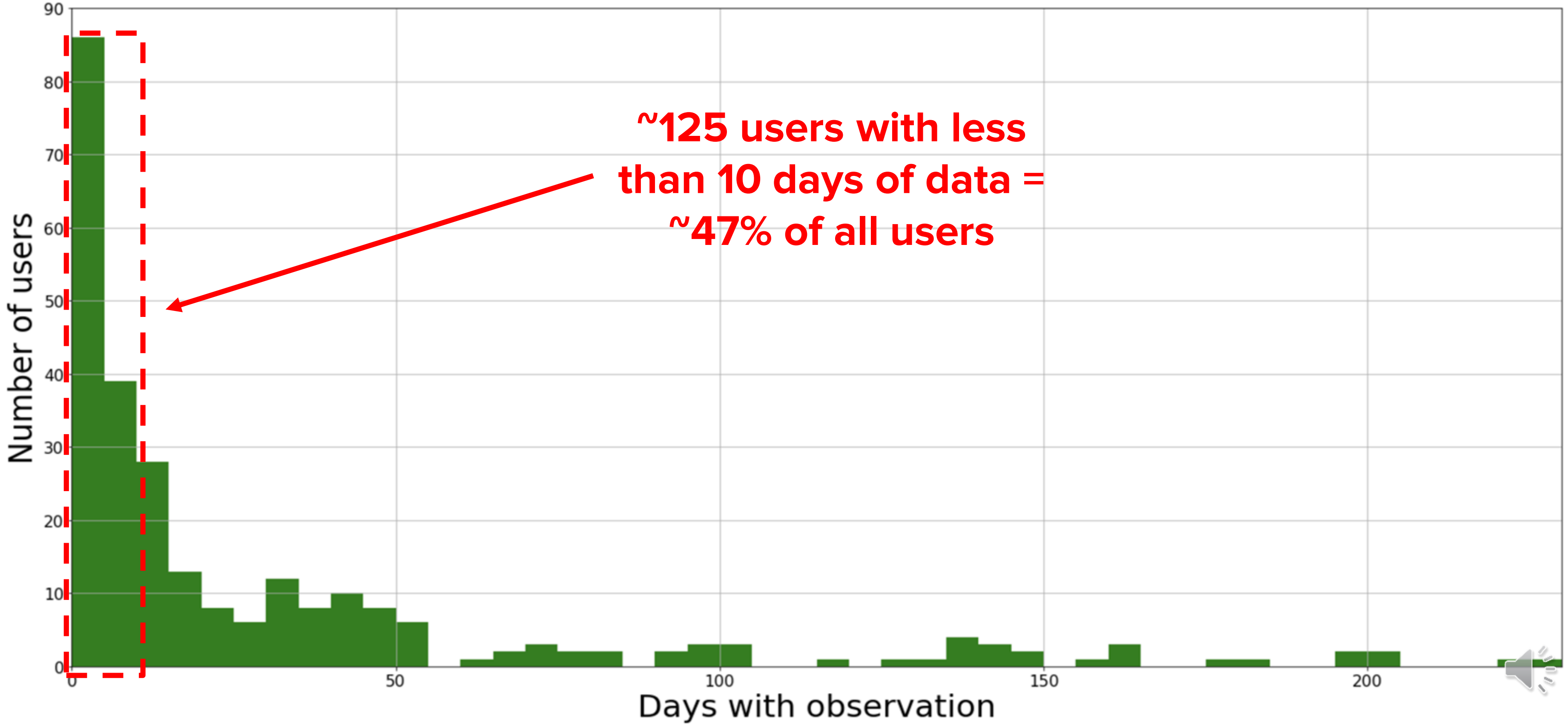
	UserKey	Date	Timestamp	Hand	Hold time	Direction	Latency time	Flight time
0	4IE6CIRI0V	160705	17:08:04.723	R	15.6	LR	31.3	31.3
1	4IE6CIRI0V	160705	17:08:04.738	L	31.3	RL	31.3	31.3
2	4IE6CIRI0V	160705	17:08:04.770	R	62.5	LR	31.3	31.3
3	4IE6CIRI0V	160705	17:08:04.910	L	62.5	RL	15.6	78.1
4	4IE6CIRI0V	160705	17:08:04.973	L	15.6	LL	31.3	15.6

Regular expression??



