**Introduction**

**Description**

Alzheimer's disease is a brain disorder that affects memory, thinking, and behavior. These impairments include memory, problem solving, multitasking, executive functions, etc. We wanted to focus on this topic because Alzheimer’s is the most common form of dementia, accounting for more than ⅔ of dementia cases in people aged 65 or older, approximately 6.5 million people in 2022, and is the 6th in leading causes of death in the U.S. By 2050, this number is projected to reach 13 million. Although many pharmaceutical companies are attempting to develop cures and treatments for Alzheimer’s, there is still no cure, only treatments for the symptoms of Alzheimer’s. The purpose of this study is to investigate the most significant factors that cause Alzheimer’s patients to become demented, and predict whether patients will become demented. By identifying the most significant factors, it will allow researchers for more effective development of targeted interventions to reduce the chances of patients to become demented, and is very important to the development of effective prevention, treatments, and cures.

**Research Questions**

1. What factors contribute to the patients’ degenerative Alzheimer's condition?
2. Does the volume of the brain correlate to the degenerative of an Alzheimer patient’s condition?

**Dataset Description**

<https://www.kaggle.com/code/techlyx/alzheimer-features-data-exploration-yx-knn/data>

The raw dataset contains 370 patients with 10 unique attributes. The dataset is from kaggle.com, and we found it as the most complete and relevant dataset to what we wanted to research on among the datasets we looked at.

**Group:** Nondemented, demented, or converted (nondemented converted to demented)

**Gender:** Male or Female

**Age:** Age of patient

**EDUC:** Years of education of the patient

**Mini Mental State Examination:** A 30 point questionnaire to measure cognitive impairment (25+ is normal, 21-24 shows mild signs, 10-20 shows moderate signs, Under 10 shows severe impairment)

**Estimated Total Intracranial Volume:** Volume of the cranial cavity (Space around the brain) and gives rough estimation of brain size. For context: Men: 1469 +/- 102 cm^3, Women: 1289 +/- 111 cm^3.

**Normalize Whole Brain Volume:** For context: Men: 1262 mL ± 113, Women: 1123 mL ± 810. Normalized Volumes, Men: 0.84 ± 0.05, Women: 0.85 ±0.05

**Atlas Scaling Factor:** The volume-scaling factor required to match each individual to the atlas target.

**Motivation**

We wanted to study Alzheimer’s disease not only because it is so common, but also because we have been personally affected by it. We both have relatives and/or friends who have suffered from Alzheimer’s, dementia, or chronic diseases that slowly killed them. Everyday, we could see the mental toll the disease had on them, forgetting their loved ones and friends one by one, and all we could do was wish a cure or better treatment existed. We were motivated to study the factors that contributed to the acceleration of Alzheimer’s disease because we understood how important living one more day is to the family of a patient with a chronic disease and we can’t even begin to imagine the pain of being forgotten by a loved one.

#Importing Data Set

patients = pd.read\_csv("alzheimer.csv")

# 'Group','M/F',

#Dropping unwanted columns

patients = patients.drop(['SES','CDR'],axis=1)

#Dropping specific patients with Converted as their Group variable

patients = patients[patients['Group'].str.contains("Converted") == False]

#Displaying current dataset after changes

patients.head()

#Changing Males and Females to numeric values

mapping\_MF = {'M': 0, 'F': 1}

#Changing Demented and Nondemented patients to numeric values

mapping\_Group = {'Nondemented':0, 'Demented':1}

#Changing non numeric value variables to numbers

patients = patients.replace({'Group':mapping\_Group, 'M/F': mapping\_MF})

#Drop all rows with missing values

patients = patients.dropna()

#Normalizing all the data within the Age Column

age = patients[['Age']].values

min\_max\_scaler = preprocessing.MinMaxScaler()

age\_scaled = min\_max\_scaler.fit\_transform(age)

patients['Age'] = pd.DataFrame(age\_scaled)

#Normalizing all the data within the Years of Education Column

education = patients[['EDUC']].values

education\_scaled = min\_max\_scaler.fit\_transform(education)

patients['EDUC'] = pd.DataFrame(education\_scaled)

#Normalizing all the data within the Mini Mental State Examination Column

mental\_state = patients[['MMSE']].values

mmse\_scaled = min\_max\_scaler.fit\_transform(mental\_state)

patients['MMSE'] = pd.DataFrame(mmse\_scaled)

#Normalizing all the data within the estimated Total Intracranial Volume Column

intracranial\_v = patients[['eTIV']].values

eTIV\_scaled = min\_max\_scaler.fit\_transform(intracranial\_v)

patients['eTIV'] = pd.DataFrame(eTIV\_scaled)

#Normalizing all the data within the Atlas Scaling Factor Column

atlas\_scaling = patients[['ASF']].values

ASF\_scaled = min\_max\_scaler.fit\_transform(atlas\_scaling)

patients['ASF'] = pd.DataFrame(ASF\_scaled)

#Just in case drop all new rows with missing values

patients = patients.dropna()

#Normalized Whole Brain Volume, doesn't need to be normalized

**Data and Data Cleaning**

Before working with the data, we had to clean it. We removed “Converted” because the variable was unclear, SES and CDR because the data was too subjective and potentially biased, and all attributes with missing values. We also normalized all remaining data to a common scale (0-1; 0 is minimum & 1 is maximum) to prepare the data for machine learning. Finally, we changed string values into numerical values. Below is the dataset summary before normalizing.

