Predicting Alzheimer's Disease

Peush Gomes (1007047802)

2024-11-08

Dataset: Alzheimer's Disease Dataset

Team Name: Peush (G 108)

Final Kaggle Score 0.90705 | Position: 115

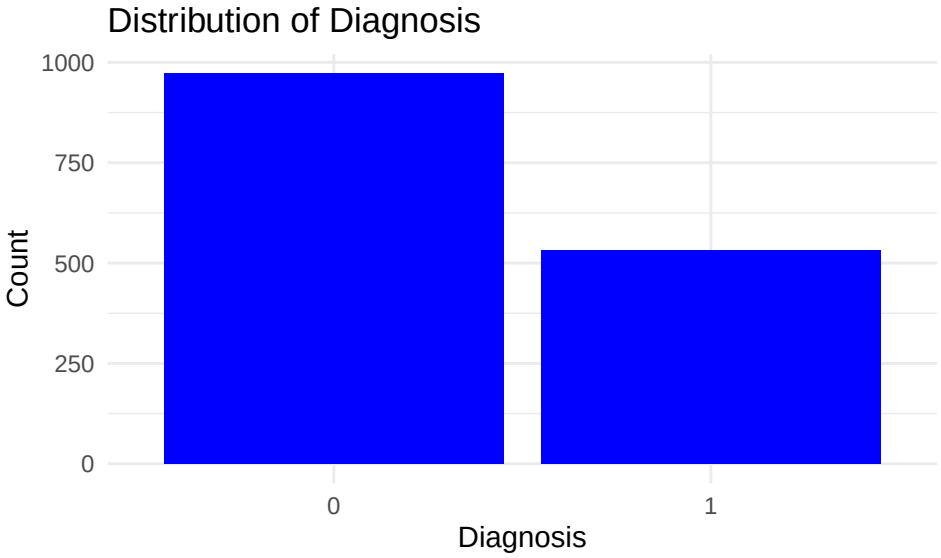
Problem Statement

Medical fields and environments are no stranger to the use of statistical learning. From biostatisical work to prediction models for certain diseases that affect our population. One certain disease that could benefit from prediction models and statistical work is Alzheimer's Disease. Alzheimer's Disease is a neurological disease that progressively deteriorates the person carrying the disease. It affects one on a daily basis and is the leading cause of dementia, thus it is quite important that we are able to detect signs of Alzheimer's as early as possible to improve patients health.

