DATA 3401 Group 6

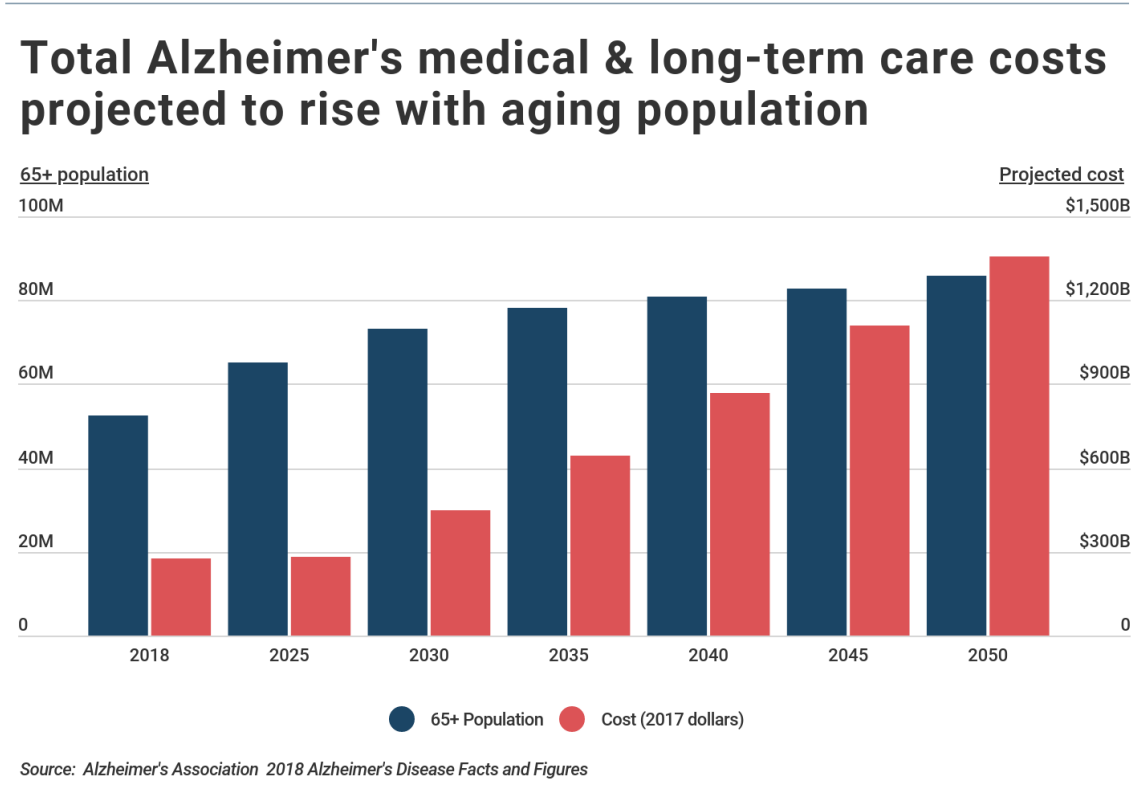
Alison Sturge and Robert Tamayo IV

Introduction

Alzheimer's is a disease that devastates patients and their families. The progressive degeneration of nerve cells affects the mental a physical abilities of its victims, eventually leaving them unable to care for themselves, unaware of their surroundings, and ultimately unable to even swallow, leading to death.

Introduction- The Future of Alzheimer's

aPlaceforMom



Source: Alzheimer's Association 2018 Alzheimer's Disease Facts and Figures

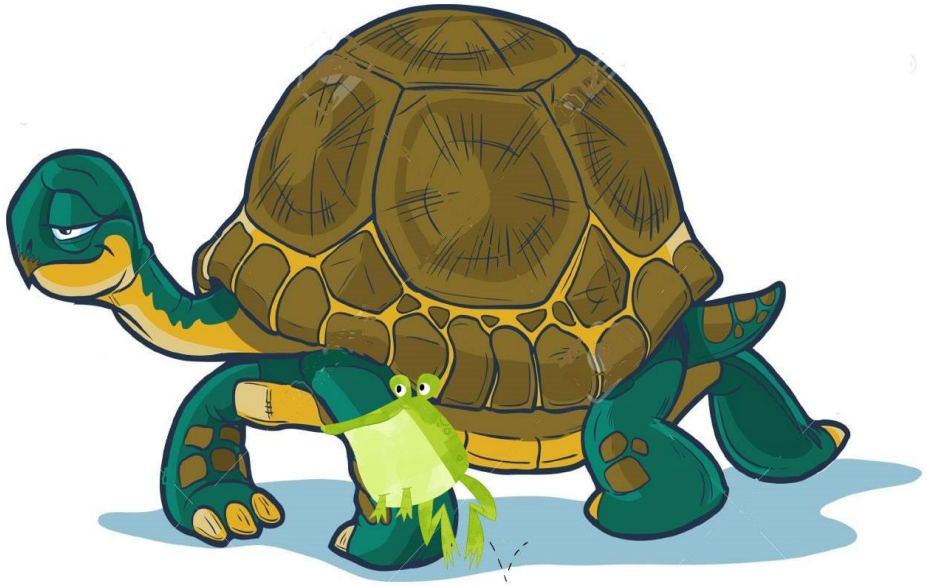
As the over-65 population grows, long term costs for Alzheimer's care is expected to explode, growing by 500% in the next 30 years.