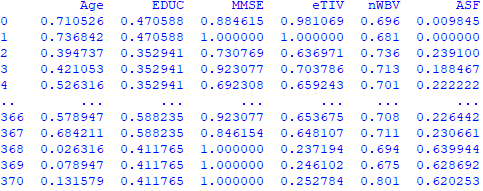
|  | Group | M/F | Age | EDUC | MMSE | eTIV | nWBV | ASF |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mean | 0.438 | 0.5687 | 77.01 | 14.61 | 27.34 | 1489.87 | 0.7295 | 1.19 |
| STD | 0.4709 | 0.4959 | 7.66 | 2.88 | 3.68 | 174.99 | 0.0372 | 0.1365 |
| Min | 0 | 0 | 60 | 6 | 4 | 1106 | 0.644 | 0.876 |
| 25% | 0 | 0 | 71 | 12 | 27 | 1358 | 0.7 | 1.1 |
| 50% | 0 | 1 | 77 | 15 | 29 | 1471 | 0.729 | 1.19 |
| 75% | 0 | 1 | 82 | 16 | 30 | 1598 | 0.756 | 1.29 |
| Max | 1 | 1 | 98 | 23 | 30 | 2004 | 0.837 | 1.59 |

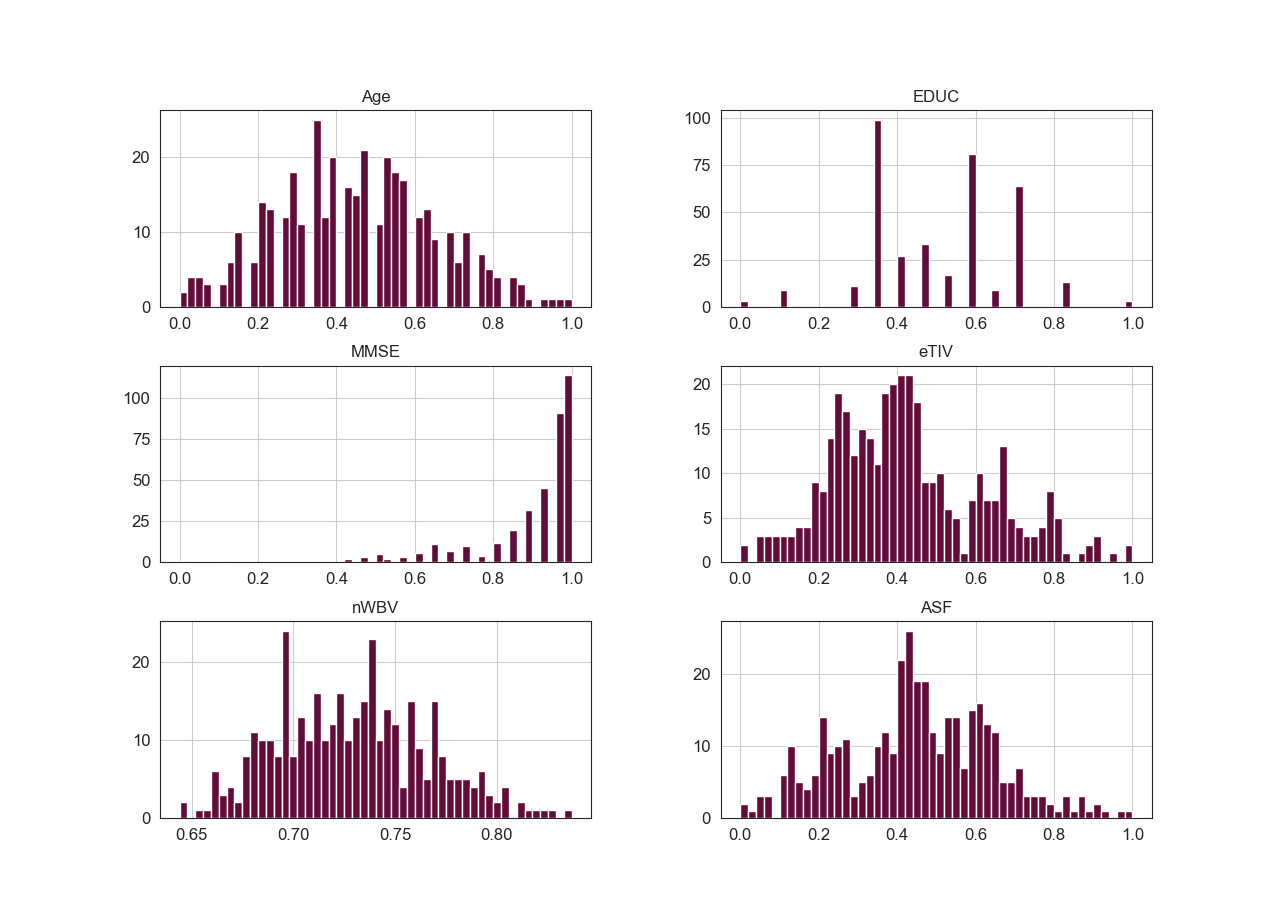
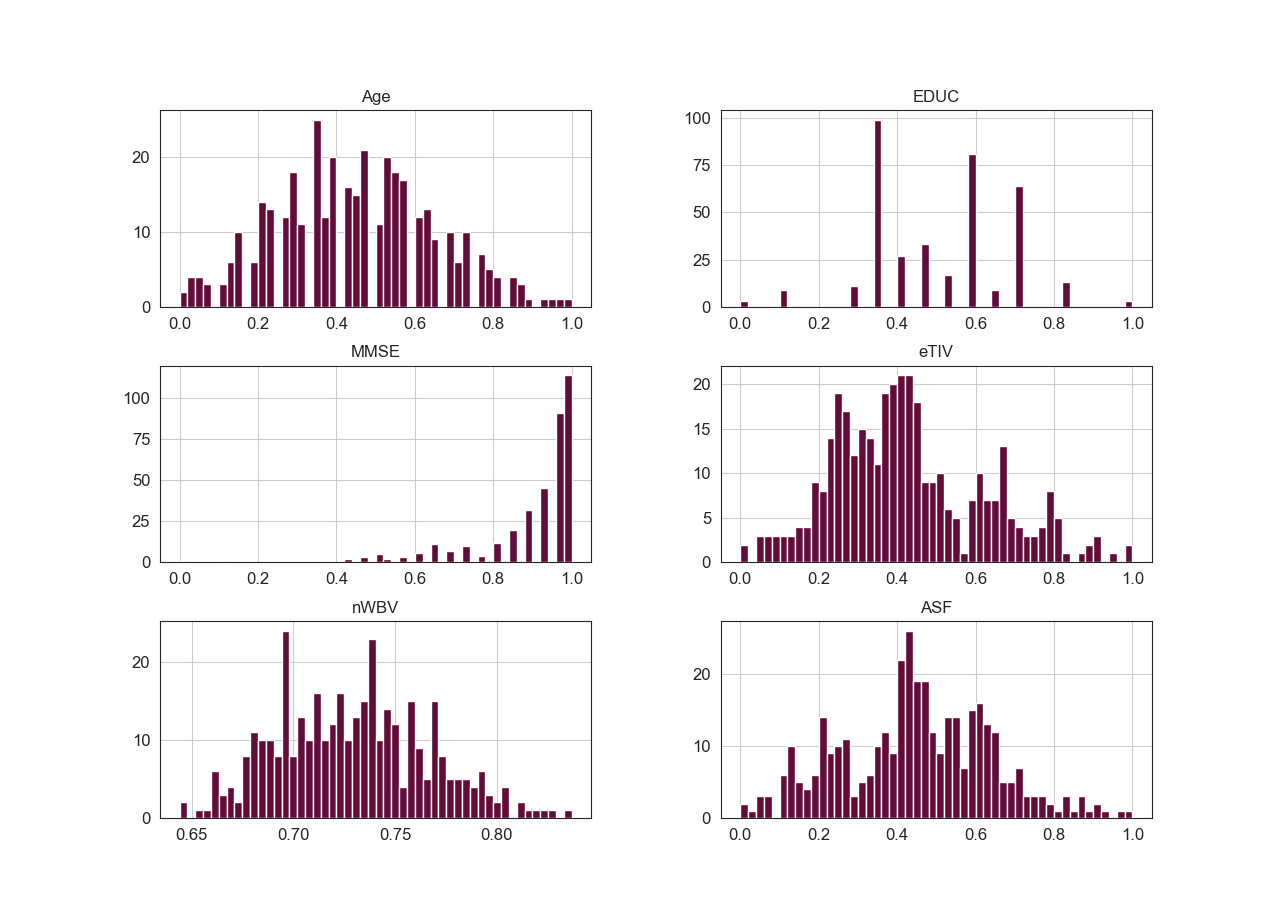