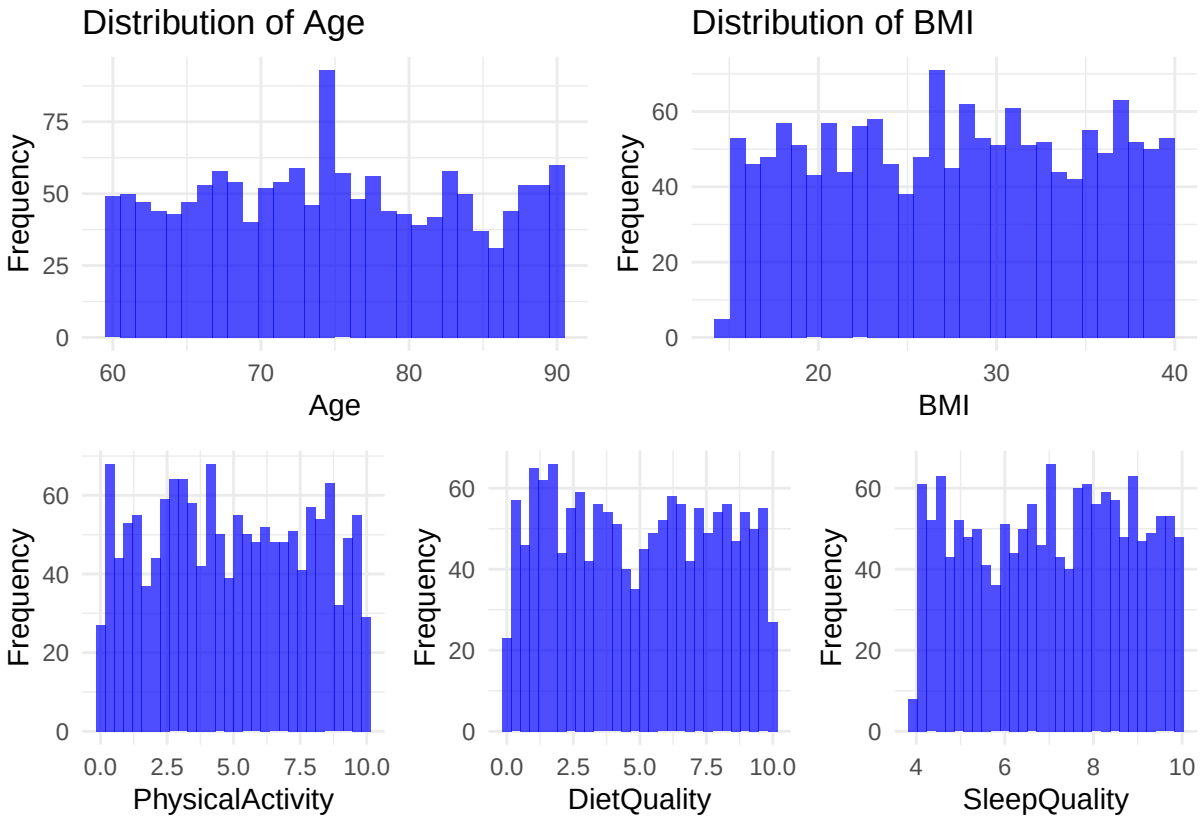
The problem arise that we must develop an accurate predictive model to classify patients into two distinct categories. Those who have Alzheimer's and does who do not. The data that we are working with to create an accurate predictive model consists of 35 different variables that can be split into different categories. There is the details of the patient's demographics, their lifesyle factors, medical history, clinical measurements, cognitive and functional assessments, and symptoms. Identifying the relationships between these variables within the data we have will assist us is diagnosis patients with Alzheimer's, contributing to continuous research efforts to combat the disease. Prior research papers have attempted to create prediction models, long short-term memory was used to make a predictive model with AUC being the determinant value. Another study used cross-validation with bio markers to make a machine learning model.