Without a cure, early diagnosis is still the best way to delay progression of the disease, which would also help keep medical costs down.

Introduction- DARWIN to the Rescue?

Diagnosis **A**lzheime**R** **WI**th ha**N**dwriting, or DARWIN, is an investigative method that uses 25 unique tasks to test both the mental and motor skills required for handwriting.

Can DARWIN diagnose Alzheimer's quickly and with minimum expense?



Introduction- The Data

The data used can be found at https://www.kaggle.com/datasets/alisonsturge/darwin. It was originally sourced from the study entitled *Diagnosing Alzheimer's disease from on-line handwriting: A novel dataset and performance benchmarking* by Cilia et al. (2022).

- Previous studies either applied to Parkinson's only, were a small sample, or had a very small control group
- Collected from 174 study participants
- 25 separate tasks, designed to target different parts of the brain

Introduction- The ML Algorithms

- These are the four ML algorithms that we used:
  - Artificial Neural Networks (ANN)
  - Support Vector Machine (SVM)
  - Random Forest (RF)
  - Logistic Regression (LR)
- Note that SVM, RF, and LR are all supervising learning models that can solve classification problems.

Data Preparation- .info() of ‘data’

Notice how there are 174 entries, which coincides with the number of participants in the DARWIN dataset. Also, there are no null entries, so we don’t have worry about removing columns, for example.

```
In [60]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 174 entries, 0 to 173
Columns: 452 entries, ID to class
dtypes: float64(300), int64(150), object(2)
memory usage: 614.6+ KB
```

Data Preparation- Changing H/P to 0/1

DARWIN's classification is noted in string format, where 'H' = healthy and 'P' = patient. To make numerical and EDA processes much easier, we converted the strings to integers 0 and 1, respectively. We can still perform ML algorithms with ‘class’.

In [7]:
```
data['class'].replace(['H','P'],[0,1], inplace = True)
data.head()
```

Out[7]:

mean_jerk_on_paper25	mean_speed_in_air25	mean_speed_on_paper25	num_of_pendown25	paper_time25	pressure_mean25	pressure_var25	total_time25	class
0.024471	5.596487	3.184589	71	40120	1749.278166	296102.7676	144605	1
0.018368	1.665973	0.950249	129	126700	1504.768272	278744.2850	298640	1
0.017174	4.000781	2.392521	74	45480	1431.443492	144411.7055	79025	1
0.019860	4.206746	1.613522	123	67945	1465.843329	230184.7154	181220	1
0.020872	3.319036	1.680629	92	37285	1841.702561	158290.0255	72575	1

Data Preparation- Getting Our Group 6 Data

Our tasks are 2, 6, 12, 15, and 22. To get our data, we simply ”hard-filtered” out columns we needed. Here’s an example:

In [10]:
```
df1 = data[['air_time2', 'disp_index2', 'gmrt_in_air2', 'gmrt_on_paper2', 'max_x_extension2',
 'max_y_extension2', 'mean_acc_in_air2', 'mean_acc_on_paper2', 'mean_gmrt2',
 'mean_jerk_in_air2', 'mean_jerk_on_paper2', 'mean_speed_in_air2', 'mean_speed_on_paper2',
 'num_of_pendown2', 'paper_time2', 'pressure_mean2', 'pressure_var2', 'total_time2', 'class']]
```

We called the whole dataset with all 5 tasks ‘df’. This is the main dataset we’ll be working with for the rest of the project. The other tasks are numbered 'df1’, ‘df2’, …,'df5.'

Data Preparation- ‘df’