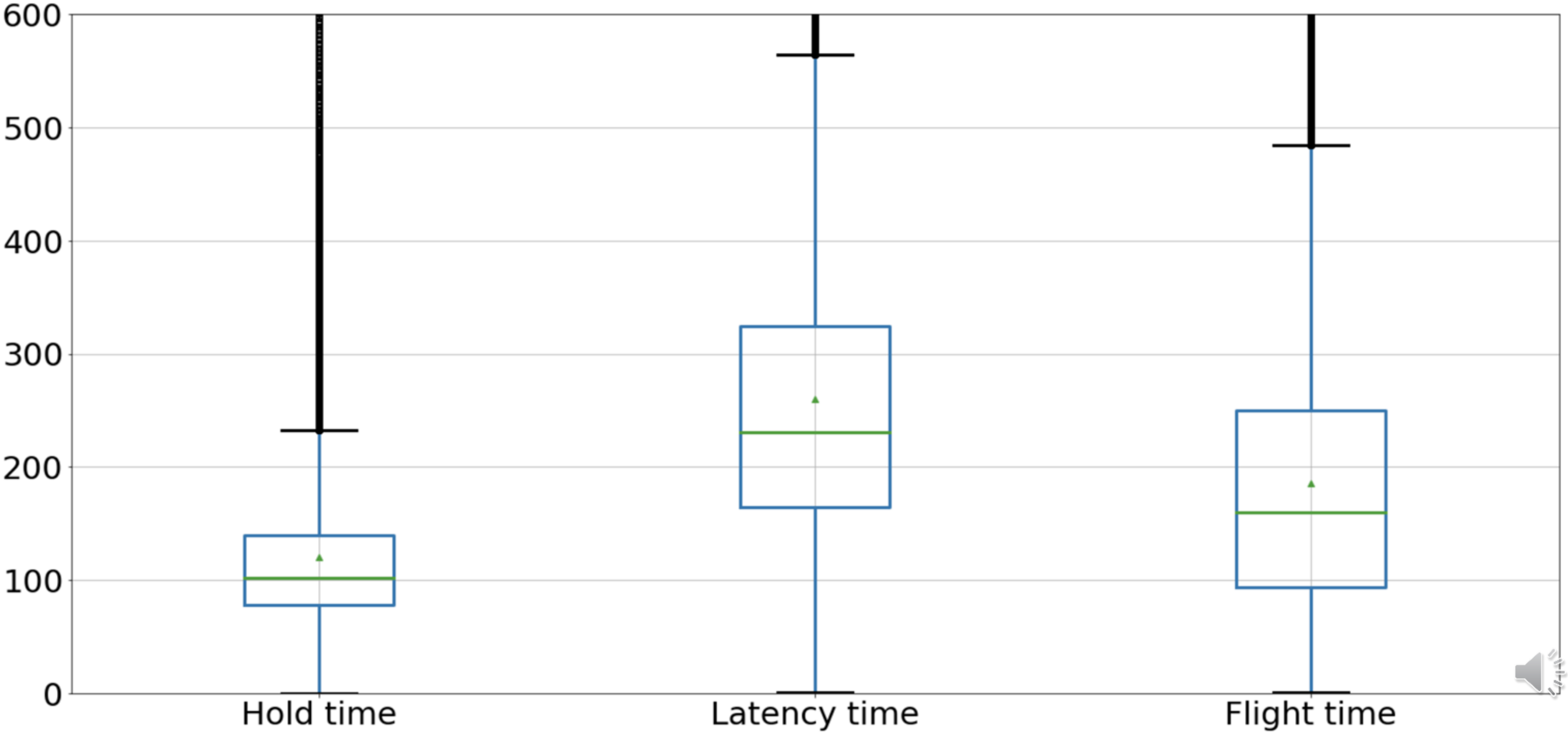
```
# Generates a regular expression pattern to parse the lines of the file.
# Uses the `file_path` as part of the expected `user_key` and `date` fields.
#
# Output:
#   - [string]: the regex pattern expected to be matched for all lines of the file
def __regex_pattern(self):
  metadata = self.__get_metadata_from_file_path()

  user_rex    = f"(?P<user_key>{metadata['user_key']})"
  date_rex    = f"(?P<date>{metadata['year_month']}\d{{2}})"
  ts_rex      = f"(?P<timestamp>\d{{2}}:\d{{2}}:\d{{2}}.\d{{3}})"
  hand_rex    = f"(?P<hand>[RLS])"
  hold_rex    = f"(?P<hold_time>\d{{4,6}}\.\d{{1}})"
  dir_rex     = f"(?P<direction>[RLS]{{2}})"
  lat_rex     = f"(?P<latency_time>\d{{4}}\.\d{{1}})"
  flight_rex = f"(?P<flight_time>\d{{4}}\.\d{{1}})"

  return f"^{user_rex}\s+{date_rex}\s+{ts_rex}\s+{hand_rex}\s+{hold_rex}\s+{dir_rex}\s+
```