Exploratory Data Analysis

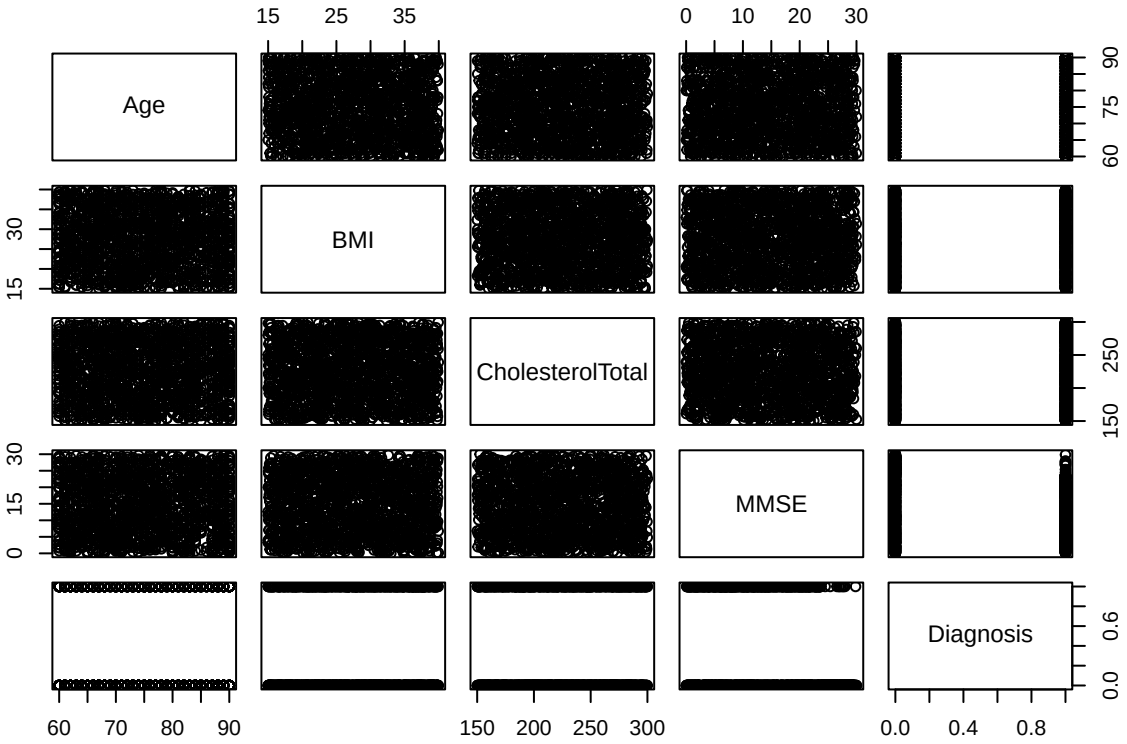
Our dataset consists of a training and test data set, the training dataset has 35 different variables consisting of different categories, with 1504 observations in total.



The distribution of our target value, more of our patients within the training dataset are classified as not having the disease than those who are diagnosised with it. There were observed to be no missing variables, however we did remove `PatientID` and `DoctorInCharge` as those are variable that are not useful in the training of our predictive model