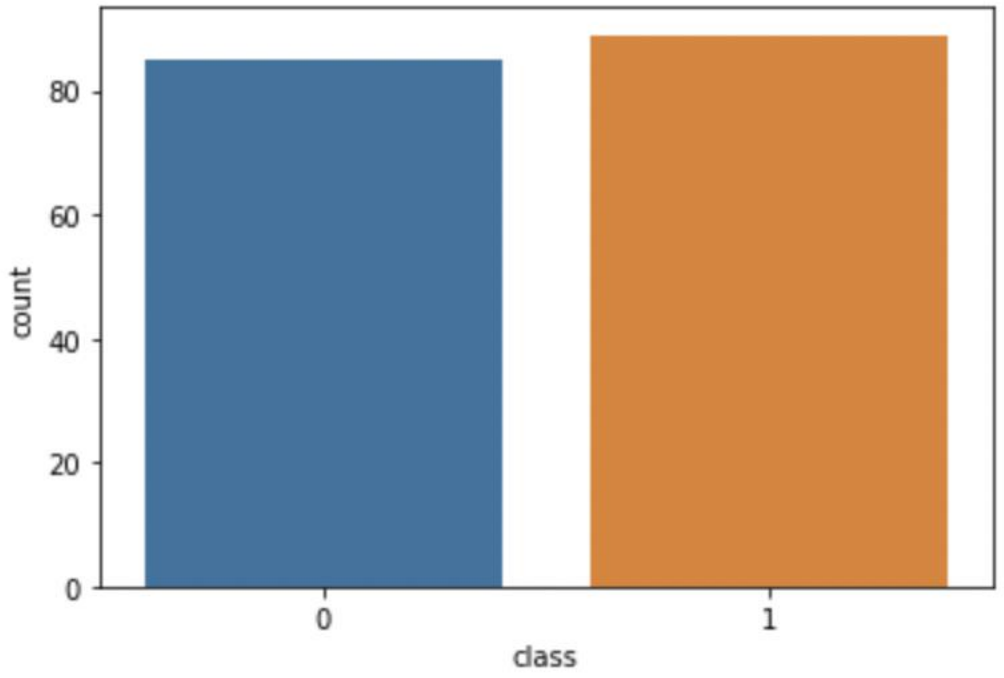
In case you were curious, here is ‘df’:

```
In [900]: df = data[['air_time2', 'disp_index2', 'gmrt_in_air2', 'gmrt_on_paper2', 'max_x_extension2',
           'max_y_extension2', 'mean_acc_in_air2', 'mean_acc_on_paper2', 'mean_gmrt2',
           'mean_jerk_in_air2', 'mean_jerk_on_paper2', 'mean_speed_in_air2', 'mean_speed_on_paper2',
           'num_of_pendown2', 'paper_time2', 'pressure_mean2', 'pressure_var2', 'total_time2',
           'air_time6', 'disp_index6', 'gmrt_in_air6', 'gmrt_on_paper6', 'max_x_extension6',
           'max_y_extension6', 'mean_acc_in_air6', 'mean_acc_on_paper6', 'mean_gmrt6', 'mean_jerk_in_air6',
           'mean_jerk_on_paper6', 'mean_speed_in_air6', 'mean_speed_on_paper6', 'num_of_pendown6',
           'paper_time6', 'pressure_mean6', 'pressure_var6', 'total_time6', 'air_time12', 'disp_index12',
           'gmrt_in_air12', 'gmrt_on_paper12', 'max_x_extension12', 'max_y_extension12',
           'mean_acc_in_air12', 'mean_acc_on_paper12', 'mean_gmrt12', 'mean_jerk_in_air12',
           'mean_jerk_on_paper12', 'mean_speed_in_air12', 'mean_speed_on_paper12', 'num_of_pendown12',
           'paper_time12', 'pressure_mean12', 'pressure_var12', 'total_time12', 'air_time15',
           'disp_index15', 'gmrt_in_air15', 'gmrt_on_paper15', 'max_x_extension15', 'max_y_extension15',
           'mean_acc_in_air15', 'mean_acc_on_paper15', 'mean_gmrt15', 'mean_jerk_in_air15',
           'mean_jerk_on_paper15', 'mean_speed_in_air15', 'mean_speed_on_paper15', 'num_of_pendown15',
           'paper_time15', 'pressure_mean15', 'pressure_var15', 'total_time15', 'air_time22',
           'disp_index22', 'gmrt_in_air22', 'gmrt_on_paper22', 'max_x_extension22', 'max_y_extension22',
           'mean_acc_in_air22', 'mean_acc_on_paper22', 'mean_gmrt22', 'mean_jerk_in_air22',
           'mean_jerk_on_paper22', 'mean_speed_in_air22', 'mean_speed_on_paper22', 'num_of_pendown22',
           'paper_time22', 'pressure_mean22', 'pressure_var22', 'total_time22', 'class']]
```