Exploring tappy dataset

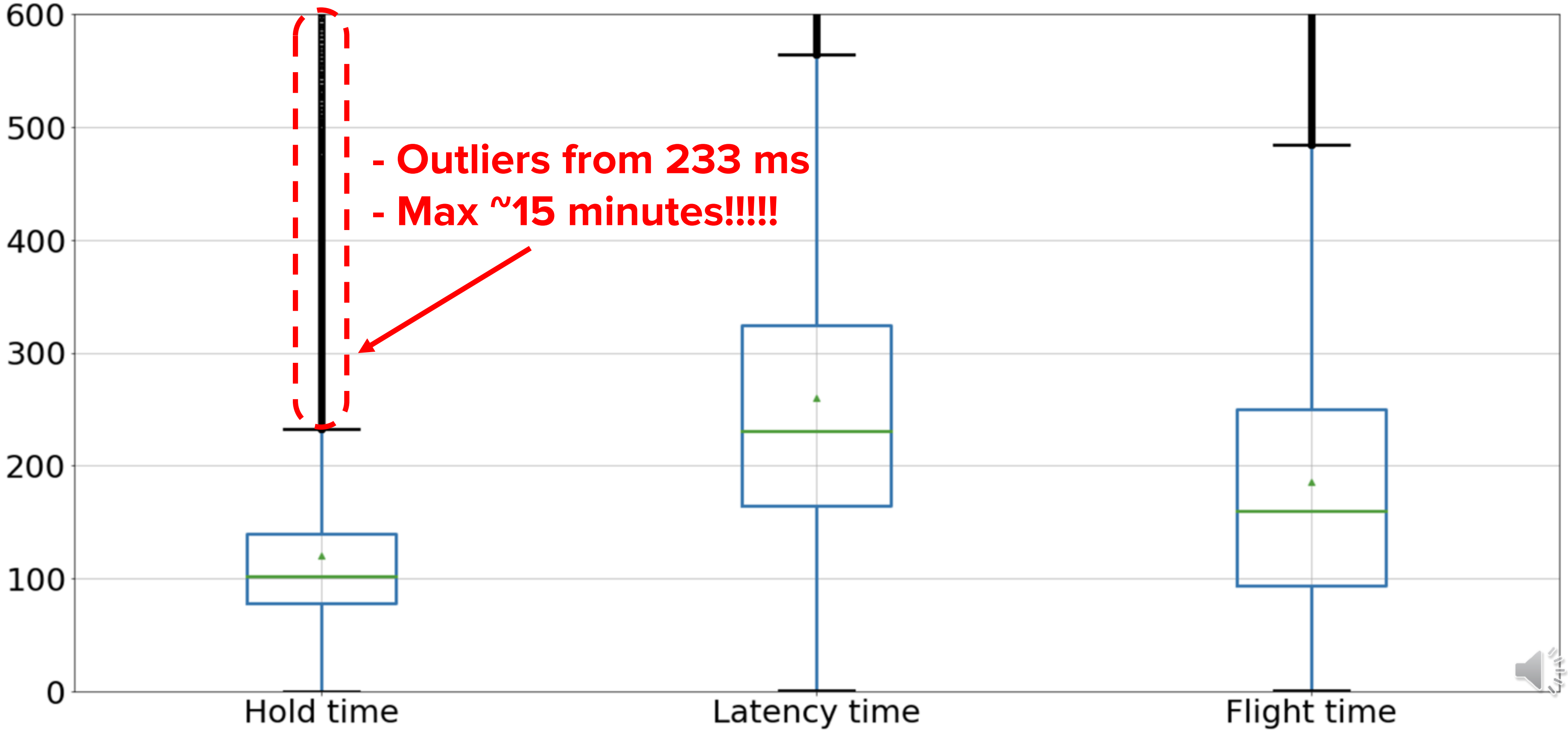


~125 users with less than 10 days of data = ~47% of all users

Exploring tappy dataset

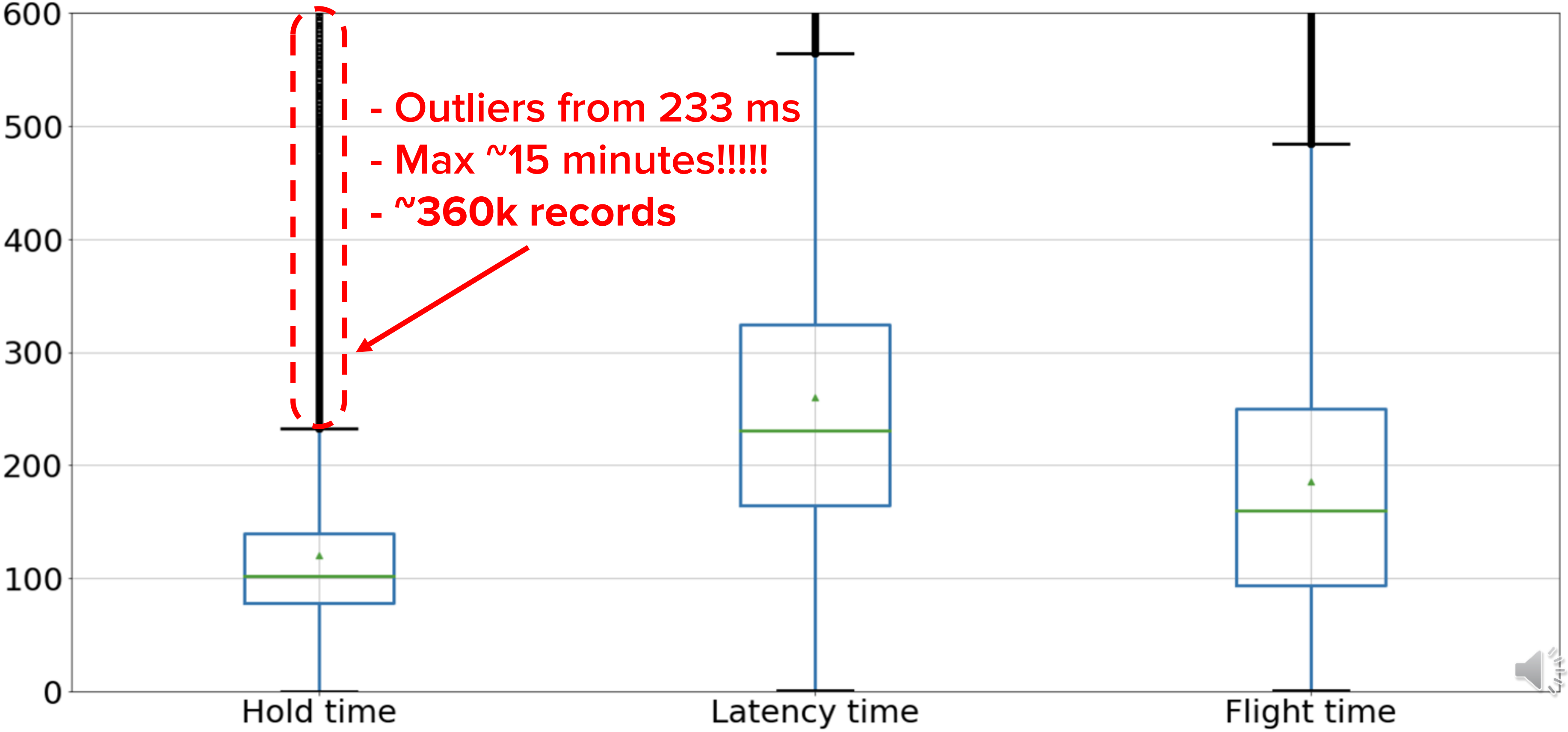


Exploring tappy dataset



- Outliers from 233 ms
- Max ~15 minutes!!!!!

Exploring tappy dataset



- Outliers from 233 ms
- Max ~15 minutes!!!!!
- ~360k records

Users Files

The filename of each user file contains a 10 character code, used to cross reference to the keystroke data files for that user.