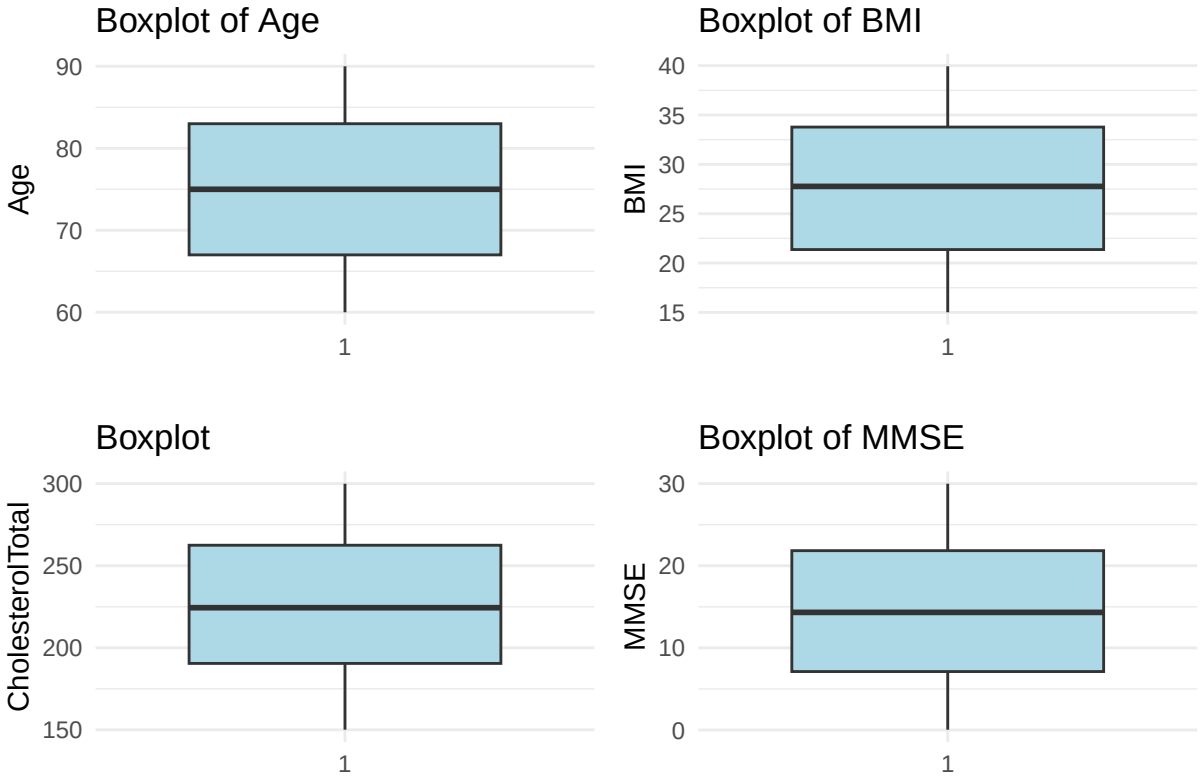


When looking at our distribution of some of our numeric parameters we can observe that they are not heavily skewed in a specific direction, `BMI` for instance is almost uniform in distribution.



It can be observed in the pair plots of our predictors, that there does not exist much of a linear relationships between our predictors. There does seems to exist a bit of relationship between `Age` and `MMSE` as this could

correlate to a decrease in cognitive ability as one gets older.



Based on the boxplots for our numerical predictors, we can also observe that most of the observations are within the first and third quantile. There are not a large amount of outliers in this dataset.

Statistical Analysis

When first creating a predictive model for our binary classfication of Alzheimer's, we started with a full linear regression model. This model would attempt to predict the classfication of an individual with the predictors being every single other variable in the dataset, besides `PatientID` and `DoctorInCharge` which we had excluded in the beginning. To improve our model we began to do stepwise selection in both direction to acheve the best model, AIC being the metric used in this case to assume the model of best fit.

The full model dictated that the meaning predictors (lowest p-values) were that of `MMSE, FunctionalAssessment, MemoryComplaints, BehavioralProblems`, and `ADL`. With an AIC of 1174.1

The step wise selection takes many steps as it is moving back and forth, it settles with a much smaller prediction model only consisting of a few predictors, being Sleep Quality, MMSE, Functional Assessment,Memory Complaints, Behavioral Problems, and ADL (Activities of Daily Living Score). This newer model's AIC is down from 1174.1 to 1134.7

Another model was created using the selected model from the step wise selection, this newer model was to create an interaction term between MMSE and Function Assessment, since both variables deal with a mental state and its given score. The AIC for the model with the interaction term was even better than the one without, leading to the assumption that this model would be of better fit.

This concluded the prediction method through the path of regression models, we also conducted the model creation of decision trees. This classification tree was created with all of the predictors in the dataset in mind. A tree was chosen to be used afterwards to compare as a classification tree is naturally robust in handling both numerical and categorical features and being able to handle nonlinear relationships in the data. This non-linearity was observed in the pair plots. A classification tree also lets us observe which features are the

most important when predicting the disease, providing large insight into which are the most contributing towards the disease.

The classification tree that was created only factored in a few key predictors, FunctionalAssessment, MMSE, ADL, MemoryComplaints, and BehavioralProblems. We are also looking to find the most accurate tree that we can model, one with a low classification error. To see if we could make it any more accurate of a tree, we looked to see if we could prune it futher, pruning the tree led us to get a classification tree with the same predictors but with now 8 terminal nodes instead of 13.