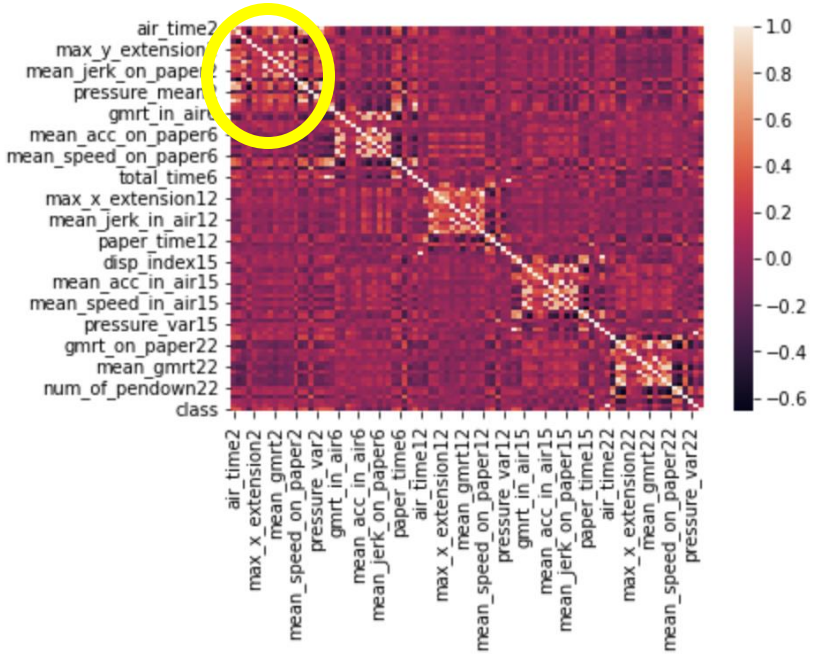
EDA- Count plot of ‘df’

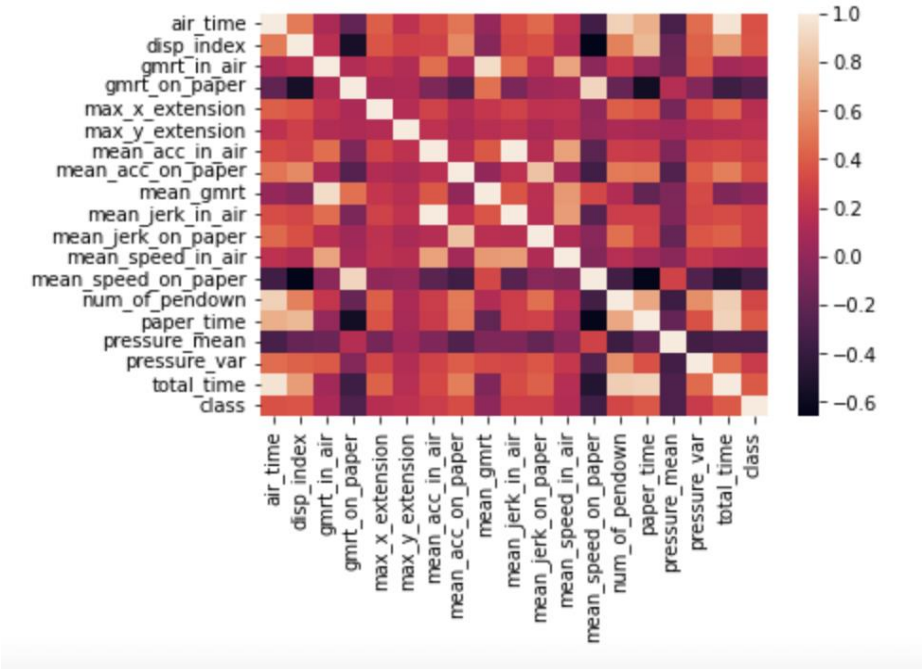
- First, we did a count plot of the full dataset.
  - The data is balanced, so we can use accuracy for comparison.
  - Specifically, 85 participants are healthy, while 89 participants are patients.

```
In [13]: sns.countplot(data = df, x = 'class')
Out[13]: <AxesSubplot:xlabel='class', ylabel='count'>
```