- **BirthYear**: User's year of birth (YYYY)
- **Gender**: Their gender [Male/Female]
- **Parkinsons**: Whether the have parkinson's or not [True/False]
- **Tremors**: Whether they have tremors [True/False]
- **Diagnosis Year**: If they have Parkinson's, when was it first diagnosed
- Whether there is **sidedness** of movement [Left/Right/None] (self reported)
- **UPDRS**: The UPDRS score (if known) [1 to 5]
- **Impact**: The Parkinsons disease severity or impact on their daily life [Mild/Medium/Severe] (self reported)
- **Levadopa**: Whether they are using Sinemet and the like [Yes/No]
- **DA**: Whether they are using a dopamine agonist [Yes/No]
- **MAOB**: Whether they are using an MAO-B inhibitor [Yes/No]
- **Other**: Whether they are taking another Parkinson's medication [Yes/No]

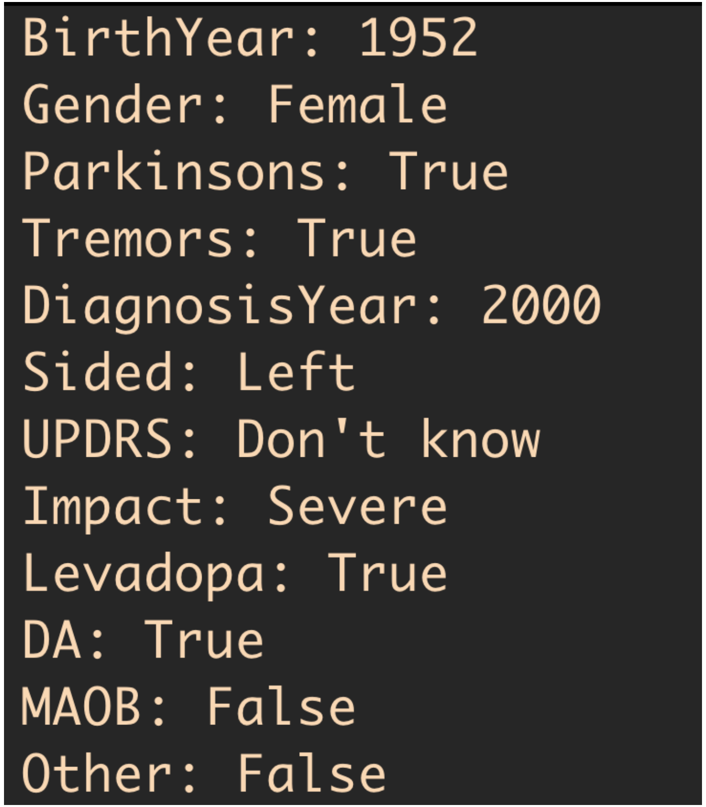
Users Files

The filename of each user file contains a 10 character code, used to cross reference to the keystroke data files for that user.

- **BirthYear**: User's year of birth (YYYY)
- **Gender**: Their gender [Male/Female]
- **Parkinsons**: Whether the have parkinson's or not [True/False]
- **Tremors**: Whether they have tremors [True/False]
- **Diagnosis Year**: If they have Parkinson's, when was it first diagnosed
- Whether there is **sidedness** of movement [Left/Right/None] (self reporte
- **UPDRS**: The UPDRS score (if known) [1 to 5]
- **Impact**: The Parkinsons disease severity or impact on their daily life [Mild/Medium/Severe] (self reported)
- **Levadopa**: Whether they are using Sinemet and the like [Yes/No]
- **DA**: Whether they are using a dopamine agonist [Yes/No]
- **MAOB**: Whether they are using an MAO-B inhibitor [Yes/No]
- **Other**: Whether they are taking another Parkinson's medication [Yes/No]

Our target variable!!

User_0EA27ICBLF.txt



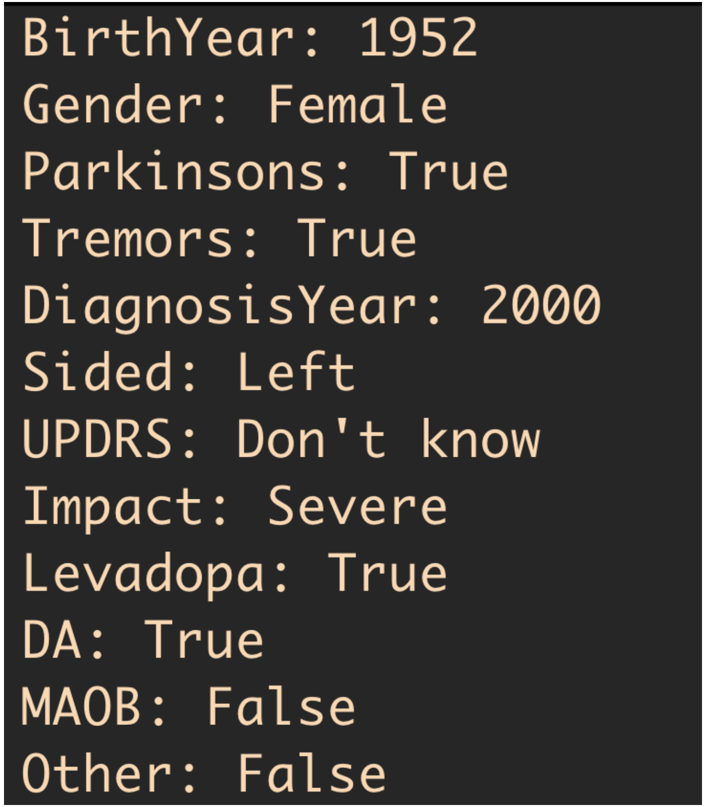
```
BirthYear: 1952
Gender: Female
Parkinsons: True
Tremors: True
DiagnosisYear: 2000
Sided: Left
UPDRS: Don't know
Impact: Severe
Levadopa: True
DA: True
MAOB: False
Other: False
```

- **Total of 227 files to load**

- A different format / structure

- .....