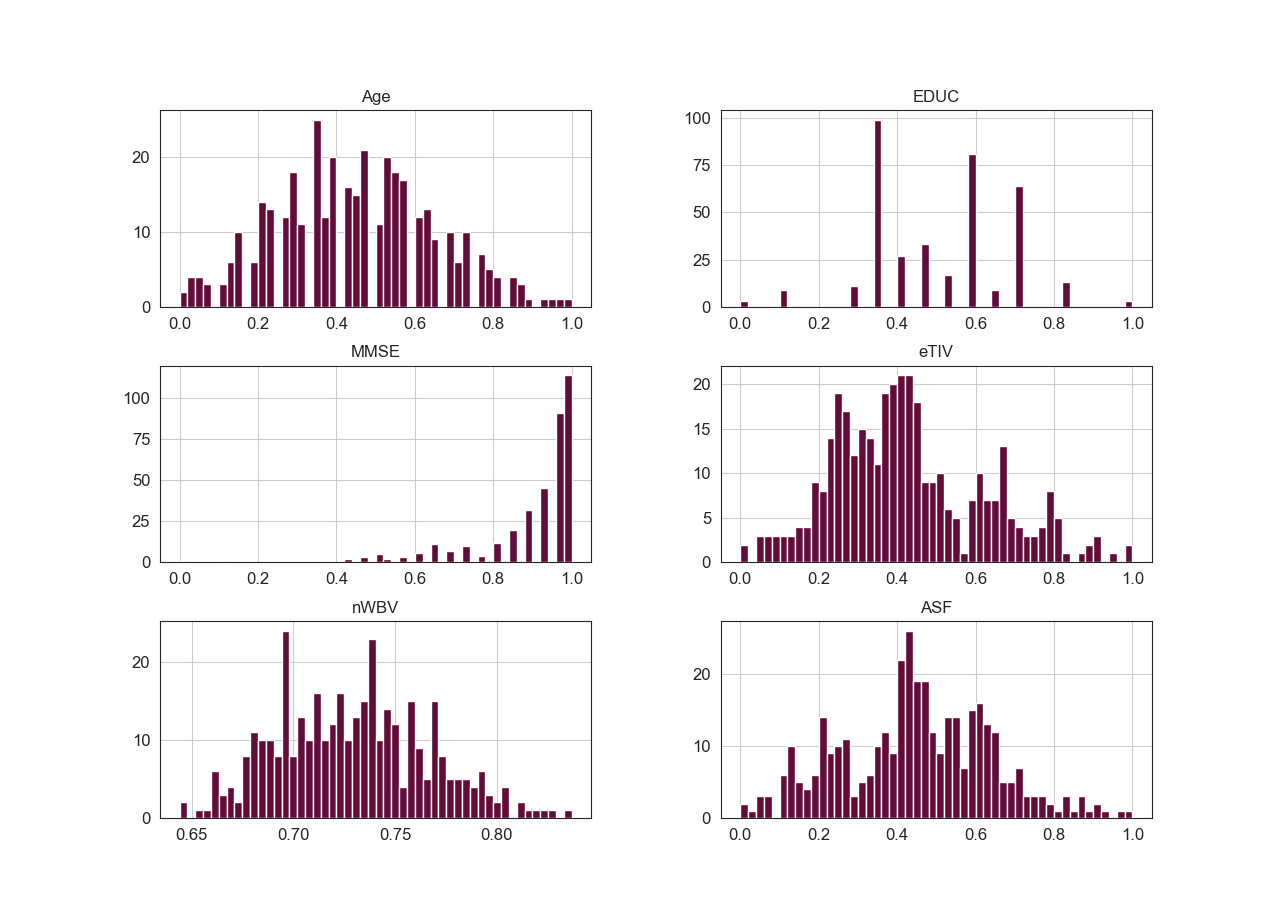
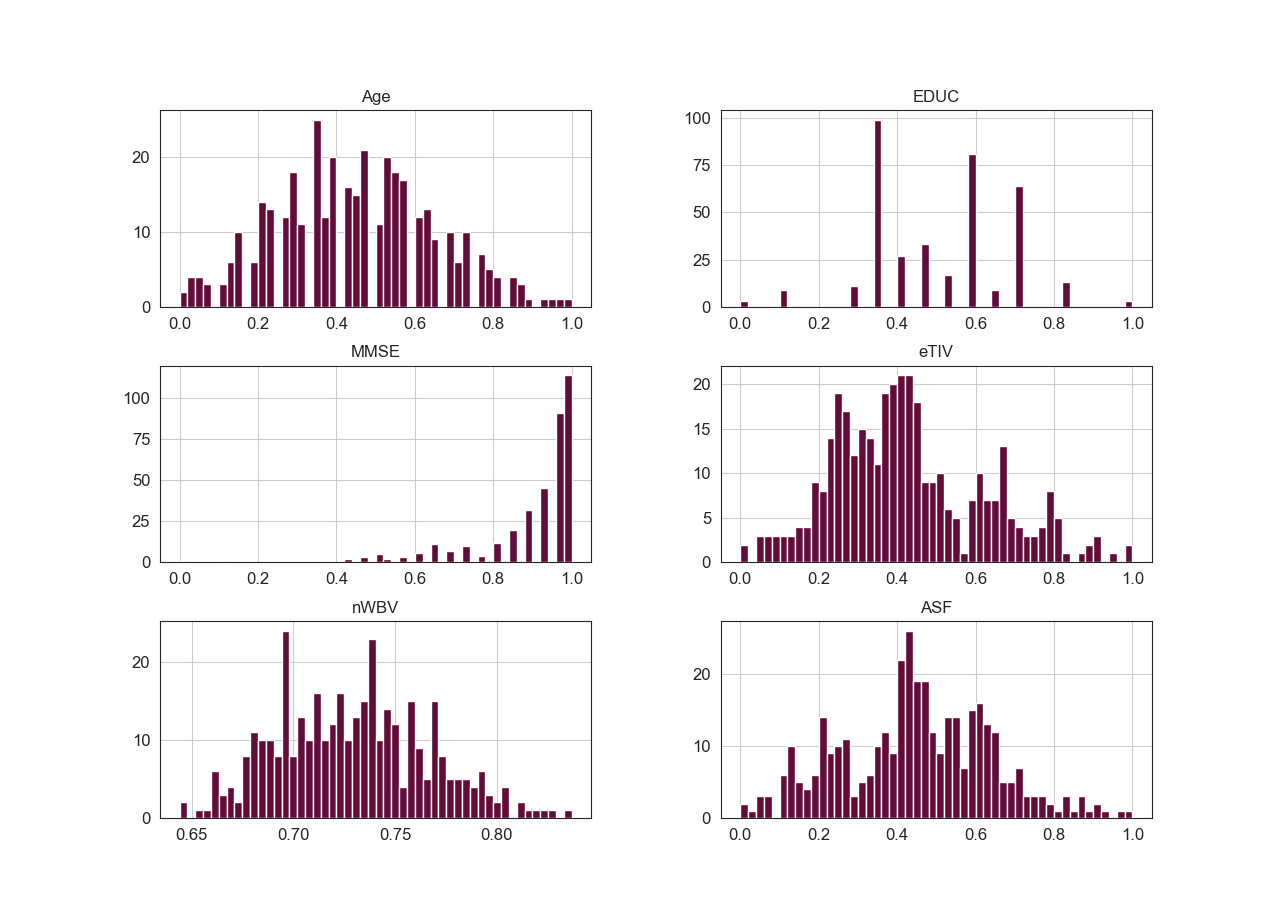
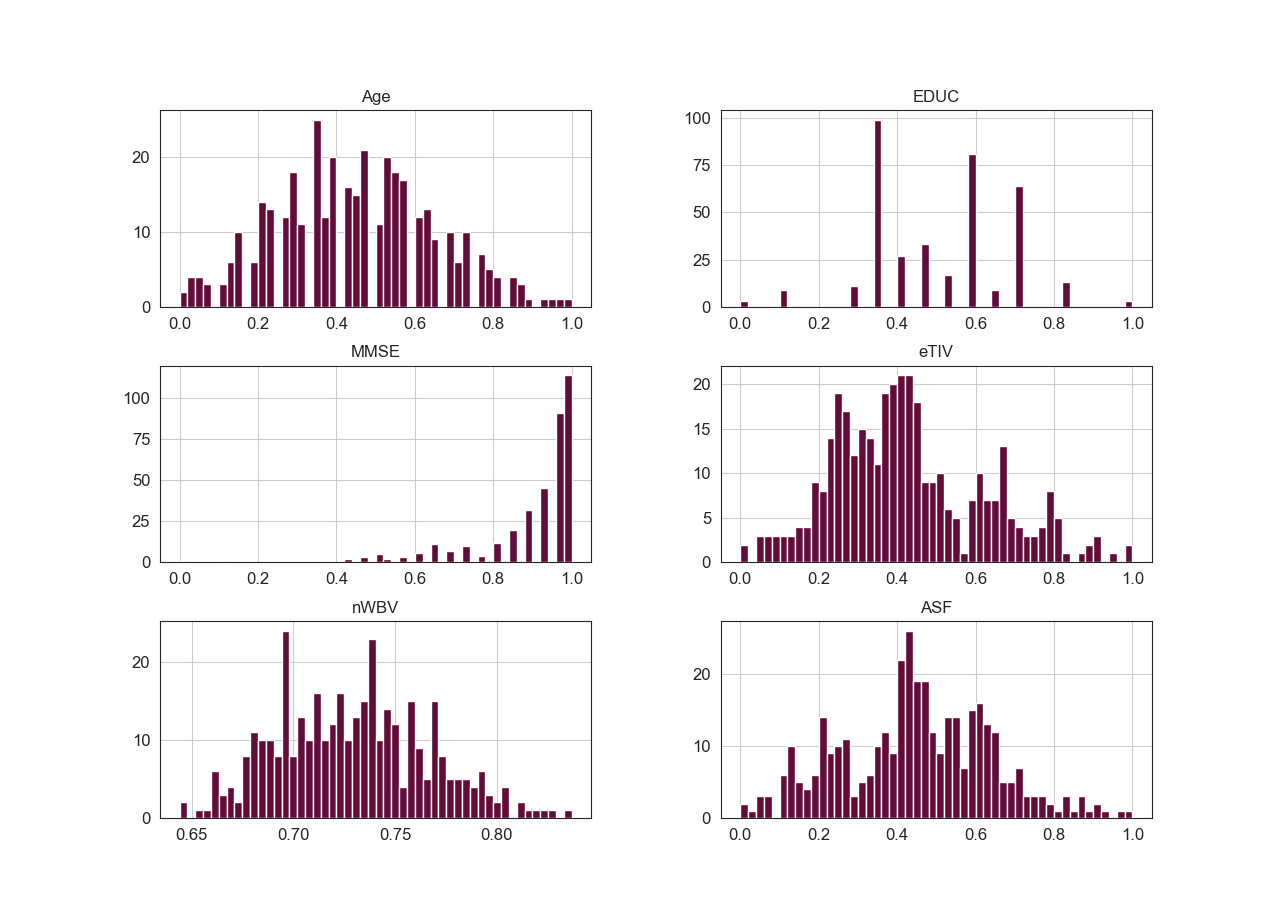
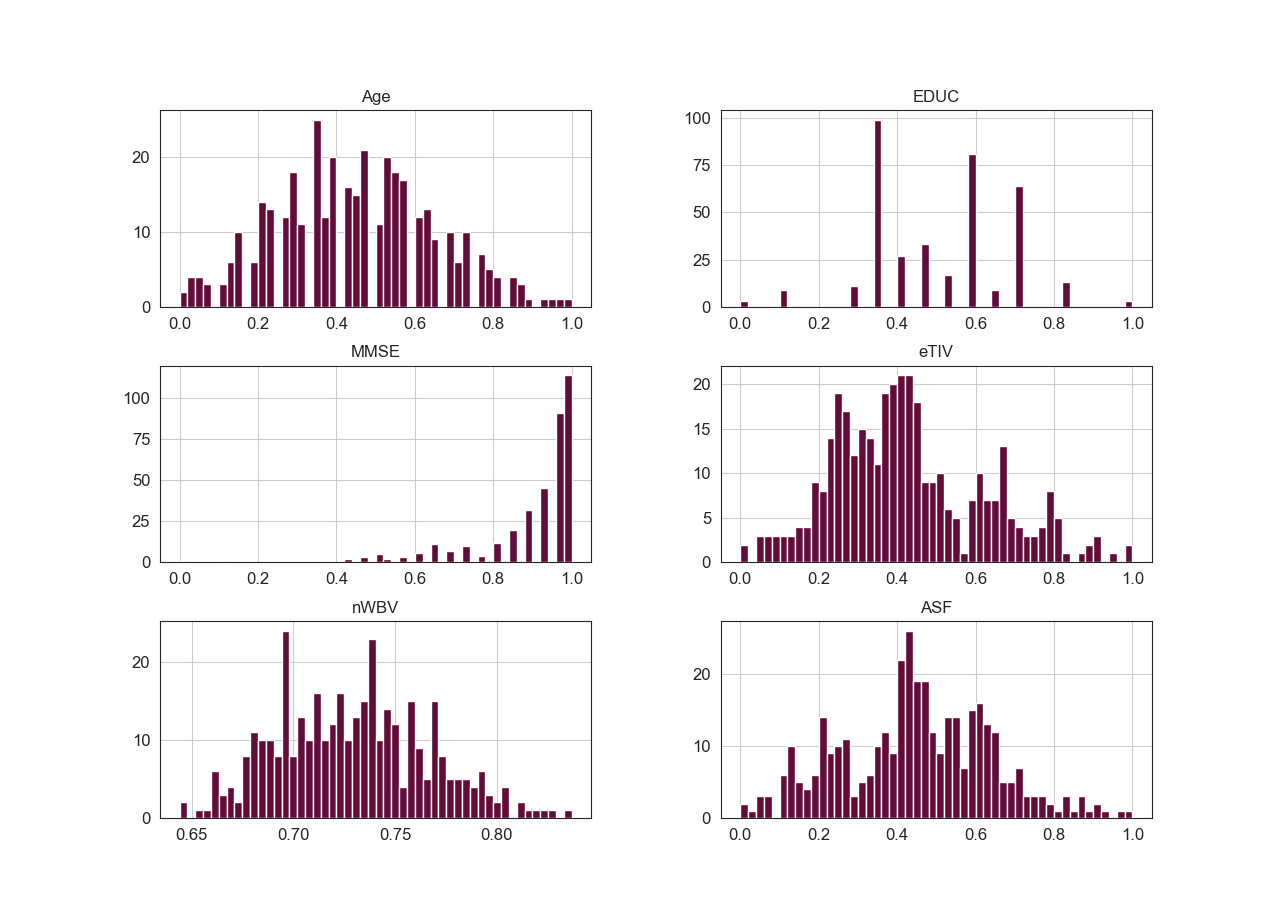
#Plot all the columns in the patients data set

