Alzheimer's Disease Recognition With K-Nearest Neighbors

Camelia Siadat, Zachary Clark, Samantha Lee

Introduction

Alzheimer's Disease (AD) is a type of dementia that damages memory, thinking skills, and other mental activities. When a person has AD, the connections between the neurons associated with memory are destroyed. Furthermore, brain tissue shrinks and brain cells eventually die. The brain controls all of the body's functions. Early detection of Alzheimer's Disease is important because any deterioration or damage to the brain will cause significant negative impacts on a person's life. Early diagnoses and necessary treatments will allow Alzheimer's patients to adjust their lifestyles to manage their symptoms better. Treating this disease earlier gives people with Alzheimer's the ability to still live a very fulfilling life. A commonly used algorithm to detect Alzheimer's disease is a support vector machine. According to Rohini and Surendran [2], this algorithm performed on a dataset from the ADNI database (Alzheimer's Disease Neuroimaging Initiative) has proven to be highly accurate in predicting Alzheimer's disease and cognitive disability. The features used in this study include MRI data, positron emission tomography, genetics related to disease progression, demographic clinical data, and APOE genotype [2]. Our goal for this project is to use the K-Nearest Neighbors algorithm, and specifically chosen features, to predict the clinical dementia rating of sample data points.

Dataset

The Open Access Series of Imaging Studies (OASIS) provides multiple neuroimaging datasets to the public to allow researchers and scientists to learn and make discoveries in neuroscience. We chose to use the dataset "OASIS-1: Cross-sectional MRI Data in Young, Middle Aged, Nondemented and Demented Older Adults". This dataset consists of cross-sectional data on 416 subjects with ages ranging from 18 to 96. It also includes a reliability data set of 20 non-demented subjects. The data was derived from three or four MRI scans on each subject during a single imaging session. The reliability dataset was given a follow-up MRI scan after their initial imaging session. Given the MRI scans, calculations were performed to provide numerical information that would be used for measurement related to typical aging and Alzheimer's Disease [1]. The dataset includes demographic data, clinical data, and derived anatomic volumes. Demographic data contains features such as age and education. Clinical data includes mini-mental state examination and clinical dementia rating. Derived anatomic volumes

contain estimated total intracranial volume, atlas scaling factor, and normalized whole brain volume [1].