User_0EA27ICBLF.txt



```
BirthYear: 1952
Gender: Female
Parkinsons: True
Tremors: True
DiagnosisYear: 2000
Sided: Left
UPDRS: Don't know
Impact: Severe
Levadopa: True
DA: True
MAOB: False
Other: False
```

- Total of 227 files to load

- **A different format / structure**

- .....

User_0EA27ICBLF.txt

```
BirthYear: 1952
Gender: Fe  le
Par    sor
Tre
MAC  . False
Other: False
```

- Total of 227 files to load

- A different format / structure

- **A bunch of missing values**

- .....

User_0EA27ICBLF.txt

```
BirthYear: 1952
Gender: Female
MAC: False
Other: False
```

- Total of 227 files to load

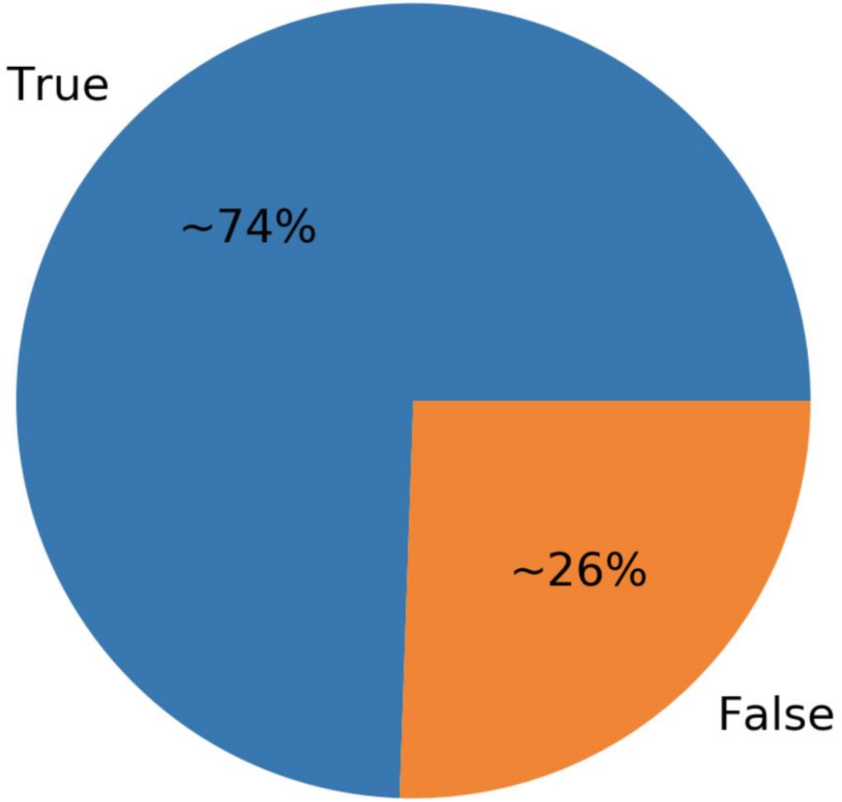
- A different format / structure

- A bunch of missing values

- After merging we went down to 217 users

Imbalanced dataset

User with Parkinson?



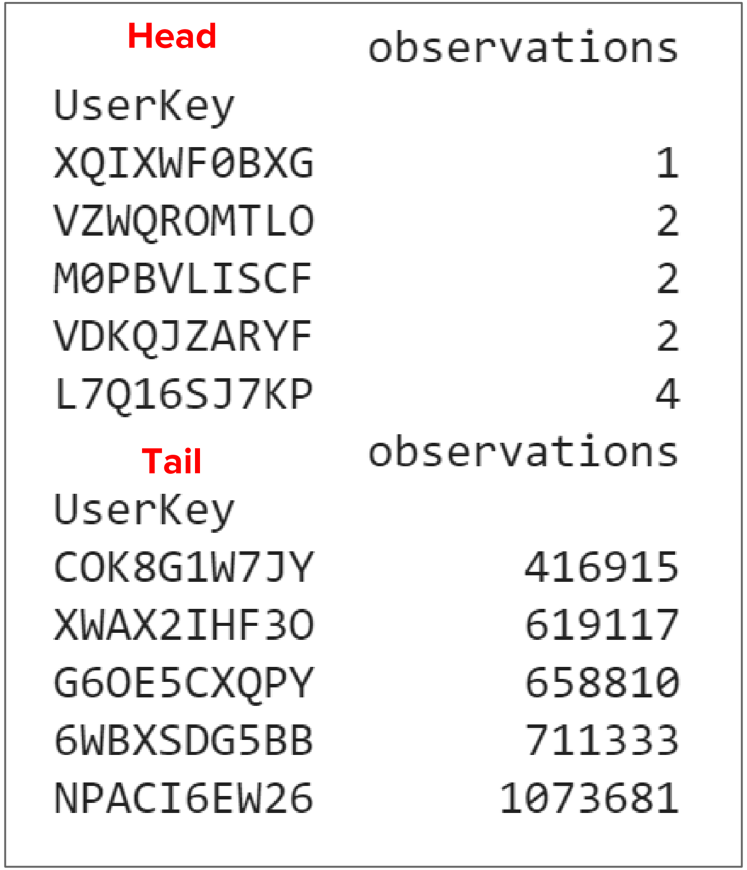
Feature Engineering

Data Cleaning

OK, maybe not ALL.

Just the ones we need.