patients.hist(bins=50,figsize=(20,15))

plt.show()

**Graphical Visualizations Dataset Summary of Normalized Data**

****



**K-Nearest Neighbor**

from sklearn.cluster import KMeans

#Finding the Sum of the squared distances for all the cluster from 1-14

Sum\_of\_squared\_distances = []

K = range(1,15)

for k in K:

km = KMeans(n\_clusters=k)

km = km.fit(patient\_aspects)

Sum\_of\_squared\_distances.append(km.inertia\_)

#Finding the silhouette score for all the cluster from 2-14

from sklearn.metrics import silhouette\_score

for n\_clusters in range(2,15):

clusterer = KMeans (n\_clusters=n\_clusters)

preds = clusterer.fit\_predict(patient\_aspects)

centers = clusterer.cluster\_centers\_

#Uses euclidean distance to score

score = silhouette\_score (patient\_aspects, preds, metric='euclidean')

print ("For n\_clusters = {}, silhouette score is {})".format(n\_clusters, score))

#Plotting the graph for the Elbow method

plt.plot(K, Sum\_of\_squared\_distances, 'gx-')

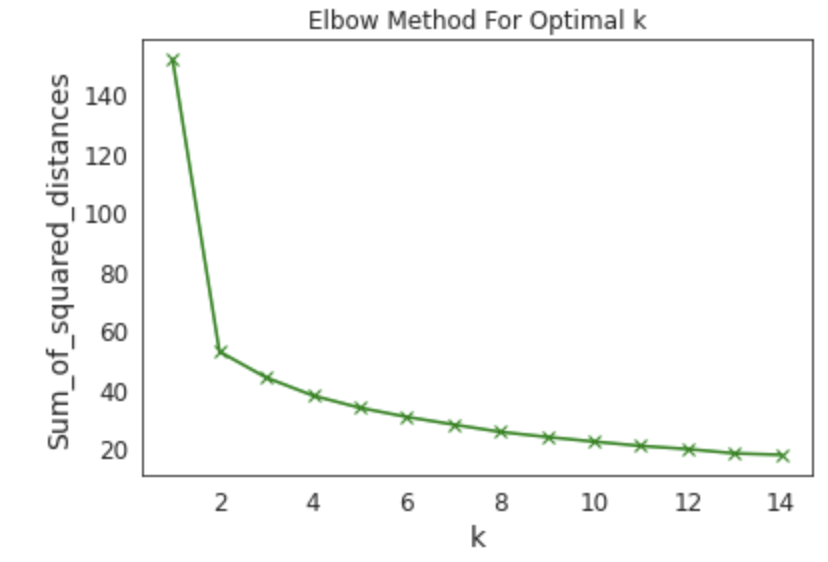
plt.xlabel('k')

plt.ylabel('Sum\_of\_squared\_distances')

plt.title('Elbow Method For Optimal k')

plt.show()

**Elbow Graph**

****

**Silhouette Score**

| # of Clusters | Silhouette Score |
| --- | --- |
| 2 | 0.5778 |
| 3 | 0.4646 |
| 4 | 0.2635 |
| 5 | 0.2603 |
| 6 | 0.2523 |
| 7 | 0.2555 |
| 8 | 0.2605 |

To get our optimal K-value, we looked for a higher silhouette score and a part of the elbow graph where it starts flattening out. So using both graphs, we decided to go 2 as the K-value,

# Create KNN classifier

knn = KNeighborsClassifier(n\_neighbors = 2)

# Fit the classifier to the data

knn.fit(X\_train,y\_train)

#show first 5 model predictions on the test data

knn.predict(X\_test)[0:5]

#check accuracy of our model on the test data

knn.score(X\_test, y\_test)

from sklearn.model\_selection import cross\_val\_score

import numpy as np

#create a new KNN model

knn\_cv = KNeighborsClassifier(n\_neighbors=2)

#train model with cv of 5

cv\_scores = cross\_val\_score(knn\_cv, X, y, cv=5)

#print each cv score (accuracy) and average them

print(cv\_scores)

print('cv\_scores mean:{}'.format(np.mean(cv\_scores)))

from sklearn.model\_selection import GridSearchCV

#create new a knn model

knn2 = KNeighborsClassifier()

#create a dictionary of all values we want to test for n\_neighbors

param\_grid = {'n\_neighbors': np.arange(1, 25)}

#use gridsearch to test all values for n\_neighbors

knn\_gscv = GridSearchCV(knn2, param\_grid, cv=5)

#fit model to data

knn\_gscv.fit(X, y)

