Alzheimer's Disease Statistical Analysis Project - OASIS Brains Kaggle Dataset

Abhinav Vedati, Chris Lippi

12/18/2020

Introduction

Alzheimer's Disease is one of the world's most devastating and deadly diseases. By 2050 the number of people age 65 and older with Alzheimer's dementia in the United States is projected to reach 13.8 million(Assocation, n.d.). Despite this devastating statistic, Alzheimer's Disease is surprisingly not very well understood. While progress has been made over the past several decades in finding cures and treatments for cancers and diseases such as Measles, there has yet to be a single drug that improves the condition of or cures Alzheimer's Disease, and there has only been a single drug approved for the treatment of the disease since 2003(Cummings et al. 2020). Part of the reason why we have not yet found a cure for Alzheimer's Disease is that we don't understand the underlying causes of the disease very well. In this observational study, we aim to take the first steps towards establishing causality in the context of Alzheimer's Disease, by finding variables that are correlated with clinical dementia rating (a 5 point scale used to characterize the development of dementia within a patient) and Mini-Mental State Exam scores (the Mini-Mental State Exam is a quiz given to dementia patients; it is graded on a scale from 0 to 30 points, where a higher score indicates milder dementia). For the purposes of this study, we will be using a dataset provided by Kaggle and the Open Access Series of Imaging Studies (OASIS)

Part 1: Finding correlations

First, we loaded our dataset and inspected it using the names function. This allowed us to see the names of the columns that our dataset contains.

```
dataFrame <- read.csv("oasis_cross-sectional.csv", header=TRUE, sep=",")
names(dataFrame)
```

```
##  [1] "ID"    "M.F"   "Hand"  "Age"   "Educ"  "SES"   "MMSE"  "CDR"   "eTIV"
## [10] "nWBV"  "ASF"   "Delay"
```

Next, we defined two functions, which attempt to fit the data to a linear model and a non-linear model, in that order. Both of these functions return the r squared correlation coefficient for their models.

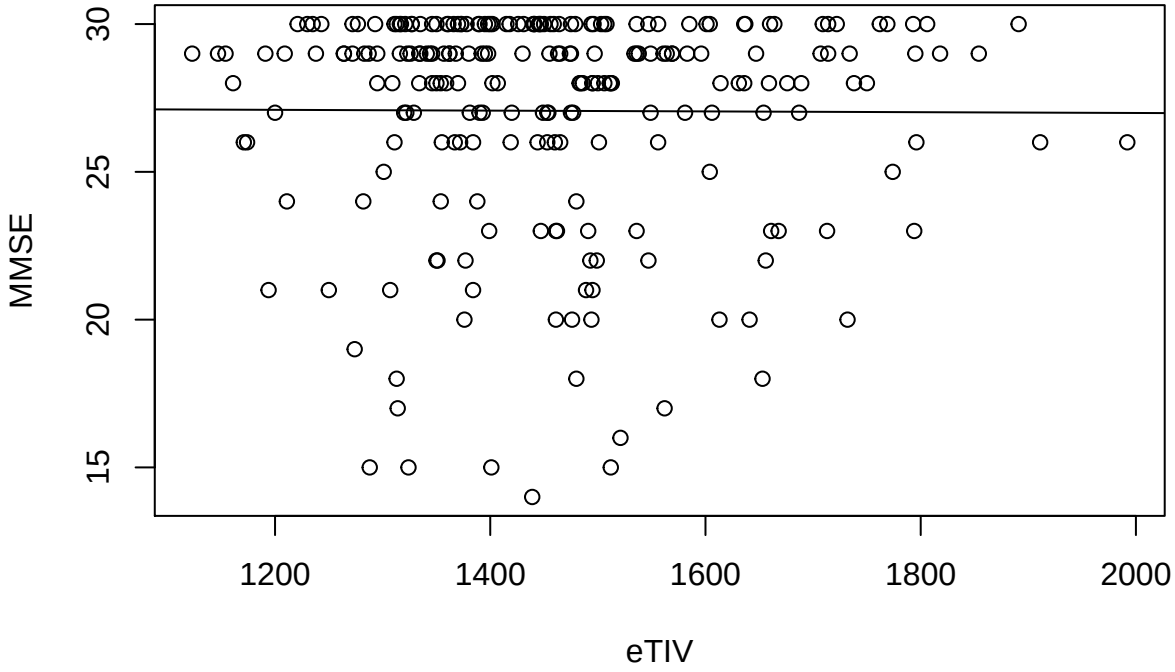
```
plot_linear <- function(columns, name1, name2) {
  neededData = na.omit(columns)
  plot(neededData[,name1], neededData[,name2], xlab=name1, ylab=name2)
  model <- lm(neededData[,name2] ~ neededData[,name1])
  abline(model)
  summary(model)$r.squared
}

plot_logarithmic <- function(columns, name1, name2) {
  neededData = na.omit(columns)
  plot(neededData[,name1],
       log(neededData[,name2]),
```