Some users have very little data



```
     Head          observations
UserKey
XQIXWF0BXG                    1
VZWQROMTLO                    2
M0PBVLISCF                    2
VDKQJZARYF                    2
L7Q16SJ7KP                    4
     Tail          observations
UserKey
COK8G1W7JY               416915
XWAX2IHF3O               619117
G6OE5CXQPY               658810
6WBXSDG5BB               711333
NPACI6EW26              1073681
```

- Remove users with < 1000 observations (keystrokes)

Removing not useful data & outliers

- Remove the keystrokes with hold_time > **1000ms** (0.999 percentile cut off is **445.3**)
- Removing keystrokes involve “**S**” (space bar is not very useful in finding hand movement)
- Remove **flight_time** column as it is redundant
  flight_time = latency_time - hold_time

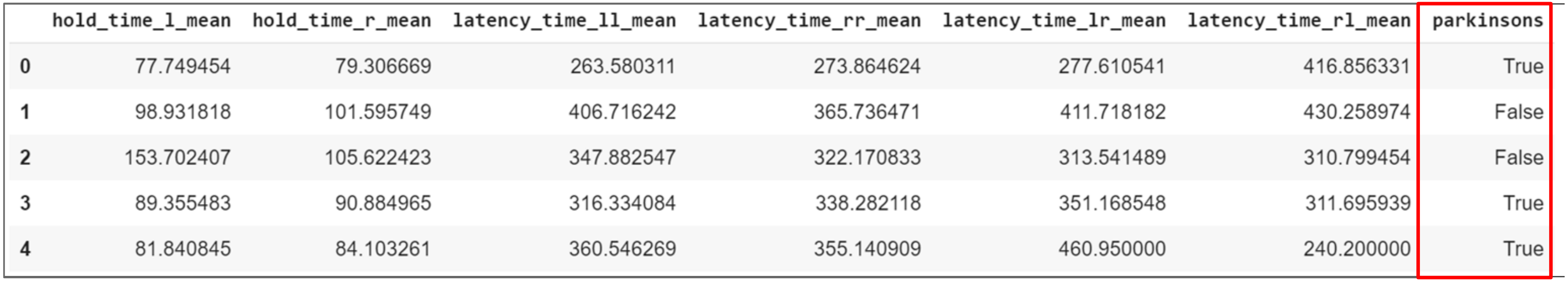
Fixing NaN

- **Impact**
  - For non Parkinson users, create a new category = **None**
  - For Parkinsons user (only 4 with missing Impact), use Mode (**Medium**)

- **Sided** - 108 users missing that data
  - All users with no Parkinson are missing this field
  - It could be a useful data, but have to drop

Basic features

6 features: mean of each hold_time and latency_time



	hold_time_l_mean	hold_time_r_mean	latency_time_ll_mean	latency_time_rr_mean	latency_time_lr_mean	latency_time_rl_mean	parkinsons
0	77.749454	79.306669	263.580311	273.864624	277.610541	416.856331	True
1	98.931818	101.595749	406.716242	365.736471	411.718182	430.258974	False
2	153.702407	105.622423	347.882547	322.170833	313.541489	310.799454	False
3	89.355483	90.884965	316.334084	338.282118	351.168548	311.695939	True
4	81.840845	84.103261	360.546269	355.140909	460.950000	240.200000	True

More features?

- **Diff** and **Abs_diff** for **hold_time_mean** for **L** and **R**
  - => 2 new feature
- **Diff** and **Abs_diff** for **latency_time_mean** for **LR <-> RL** and **LL <-> RR**
  - => 4 new features
- **Std**, **Skew**, **Kurtosis** of each mean
  - Holdtime (L, R) => 6 new features
  - Latency_time (LL, LR, RL, RR) => 12 new features

Now we have 30 features in total

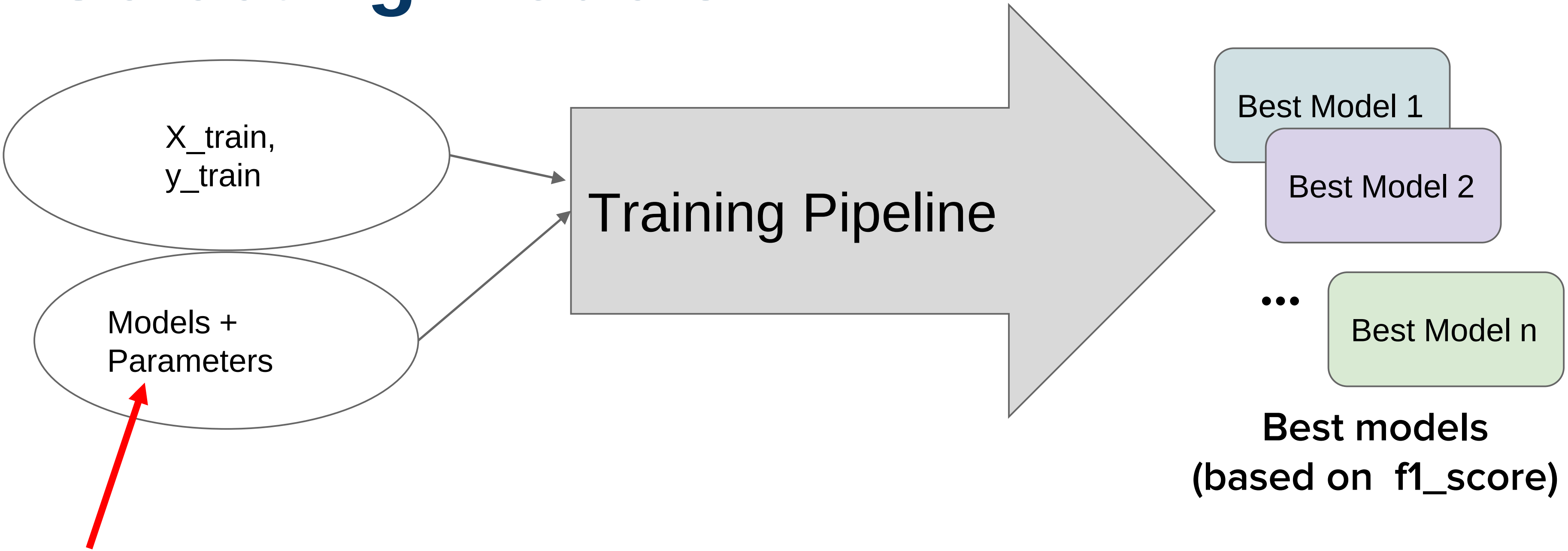
Training Models

Tedious to train model one by one...