EDA- Heatmap





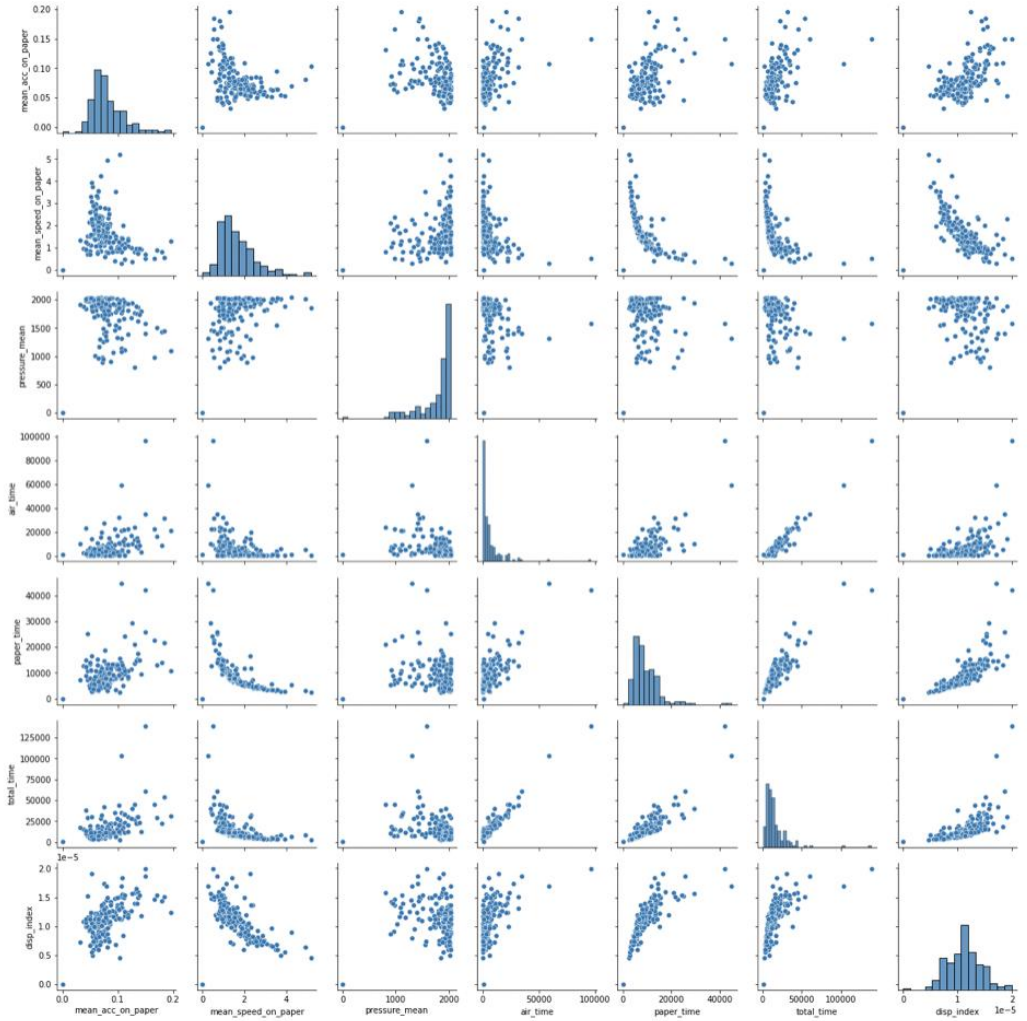
Next, we made a heatmap for the full dataset to show the correlation between features in our tasks. If you look closely, the heatmap shows 5 different heatmaps!
So we did easier-to-read individual heat maps for each task.

EDA- Pair plot

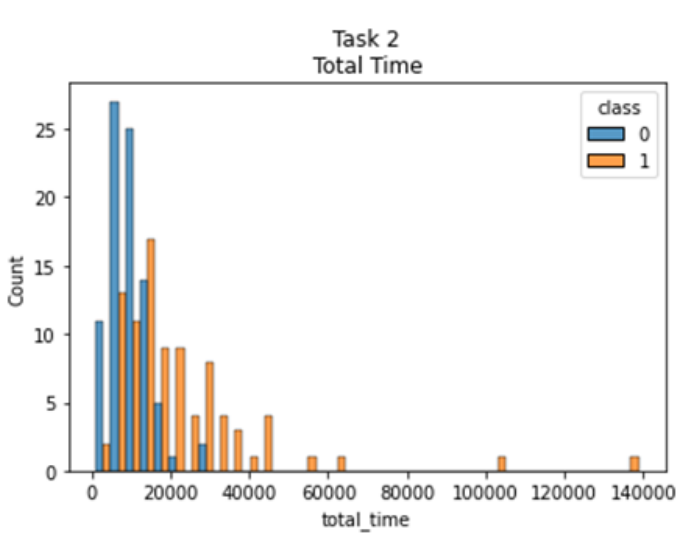
- Thanks to the heatmap, we're now able to do a pair plot of columns with the best correlation values.
- Now, we'll have strong scatterplots that are either clearly increasing or decreasing.
- Hardly any of the scatterplots have no correlation.

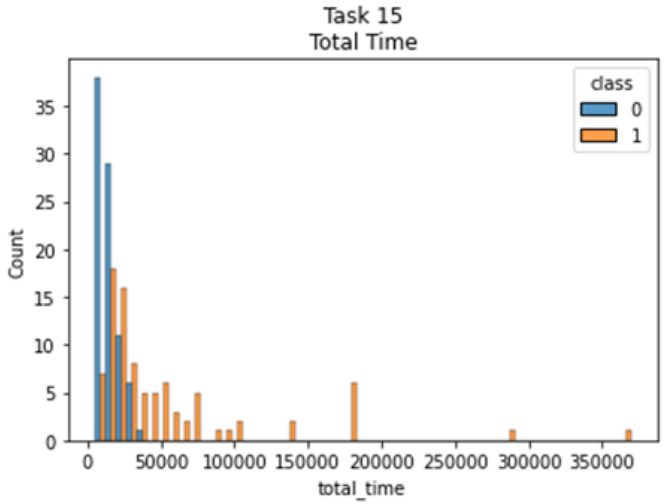
```
In [77]: #Plot of columns with greatest correlation values

         cols_to_plot = ['mean_acc_on_paper', 'mean_speed_on_paper', 'pressure_mean', 'air_time'
         sns.pairplot(df1[cols_to_plot])
```