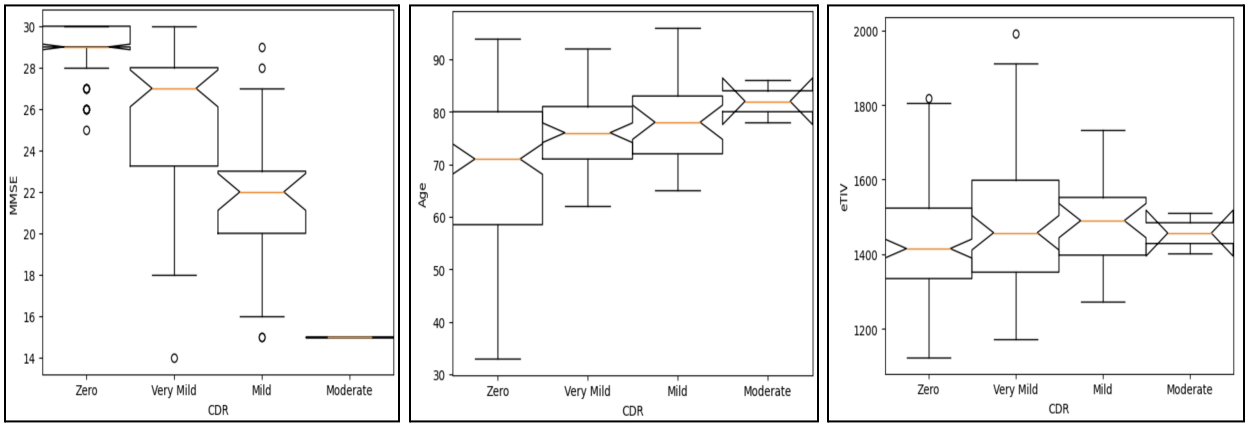
Design

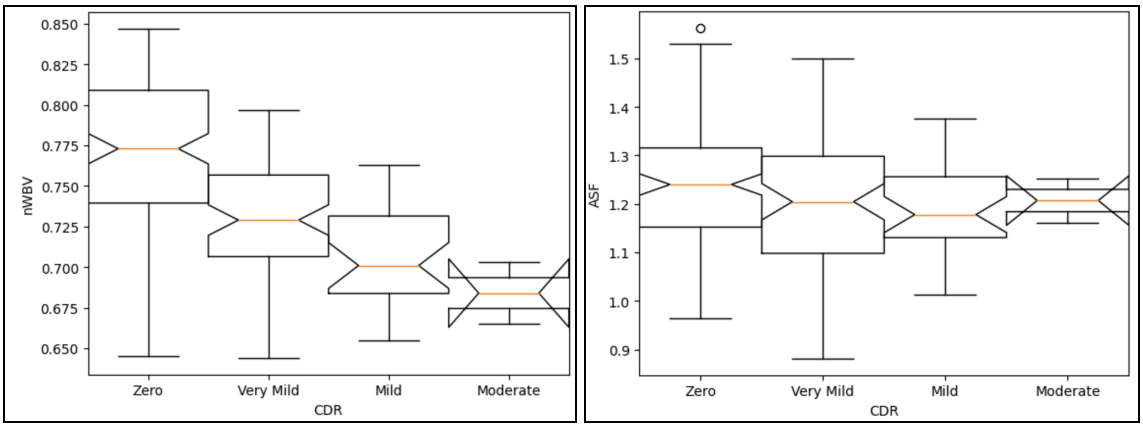
The design of our project involved identifying the learning task, setting up the experiment, and putting forth an objective and a hypothesis. The learning task involved predicting the clinical dementia rating, which is used for assessing the progression of Alzheimer's disease, based on MRI data. To set up the experiment, we needed to clean the data and normalize it so that it contributes equally to the distance computation. Additionally, we needed to resolve issues with our data including identifying features missing large amounts of data, taking note of potential biases in the use of demographic features, and dropping missing value rows for the target feature. Overall, our objective was to develop a KNN model capable of accurately predicting dementia status based on MRI scans. Our hypothesis is that KNN, a distance-based classifier, can effectively utilize features from our MRI data to classify individuals into different levels of dementia.

Feature Selection

Quantitative Features

To decide which numeric quantitative features to use for our KNN algorithm, we used the dataset.info() function to indicate which of these features were missing data, and to what extent these features were lacking data. From this output, we came to the following conclusion: the MMSE feature was missing 201 values and the Delay feature was missing 416 values. At this time, since less than 10% of the dataset included values for the Delay feature, we decided to drop this feature. Data visualization was then used to visualize the relationship between MMSE and CDR, Age and CDR, eTIV and CDR, nWBV and CDR, and ASF and CDR. The below graphs indicate the importance of including most of these features in our KNN classification.

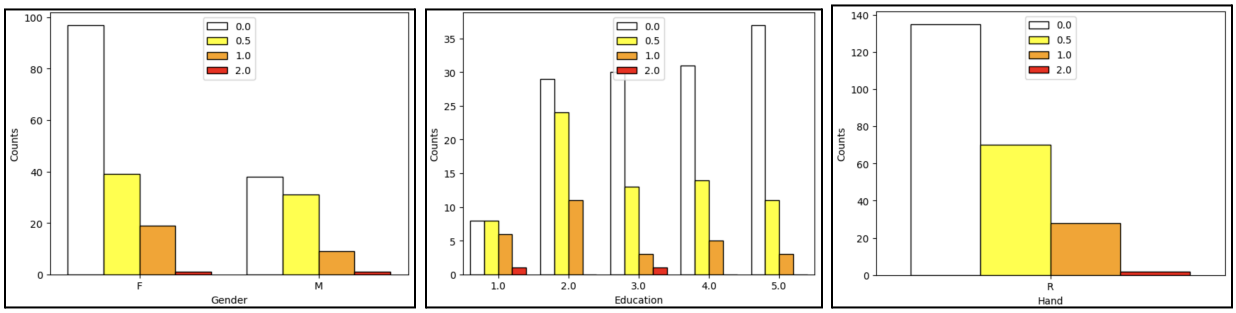




Figures 1-5 Graphs for Numeric Features

Qualitative Features

To decide which numeric qualitative features to use for our KNN algorithm, we needed to first identify the purpose of each kind of feature. Without needing to do any data visualization, we were able to rule out the ID categorical and nominal feature, as KNN would not be able to use this information for its classification process. To gather more information about the dataset, we printed the number of values present in each feature using the dataset.info() function. From this output, we came to the following conclusion: the education feature (Educ) was missing 201 values and the socioeconomic status feature (SES) was missing 220 features. Since the SES feature was missing more than half of its values, we decided to drop this feature as well. The next step was to use data visualization strategies to understand the relationship between M/F and CDR, Educ and CDR, and Hand and CDR. The below graphs indicate the potential importance of using the Educ numerical feature, as well as the futility of using the Hand (handedness) feature since all of the individuals in the study were right-handed. While gender at first glance appears to be an important feature for classification, further examination of the gender (M/F) feature reveals that 60% of the study participants were female, while only 40% of the study participants were male. This imbalance, then, required that we also drop the M/F feature from our dataset before classification.