Use Pipeline and functions to train and compare models

```
training_pipeline = Pipeline(verbose=False, steps=[
  ('pol', PolynomialFeatures(degree = 1)),
  ('nor', Normalizer()),
  ('clf', EstimatorDecorator())
])
param_grid = [{
    'clf__estimator': [RandomForestClassifier()],
     #… and all other models}]

best_models, grid_search =
    train_and_select_any_top_n(X_train, y_train,
                               training_pipeline, param_grid,
                               cv=3, scoring=’f1’)
```

Selecting models

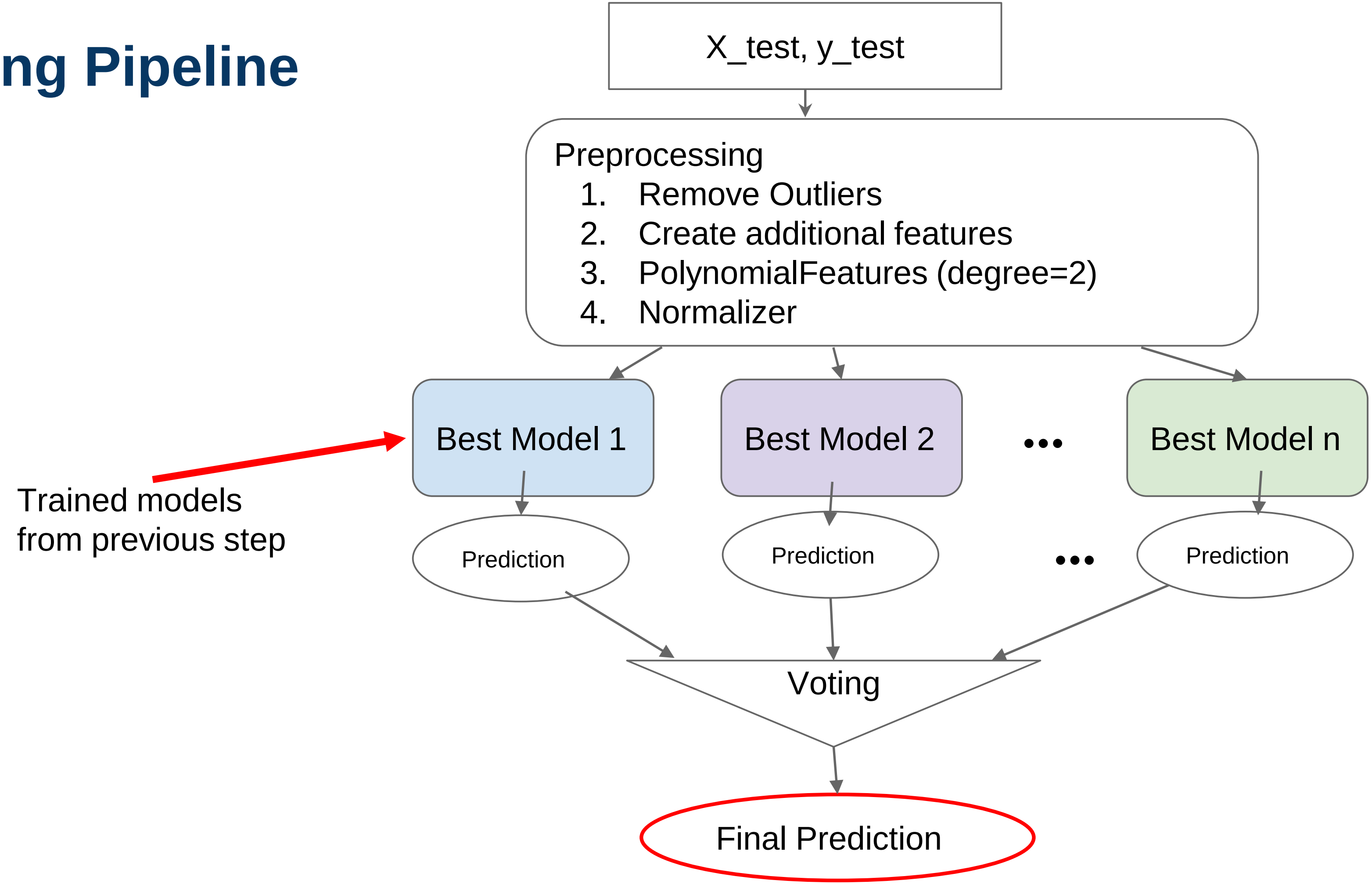


Best models (based on f1_score)

Models included:

RandomForestClassifier, LogisticRegression, SVC,GradientBoostingClassifier, KNeighborsClassifier, GaussianNB, DecisionTreeClassifier, XGBClassifier, AdaBoostClassifier

Testing Pipeline



X_test, y_test

Preprocessing

1. Remove Outliers
2. Create additional features
3. PolynomialFeatures (degree=2)
4. Normalizer

Best Model 1

Best Model 2

...

Best Model n

Trained models from previous step

Results

VotingClassifier (soft)

```
> F1 delta (train-test): 0.004963

> Scores          <train>  | <test>
> F1           : 0.853448 | 0.848485
> Precision    : 0.744361 | 0.736842
> Recall       : 1.000000 | 1.000000
> Specificity: 0.000000 | 0.000000
> Accuracy     : 0.744361 | 0.736842
> AUC          : 0.500000 | 0.500000
```

Looks pretty good, except....