#check top performing n\_neighbors value

knn\_gscv.best\_params\_

#check mean score for the top performing value of n\_neighbors

knn\_gscv.best\_score\_

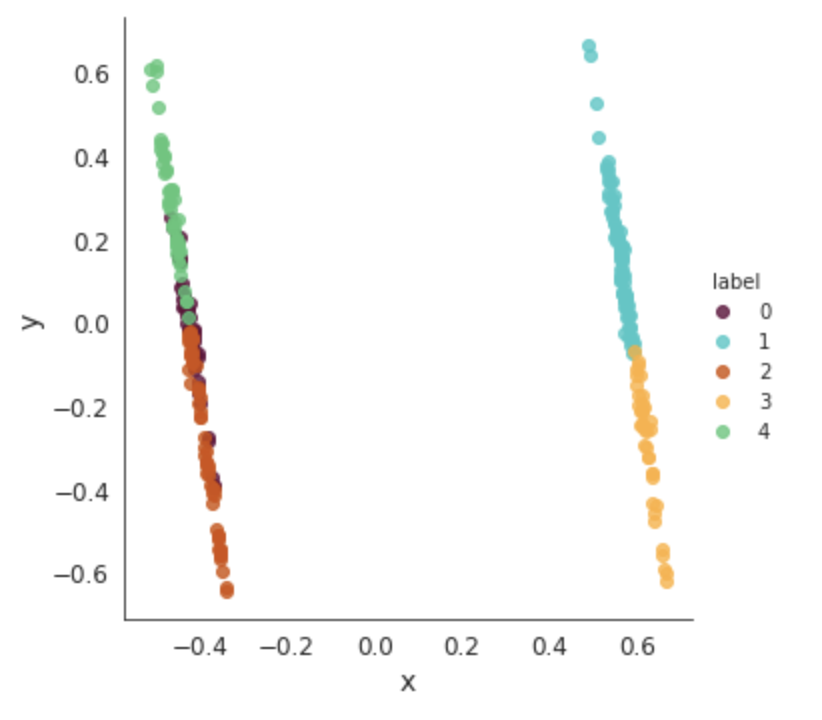
However, we later decided we wanted to better optimize our accuracy after running our code. Instead of using the holdout method which does a standard 80/20 split on the training and testing data, we decided to use k-fold cross validation. Cross-validation randomly splits the dataset up into ‘k’ groups. One of the groups is used as the test set and the rest are used as the training set. Cross-validation gives the model an opportunity to test on multiple splits so we can get a better idea on how the model will perform on unseen data. Due to our dataset being generally small we can afford to use k-fold cross validation which takes more computational power and time to run than using the holdout method.

Furthermore, using GridSearchCV we can train our model multiple times on a range of parameters that we specify. This way, we can test our model with each parameter and figure out the optimal values to get the best accuracy results. Which ends up being 5.

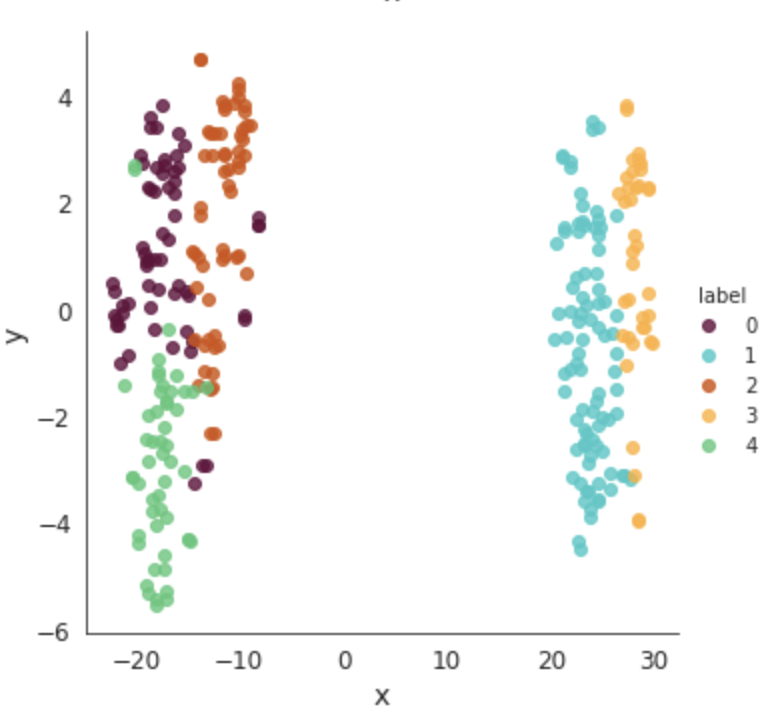
**2D Visualisation of Alzheimer Patient Clusters**

We created two patient cluster graphs: one using Principal component analysis (PCA) and one using t-Distributed Stochastic Neighbour Embedding (t-SNE). PCA will preserve large distances between points, while tSNE will preserve points which are close to each other in its representation. PCA is also a deterministic algorithm (produces same output each time), whereas t-SNE is non deterministic in nature (may produce different output each time). Below are the clusters and the cluster summaries.

**PCA Cluster**



**t-SNE**

****

#Printing out 10 of the bottom data points in each cluster and the mean of each cluster

print(patients[patients['label'] == 0].tail(10))

print(patients\_2[patients\_2['label'] == 0].mean())

print(patients[patients['label'] == 1].tail(10))

print(patients\_2[patients\_2['label'] == 1].mean())

print(patients[patients['label'] == 2].tail(10))

print(patients\_2[patients\_2['label'] == 2].mean())

print(patients[patients['label'] == 3].tail(10))

print(patients\_2[patients\_2['label'] == 3].mean())

print(patients[patients['label'] == 4].tail(10))

print(patients\_2[patients\_2['label'] == 4].mean())