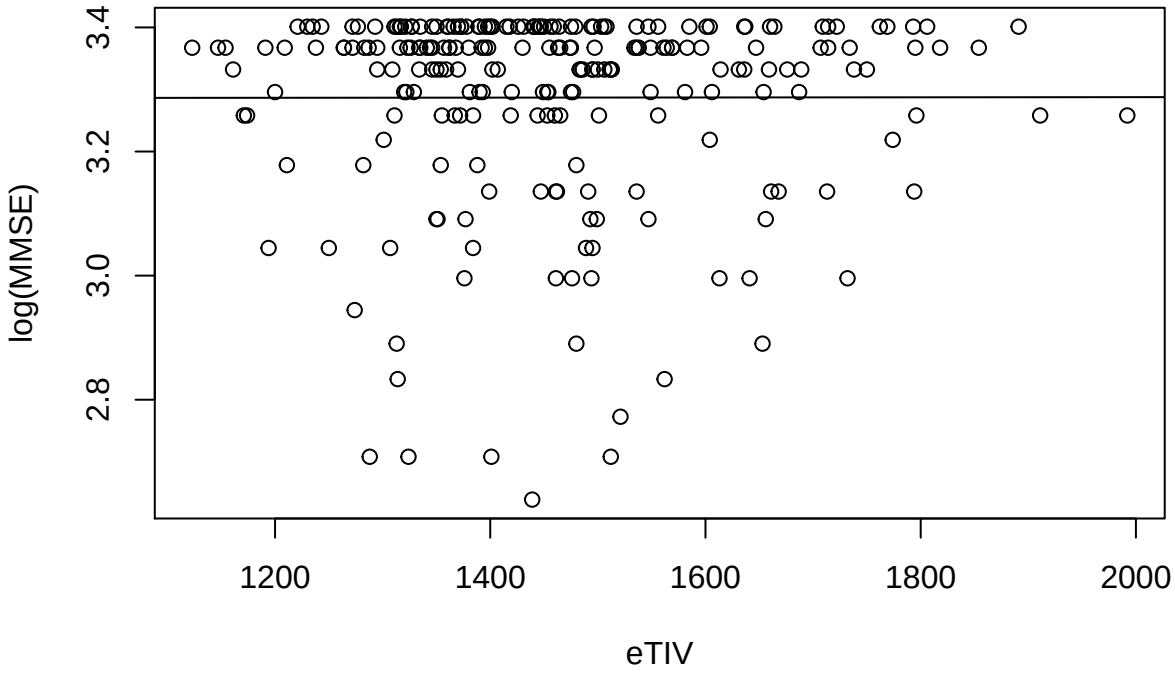
```
       xlab=name1,
       ylab=paste("log(",name2,")", sep="")
       )
  model <- lm(log(neededData[,name2]) ~ neededData[,name1])
  abline(model)
  summary(model)$r.squared
}
```

We then invididually graphed patient estimated total intracranial volume (eTIV) and patient normalized whole brain volume (nWBV) against patient Mini–Mental State Examination (MMSE) scores. For both graphs, we fit the data to a linear and a nonlinear model using our previously defined models.