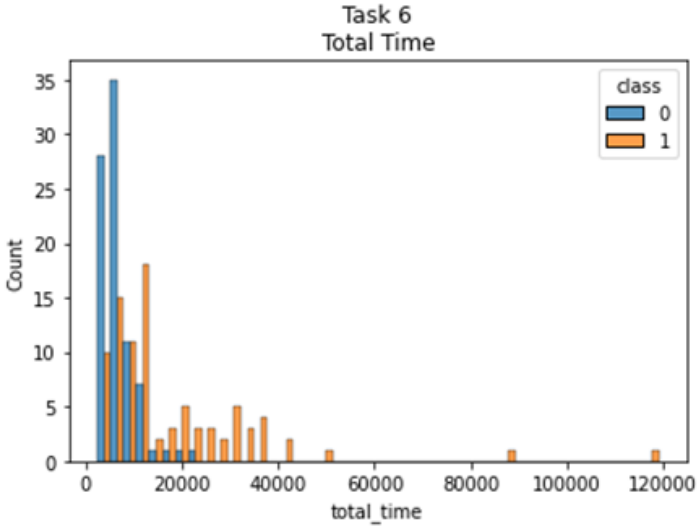
```
Out[77]: <seaborn.axisgrid.PairGrid at 0x7f547f0e41d0>
```