“Always True” Classifier

It is same as the Always True Classifier…

```
> Scores         <train>  | <test>
> F1          : 0.853448 | 0.848485
> Precision   : 0.744361 | 0.736842
> Recall      : 1.000000 | 1.000000
> Specificity: 0.000000 | 0.000000
> Accuracy    : 0.744361 | 0.736842
> AUC         : 0.500000 | 0.500000
>
> ConfMatrix : [[ 0 34] | [[ 0 15]
                [ 0 99]]|  [ 0 42]]
```

What to do?

Are we using the right data set?

“Early stage of Parkinson’s….”

Keyword: ***Early*** stage

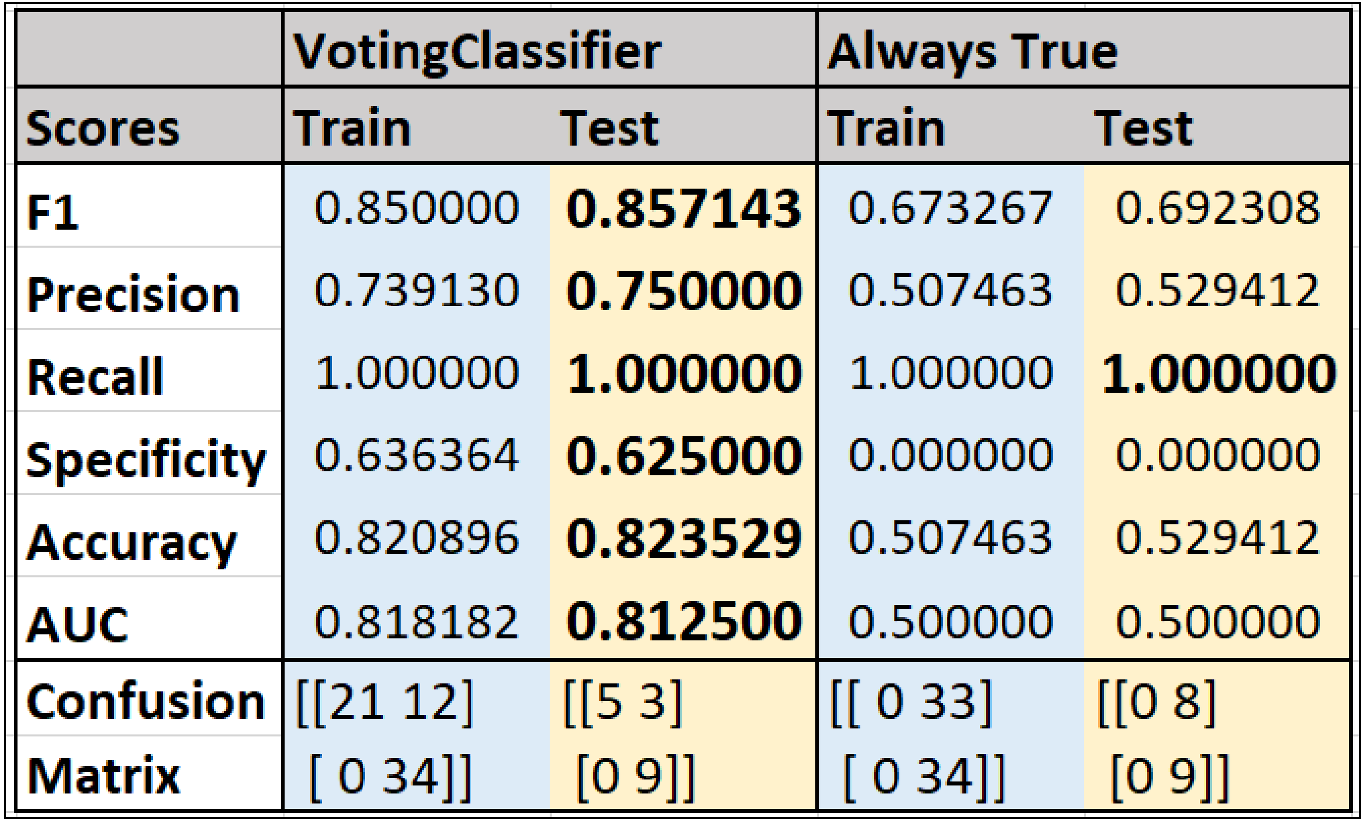
May need further filtering…

=> Only include Parkinson’s users with Mild impact.

VotingClassifier (soft)

```
> Scores     <train>    | <test>
> F1         : 0.850000 | 0.857143
> Precision  : 0.739130 | 0.750000
> Recall     : 1.000000 | 1.000000
> Specificity: 0.636364 | 0.625000
> Accuracy   : 0.820896 | 0.823529
> AUC        : 0.818182 | 0.812500
> Conf Matrix: [[21 12] | [[5 3]
                [ 0 34]]|  [0 9]]
```

Compare with “Always True” predictor



	VotingClassifier		Always True	
Scores	Train	Test	Train	Test
F1	0.850000	**0.857143**	0.673267	0.692308
Precision	0.739130	**0.750000**	0.507463	0.529412
Recall	1.000000	**1.000000**	1.000000	**1.000000**
Specificity	0.636364	**0.625000**	0.000000	0.000000
Accuracy	0.820896	**0.823529**	0.507463	0.529412
AUC	0.818182	**0.812500**	0.500000	0.500000
Confusion	[[21 12]	[[5 3]	[[0 33]	[[0 8]
Matrix	[0 34]]	[0 9]]	[0 34]]	[0 9]]

Conclusion

- Models improved to over 0.8 (both f1_score and accuracy) after reducing the dataset to:
  - Keystroke data with <= 1000ms hold time
  - At least 1000 keystroke data per user
  - User with Parkinson’s impact == Mild only
- The models can further be improved as the research paper has obtained over 0.9 sensitivity and specificity.

Thank you!