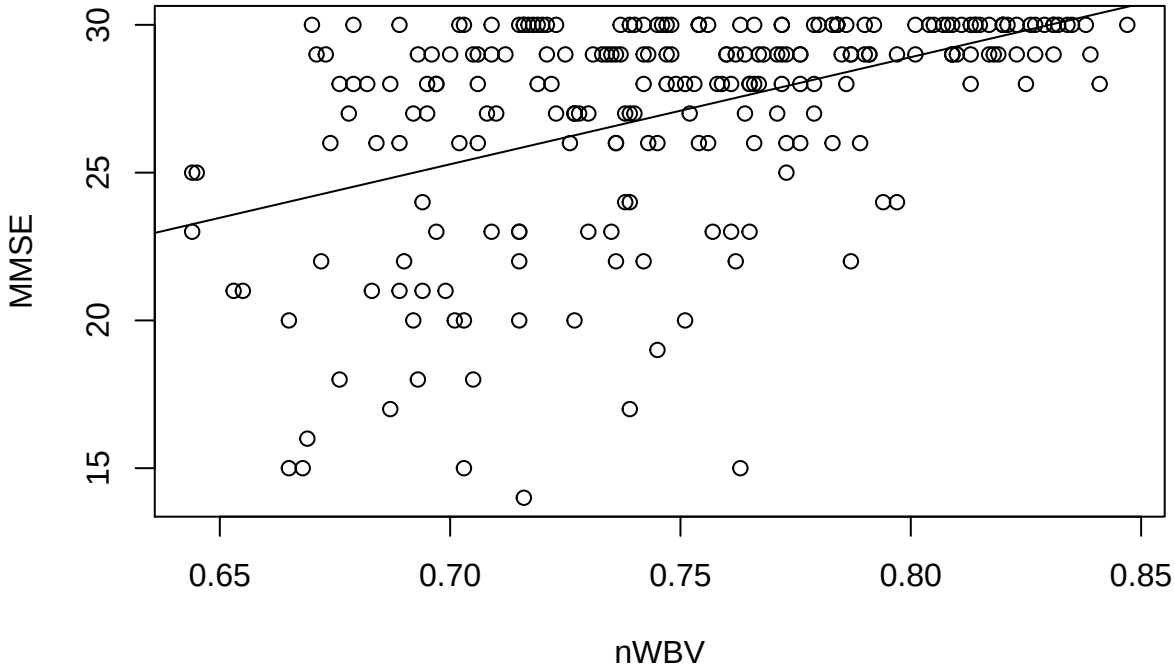
```
R_squared_eTIV_MMSE_lin <- plot_linear(dataFrame[,c(7,9)], "eTIV", "MMSE")
```



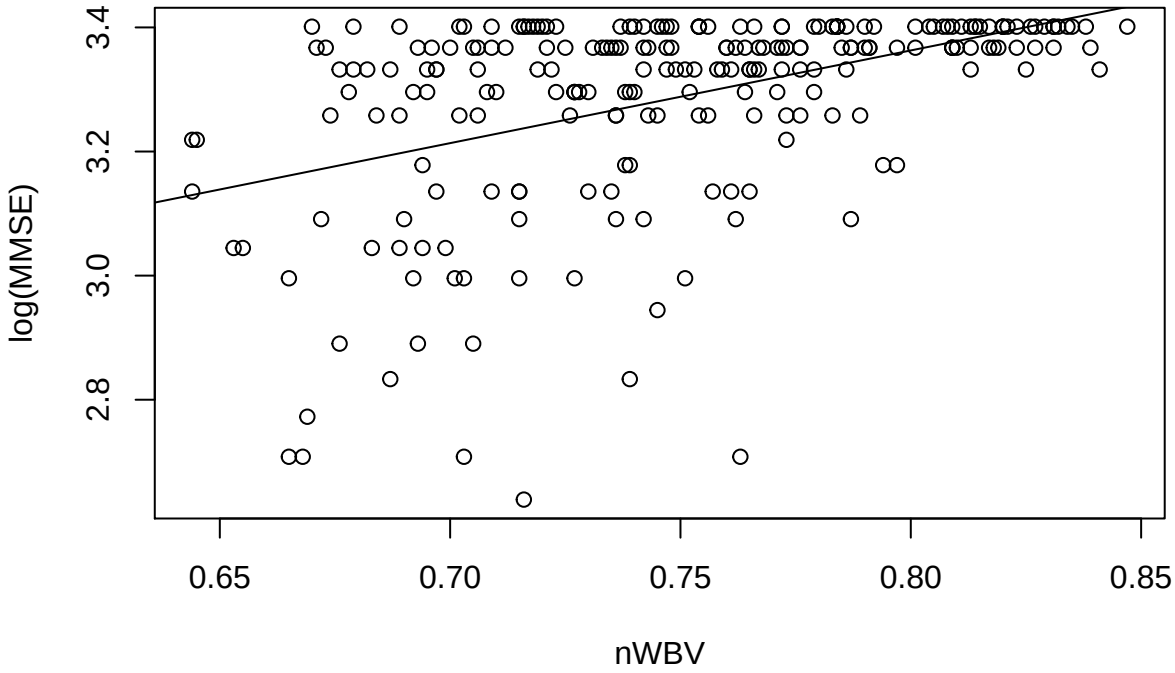

```
R_squared_eTIV_MMSE_log <- plot_logarithmic(dataFrame[,c(7,9)], "eTIV", "MMSE")
```




```
R_squared_nWBV_MMSE_lin <- plot_linear(dataFrame[,c(7,10)], "nWBV", "MMSE")
```




```
R_squared_nWBV_MMSE_log <- plot_logarithmic(dataFrame[,c(7,10)], "nWBV", "MMSE")
```




```
writeLines(c(
  paste0("R_squared_eTIV_MMSE_lin: ", R_squared_eTIV_MMSE_lin),
  paste0("R_squared_eTIV_MMSE_log: ", R_squared_eTIV_MMSE_log),
  paste0("R_squared_nWBV_MMSE_lin: ", R_squared_nWBV_MMSE_lin),
  paste0("R_squared_nWBV_MMSE_log: ", R_squared_nWBV_MMSE_log)
))
```

```
## R_squared_eTIV_MMSE_lin: 3.27957081573072e-05
## R_squared_eTIV_MMSE_log: 1.67342742232671e-06
## R_squared_nWBV_MMSE_lin: 0.220523157737581
## R_squared_nWBV_MMSE_log: 0.204735497944112
```