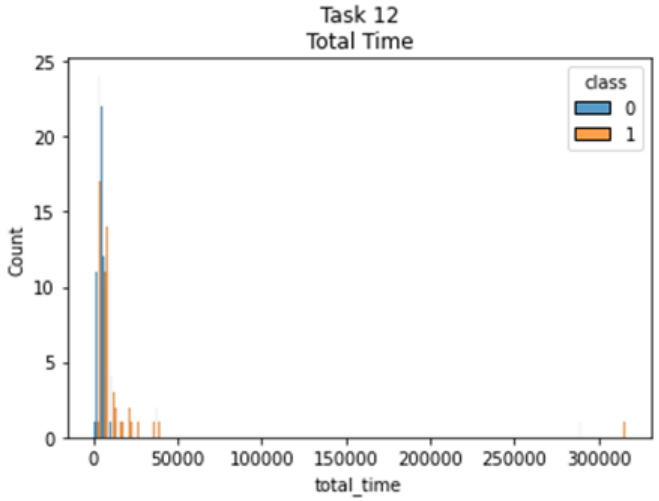


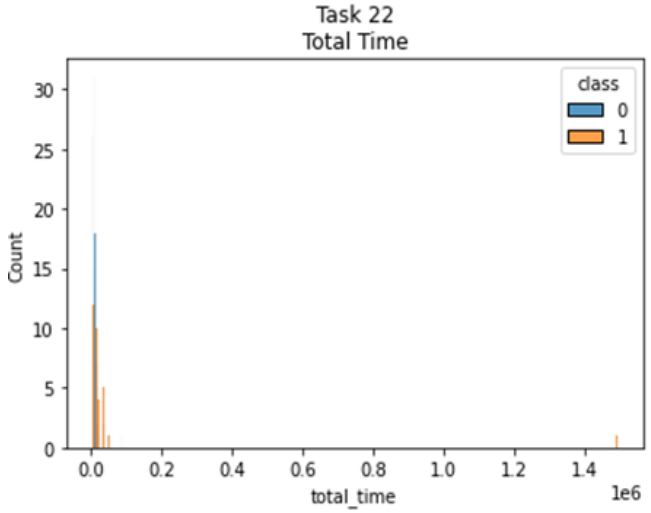
EDA – Histograms of Total Time





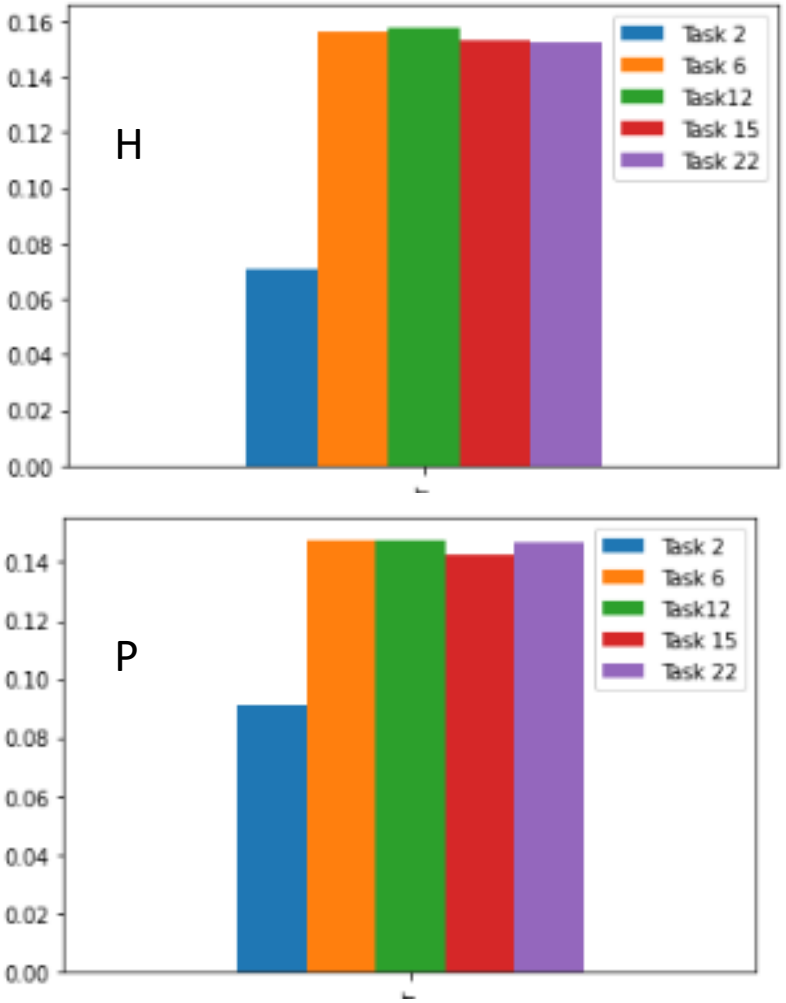




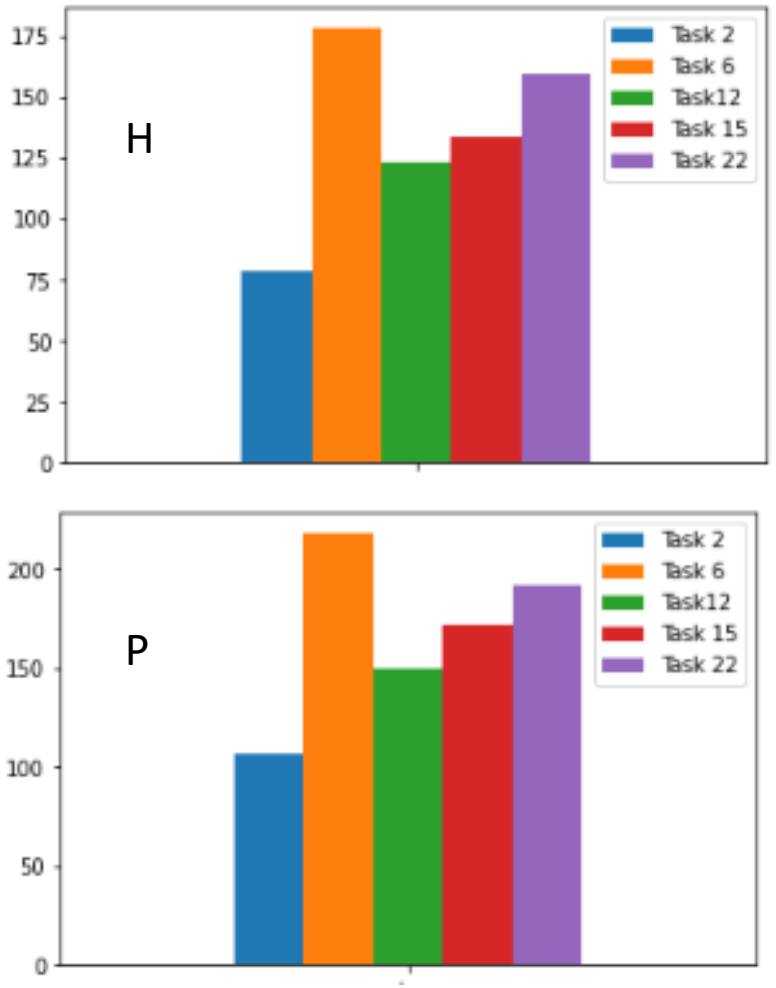


Healthy participants perform the tests much faster than patient participants. All tasks have outliers among Alzheimer's patients, with the most impactful outliers on tasks 12 and 22.