**Cluster Summary**

| Cluster | 0 | 1 | 2 | 3 | 4 |
| --- | --- | --- | --- | --- | --- |
| # | 62 | 89 | 59 | 38 | 52 |
| Group | 0.387 | 0.5843 | 0.2712 | 0.5789 | 0.2885 |
| M/F | 1 | 0 | 1 | 0 | 1 |
| Age | 0.2568 | 0.4137 | 0.5616 | 0.4813 | 0.5111 |
| EDUC | 0.4763 | 0.4964 | 0.5912 | 0.5124 | 0.4276 |
| MMSE | 0.8424 | 0.8933 | 0.9374 | 0.8694 | 0.926 |
| eTIV | 0.3946 | 0.3662 | 0.5875 | 0.7113 | 0.2028 |
| nWBV | 0.7489 | 0.7183 | 0.7357 | 0.7186 | 0.7229 |
| ASF | 0.4655 | 0.4952 | 0.2894 | 0.1882 | 0.692 |

model = Ridge(alpha=1e-2).fit(X\_train, y\_train)

model.score(X\_val, y\_val)

from sklearn.inspection import permutation\_importance

r = permutation\_importance(model, X\_val, y\_val,n\_repeats=30,random\_state=0)

print(r.importances\_mean.argsort())

for i in r.importances\_mean.argsort():

print(f"{r.importances\_mean[i]:.3f}"f" +/- {r.importances\_std[i]:.3f}")

**Feature Importance**

| nWBV | Age | ASF | EDUC | M/F | eTIV | MMSE |
| --- | --- | --- | --- | --- | --- | --- |
| -0 +/- 0.052 | 0.021 +/- 0.031 | 0.028 +/- 0.024 | 0.03 +/- 0.031 | 0.09 +/- 0.049 | 0.153 +/- 0.063 | 0.472 +/- 0.1 |

#Remove target variable

X = patients.drop(columns=['Group'])

#Set y to target variable

y = patients['Group']

#Split the data into 80/20 split for training and testing,

#stratify=y ensures that our splits will always be consistent 80/20 split of patients with Dementia and patients without Dementia

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size = 0.2, random\_state = 1, stratify = y)

#Checking number of y\_tests

print(len(y\_test))

classifier = KNeighborsClassifier(n\_neighbors=5)

classifier.fit(X\_train, np.ravel(y\_train,order='C'))

y\_pred = classifier.predict(X\_test)

print(len(y\_test),len(y\_pred))

#Making Confusion Matrix

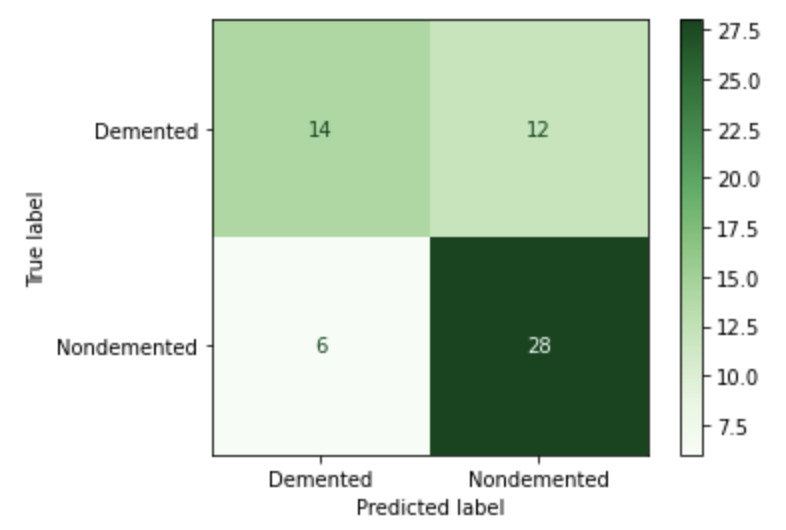
cm = confusion\_matrix(y\_test, y\_pred, labels=classifier.classes\_)

#Plotting Confusion Matrix

disp = ConfusionMatrixDisplay(confusion\_matrix=cm,display\_labels=classifier.classes\_)

disp = disp.plot(cmap=plt.cm.Greens,values\_format='g')

**Confusion Matrix**

****

This results from the confusion matrix show that our model correctly predicted 60 patients’ current condition, 14 being nondemented, and 28 being demented. This means our model has a 70% accuracy rate within our own data set, which definitely could have been improved if we removed normalized Whole Brain Volume.

**Conclusion**

To answer our first question, the factors most important to the degenerative condition, the most significant factor is the Mini Mental State Examination. Some other significant factors according to our data are the Atlas Scaling Factor, And close behind it, the Estimated total intracranial volume. The gender factor is high, and we believe the reason for this is likely because of the skewed data: there are 14% more females than males in our data, and also because females do have a greater risk of developing dementia. They comprise 2/3rds of Alzheimer’s patients. To answer the second question, whether the volume of the brain correlates to the degenerative condition, according to our results, it is the least significant factor out of the ones we studied, sitting at 0.03. This is likely because, looking at our data, the normalized brain volume standard deviation is 0.0372, and according to the National Center for Biotechnology Information, the standard deviation of brain volumes for both males and females is 0.05.

To recall what the Mini Mental State Examination is, it’s basically a screening tool that measures cognitive impairment, and our findings suggest that this tool may be useful in deciding whether individuals have dementia. The Atlas Scaling Factor and Estimated total intracranial volume basically measure the brain’s structure and volume, so our findings also suggest that changes in these areas are also important in whether the patient has dementia and could be targeted and focused on in treatments.

By identifying the significant factors that cause the development of dementia, more effective diagnostic tools and screening methods that can help identify individuals who have dementia can be created. More targeted and effective interventions and treatments can also be developed, aiming at preventing or delaying the dementia of Alzheimer's disease.

**Critique**

One critique is our second question dealing with the brain volume. A better way to measure whether brain volume has an impact on the patient’s condition is seeing the change or reduction in the brain volume, since this means we wouldn’t have to account for the fact that everyone’s brain volumes are different.

Another improvement is putting weighting on the factors now that we have a general idea of which factors are most significant. This would likely raise our accuracy of the model.

Finally, our data definitely did not have a large enough sample size. Typically, we want 10% of the population to be able to generalize our findings, however we only had approximately 300 attributes. In the US, there are 6.5 million people with Alzheimer’s, so technically, we would need 10% of 6.5 million people as our sample size.