There is no significant linear or nonlinear correlation between eTIV and MMSE. There is a weak linear correlation between nWBV and MMSE. In previous tests, trying to calculate a nonlinear correlation between nWBV and MMSE resulted in an error, thus there is no significant nonlinear correlation between nWBV and MMSE.

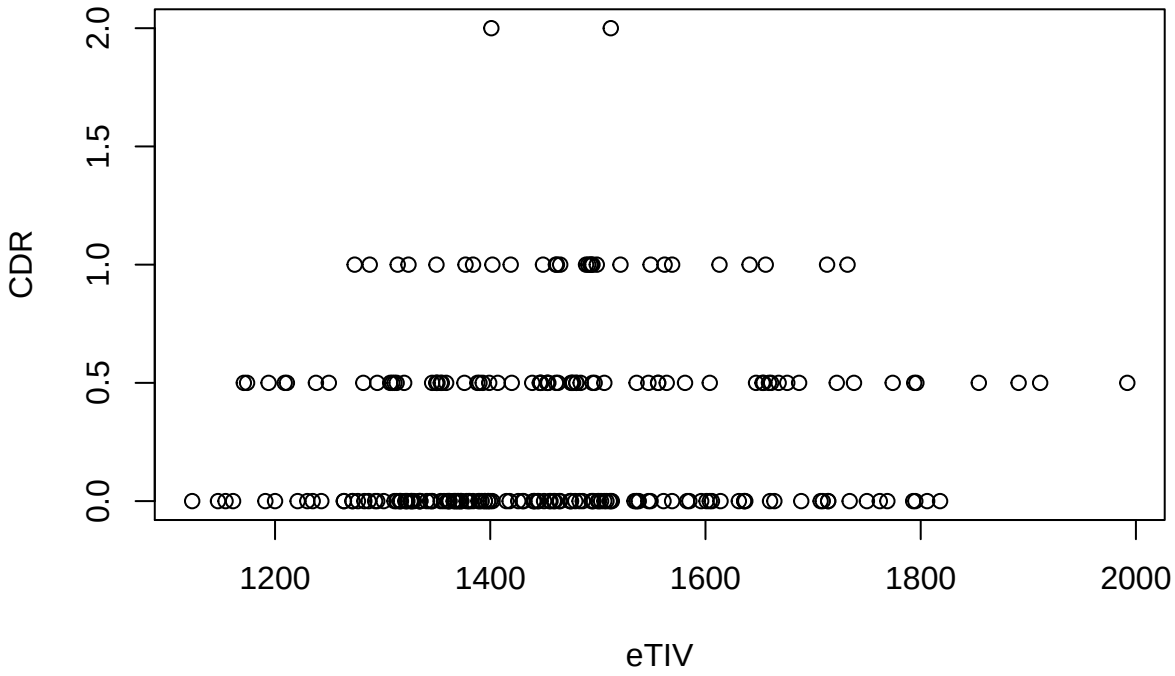
Part 2: Further inspection

We then used some of our categorical variables, such as SES (Socio-economic status) and CDR (Clinical Dementia Rating) to find new associations in our dataset using the variance form of the Kruskal-Wallis H Test(Kruskal and Wallis 1952).