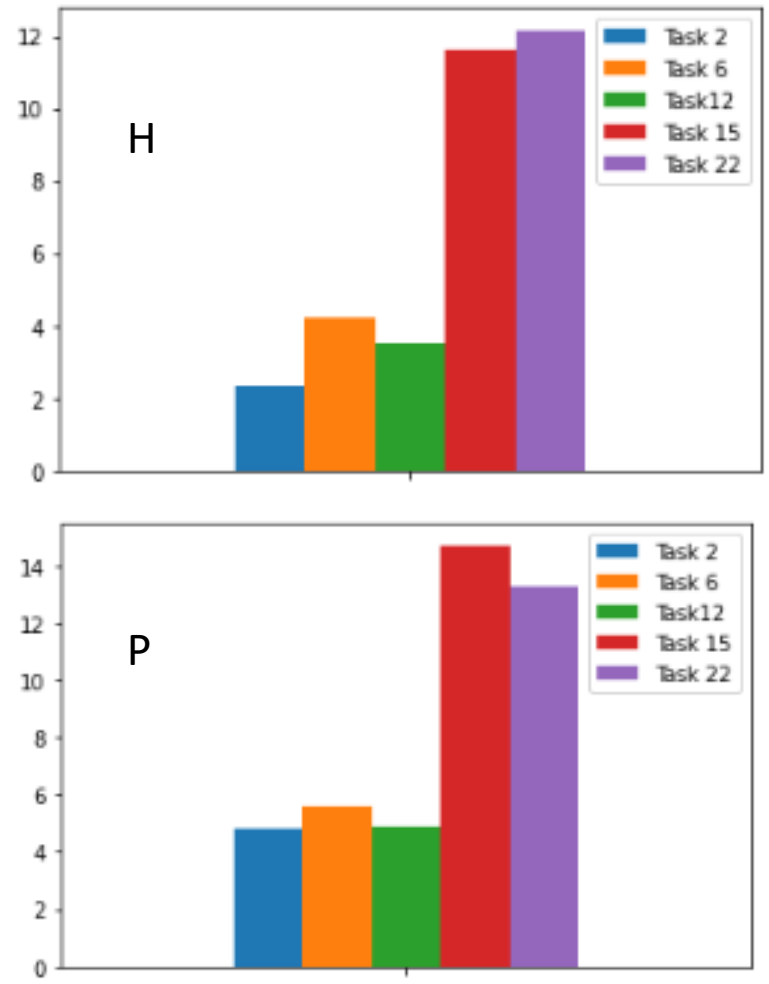
Acceleration on Paper



Tremor on Paper



Pendown



These plots, grouped by feature and class, show that the averages for Task 2 tend to be lower than the rest of the tasks. We cannot be sure if this indicates more or less significance in diagnosing Alzheimer's, or if it is merely due to Task 2 being the only task that is not a copying task.

ML Algorithms

- Before we run the ML algorithms, we split the data into train/test.
- Once the data is split, we use MinMaxScaler to scale the training data.
- We made test_size = 0.25 to maximize the training set, and we made random_state = 100 for reproducible results.

In [ ]:
```
X = df.drop("class", axis=1)
y = df["class"]
X_train,X_test,y_train,y_test = train_test_split(X, y, test_size = 0.25, random_state = 100)
```

ML Algorithms- Accuracies

We applied ANN, SVM, RF, and LR algorithms on both the full dataset as well as on individual tasks. Average represents the mean of the 5 individual tasks.

Accuracies pulled from the classification reports found as follows:

Accuracy as Percent	Full dataset	Task 2 Study	Task 2	Task 6 Study	Task 6	Task 12 Study	Task 12	Task 15 Study	Task 15	Task 22 Study	Task 22
ANN	86	60.14 (±6.18)	68	73.14 (±6.52)	70	60.14 (±6.18)	70	69.29 (±6.07)	86	67.29 (±6.90)	73
SVM	82	60.86 (±6.75)	64	74.71 (±5.73)	66	60.86 (±6.75)	82	71.14 (±7.35)	84	68.57 (±9.63)	75
RF	86	67.14 (±7.78)	66	72.43 (±7.66)	73	67.14 (±7.78)	70	72.14 (±6.35)	68	75.00 (±7.80)	75
LR	86	62.57 (±5.93)	68	74.00 (±6.81)	64	62.57 (±5.93)	73	73.71 (±6.85)	89	67.29 (±9.15)	68

ML Algorithms cont.

- We saw the best accuracy on Task 15, with logistic regression achieving 0.89. The whole dataset got good results as well, with accuracy score on all algorithms in excess of 0.80.
- Out of the individual tasks, SVM performed the best on average, and the scores for the accuracies ranged from 0.64 to 0.89.
- Miraculously, all the ML Algorithms got a chance to shine!

Conclusion

So, is DARWIN a viable means of diagnosing Alzheimer's? The high accuracy level achieved with some models on both the full dataset and some tasks suggests that it is.

A better look at the data by modeling all 25 individual tasks would give us more insight into whether some tasks have more influence on the outcome of the diagnosis.