Figures 6-8 Graphs for Demographic Features

Model Selection

We selected the KNN model for several reasons. The dataset's target variable is a multiclass variable, meaning that a classification technique for binary classification, such as logistic regression, would not fit the purposes of our project. Due to our large number of features with diverse ranges of values, we concluded that our data is most likely not linearly separable. Additionally, KNN exhibits simplicity and effectiveness in handling high-dimensional data. Experimentation with a multitude of varying values of K is essential to understanding what value of K best classifies new data points based on our selected features. KNN was also an appealing choice because it does not make assumptions about underlying data distribution. With appropriate preprocessing and feature selection, KNN can be effective in classifying individuals based on their MRI scans. Furthermore, KNN has straightforward interpretation and implementation.

Implementation

Preprocessing and Data Visualization

The first step in implementing our KNN algorithm was identifying and dropping extraneous features that would have no bearing on the final outcome. We dropped columns such as ID because it is a categorical nominal value and cannot be used in our algorithm. Upon visually inspecting the data, we noticed that Socioeconomic Status was missing over half of its values, so it could not be reliably used for classification. Upon further inspection and some visualization through the means of Figures 6-8, we also decided to drop Hand and M/F (gender) due to the values of these features being unevenly distributed. Using boxplots shown in Figures 1-5, we zoned in on features that we found to correlate with one another. After analyzing the boxplots, we found that Mini Mental State Examination (MMSE) and Normalize Whole Brain Volume (nWBV) were inversely related to Clinical Dementia Rating (CDR). Age had a positive correlation and the rest of the features did not appear to have a significant correlation and thus were dropped from the data frame. After visualizing and processing the data, we decided it was appropriate to begin implementing the KNN algorithm.

KNN Algorithm

To implement the KNN algorithm, we needed to use the pandas cut function, as well as binning, to sort CDR values into bins and assign labels. The reason for this was that the CDR values were encoded using floating point values, but the only possible values for this feature were 0.0, 0.5, 1.0, or 2.0; since there were only four possible values for CDR, we knew that this was a KNN classification problem. After putting the CDR values into bins, we utilized the KNeighborsClassifier function from sklearn.neighbors. After computing KNN with a range of N between 2 and 10, we settled on K= 5 for our final choice.

Results + Interpretation

After performing KNN on the processed dataset, there were a few things that we found. First, when we preprocessed the dataset, our entries had been reduced to 235 with only 2 belonging to the ‘2.0’ category. There were so few entries with a CDR of ’2.0’, that it would be impossible to classify a test point from ‘2.0’ into the ‘2.0’ category. We also found an overall accuracy of 72% which seems good, but closer inspection would reveal that our precision kept decreasing with each category. For example, for classifying entries with a CDR of ‘0’ we had a precision of 81%, meaning that for the ones we classified as ‘0’, 81% were correct. However, for classifying entries with a score of ‘1.0’ we had a much lower precision of 33%. This suggests that the model is making more false positive errors for this category, potentially misclassifying instances as category '1.0' when they are actually from other categories. We concluded that the dataset was skewed to be mostly values of ‘0’, so when the model tried to predict values such as ‘0.5’ or ‘1.0,’ it was having a difficult time. This could potentially be fixed by a larger dataset to have a comfortable amount of ‘1.0’ and ‘2.0’ values in the dataset.

```
Accuracy: 0.72
Confusion Matrix:
[[25  5  0]
 [ 5  8  2]
 [ 1  0  1]]
```

```
Classification Report for n = 5:
              precision    recall  f1-score   support

           0       0.81      0.83      0.82        30
         0.5       0.62      0.53      0.57        15
         1.0       0.33      0.50      0.40         2

    accuracy                           0.72        47
   macro avg       0.59      0.62      0.60        47
weighted avg       0.73      0.72      0.72        47
```

Figures 9-10 Accuracy, Confusion Matrix, and Classification Report for K = 5

Acknowledgments

Data were provided [in part] by OASIS 1: Cross-Sectional: Principal Investigators: D. Marcus, R, Buckner, J, Csernansky J. Morris; P50 AG05681, P01 AG03991, P01 AG026276, R01 AG021910, P20 MH071616, U24 RR021382

References

I. OASIS-1: Cross-Sectional: https://doi.org/10.1162/jocn.2007.19.9.1498

II. Rohini, M., Surendran, D. Toward Alzheimer's disease classification through machine learning. Soft Comput 25, 2589–2597 (2021). https://doi.org/10.1007/s00500-020-05292-x