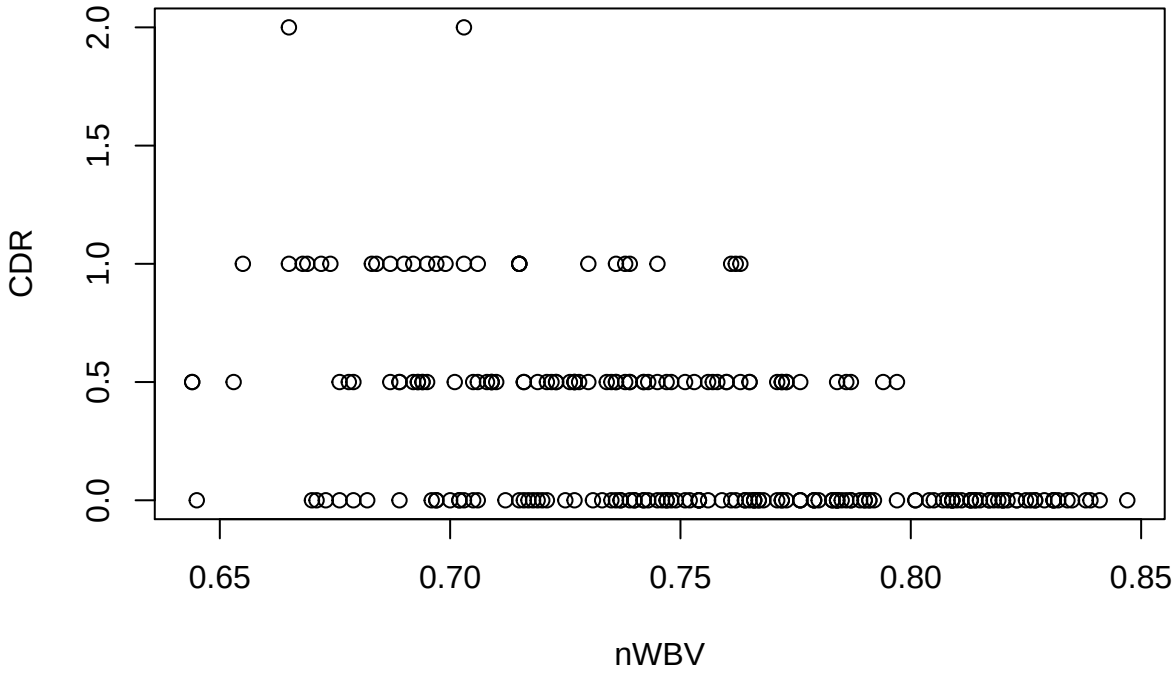
```
# http://www.sthda.com/english/wiki/one-way-anova-test-in-r
# http://onetipperday.sterding.com/2015/08/using-anova-to-get-correlation-between.html

anova_correlation <- function(columns, name1, name2) {
  neededData = na.omit(columns)
  plot(neededData[,name1], neededData[,name2], xlab=name1, ylab=name2)
  model <- aov(neededData[,name2] ~ neededData[,name1])
  summary.lm(model)$r.squared
}
```

```
R_squared_eTIV_CDR_anova <- anova_correlation(dataFrame[,c(8,9)], "eTIV", "CDR")
```




```
R_squared_nWBV_CDR_anova <- anova_correlation(dataFrame[,c(8,10)], "nWBV", "CDR")
```




```
writeLines(c(
  paste0("R_squared_eTIV_CDR_anova: ", R_squared_eTIV_CDR_anova),
  paste0("R_squared_nWBV_CDR_anova: ", R_squared_nWBV_CDR_anova)
))
```

```
## R_squared_eTIV_CDR_anova: 0.0110678538754664
## R_squared_nWBV_CDR_anova: 0.251015265902633
```

Neural Network

Given the lack of statistically significant assocations between any pair of variables (regardless of the type of variable and the method of assocation), we hypothesized that some combination of multiple variables can meaningfully predict the patient clinical dementia rating (CDR) or patient MMSE score. Below is a neural network written in the python programming language that predicts a patient's (CDR) based on their patient profile, which consists of their gender, age, education level, socio-economic status, MMSE score, estimated total intracranial volume, and normalized whole brain volume. To change which variable is being predicted by this neural network, the program uses a variable called mode (defined on line 127).

```
# Sources:
# https://www.tensorflow.org/tutorials/load_data/csv
# https://www.youtube.com/watch?v=JdagdilOFIw
# https://medium.com/themlblog/splitting-csv-into-train-and-t #est-data-1407a063dd74
# https://www.tensorflow.org/tutorials/load_data/pandas_dataframe
# https://stackoverflow.com/questions/26414913/normalize-columns-of-pandas-data-frame
# https://stackoverflow.com/questions/41925157/logisticregression-unknown-label-type-continuous-using-s
# https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
# https://scikit-learn.org/stable/modules/tree.html

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import MinMaxScaler, StandardScaler, LabelEncoder
from sklearn.model_selection import train_test_split

#mode = ('MMSE', 4)
mode = ('CDR', 5)
# change this to change which variable is being predicted and it's column index.

def get_dataset(file_path):
    data = pd.read_csv(file_path) \
             .drop('Delay', axis=1) \
             .drop('ID', axis=1) \
             .drop('ASF', axis=1) \
             .drop('Hand', axis=1)

    data.dropna(inplace=True)
    data['M/F'] = pd.Categorical(data['M/F'])
    data['M/F'] = data['M/F'].cat.codes

    values = data.values
    values_scaled = MinMaxScaler().fit_transform(values)
    data = pd.DataFrame(values_scaled)

    y = data[mode[1]] # CDR
    X = data.drop(mode[1], axis=1) # everything except CDR
```

```
    lab_enc = LabelEncoder()
    y = lab_enc.fit_transform(y)

    return X, y

def test_models():
    X, y = get_dataset('oasis_cross-sectional.csv')
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1)

    classifiers = [DecisionTreeClassifier(),
                   RandomForestClassifier(n_estimators=1000),
                   KNeighborsClassifier(),
                   LinearRegression(),
                   LogisticRegression(),
                   SVC()]
    names = ['DecisionTreeClassifier',
             'RandomForestClassifier',
             'KNeighborsClassifier',
             'LinearRegression',
             'LogisticRegression',
             'SVC']
    scores = [[] for clf in classifiers]
    for i in range(len(classifiers)):
        clf = make_pipeline(StandardScaler(), classifiers[i])
        clf = clf.fit(X_train, y_train)
        for j in range(1000):
            scores[i] += [clf.score(X_test, y_test)]
    print('\n'.join(
              [name + ': ' + str(round(sum(score) * 1.0) / 1000.0) + '% accuracy'
              for name, score in zip(names, scores)]))

if __name__ == '__main__':
    test_models()
```

```
## DecisionTreeClassifier: 0.636% accuracy
## RandomForestClassifier: 0.727% accuracy
## KNeighborsClassifier: 0.682% accuracy
## LinearRegression: 0.552% accuracy
## LogisticRegression: 0.636% accuracy
## SVC: 0.727% accuracy
```

The following machine learning models were tested on our dataset: a decision tree, a random forest with 1000 trees, a K Nearest Neighbors classifier, a linear regression model, a logistic regression model, and a support vector machine. All of these models are implemented by the python library scikit-learn. Each model was tested 1000 times on the testing data, and although there was a significant amount of variance in the scores of each model, the best and most consistent models averaged close to 80% accuracy each time.

Conclusion

In the future, we hope to test these neural network models on a more robust dataset with more patients, as well as find new correlations that can influence future predictors.

References

Assocation, Alzheimer's. n.d. "2020 Alzheimer's Disease Facts and Figures."

Cummings, Jeffrey, Garam Lee, Aaron Ritter, Marwan Sabbagh, and Kate Zhong. 2020. "Alzheimer's Disease Drug Development Pipeline: 2020." *Alzheimer's & Dementia: Translational Research & Clinical Interventions* 6 (1): e12050. https://doi.org/https://doi.org/10.1002/trc2.12050.

Kruskal, William H., and W. Allen Wallis. 1952. "Use of Ranks in One-Criterion Variance Analysis." *Journal of the American Statistical Association* 47 (260): 583–621. https://doi.org/10.1080/01621459.1952.10483441.