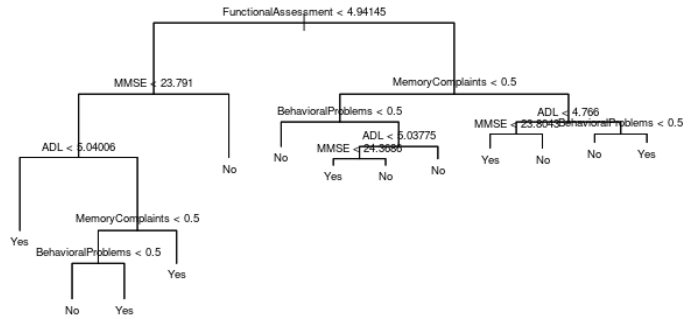


Figure 1: Classification Tree

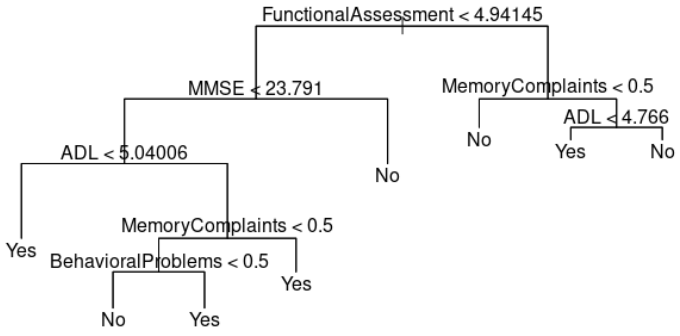


Figure 2: Pruned Tree

A random forest of tree was also created, to see if we could improve on the accuracy of the tree that we had before, as forests have a robustness to overfitting, as our forest consists of 500 different tree all considered each predictor in the dataset.

Results

Table 1: Regression Models

Model	AIC
Full Model	1174.1
Stepwise Selected	1134.7
Interaction Term	1111.9

The full linear regression model with all predictors was expected the least accurate prediction model, its AIC was only 1174. When in comparison the stepwise selected and the interaction term model both had a smaller AIC. However when we used the prediction model and tested it against the testing data, the stepwise selected model was actually the more accurate model, even when it AIC was higher than the interaction term model. The private score for the both of them was the same on the kaggle submission, stepwise performed better on the public score.

However, the trees had performed even better than the regression models. Even though we had pruned the classifcation tree to increase its performance and accuracy, it had actually increased it missclassfication error rate.

Tree	Missclassification Error Rate
Tree (13 nodes)	0.04189
Pruned Tree (8) Nodes	0.07779

Even with the missclassificaiton error rate being higher than the first made tree, the confusion matrix of the pruned tree seemling gave better false positive rates and false negatives. The random forest was similar, in which its confusion matrix heavily resembled the matrix of the pruned tree.

Table 3: Confusion Matrix for Forest

	No	Yes
No	938	34
Yes	40	492

The random forest was seemingly the best fit for a prediction model to classify the disease. When taken to test against the test data of kaggle, the forest had the greatest public score of 0.94894, where as the pruned tree was the worst at 0.89743, with the first tree having a score of 0.94594. However, with all of the test data released, the forest's score had actually gotten worst, to 0.90705. And the first decision tree was now at 0.91025. Meaning that the Decision tree with 13 terminal nodes performed slightly better than that of the random forest. Giving our best model being the decision tree.

Conclusion

In Conclusion we can observe and result that the decision tree and the random forest had generated the greatest prediction model for classifying if a patient has Alzheimer's Disease or not. It out performed both a full regression model as well as it step-wise selection counterpart. Meaning that the robustness of the trees and forest were enough to deduce accurately if a patient has the disease.

Discussion

There was a slight limitation on testing from the testing dataset as it was locked behind a twice a day submission on kaggle, perhaps our models could be even more accurate and designed even better if we had a much larger training dataset. The future for predictive models for this disease are very bright, as decision trees that result in a binary outcome of yes or no do a fantastic job at predicting the outcome. The only downside to them would be if they become to complex they could definitely over fit their training data. To simplistic and then you encounter more miss classifications.

Future work could consider biomarkers, within genetic features of patients being tested. Aswell as implementing even greater and more advanced modeling techniques to further improve accuracy. Being able to test the model other other indepent datasets and ones from different regions could also help generalize our findings.

References

Dataset: https://www.kaggle.com/competitions/classification-of-the-alzheimers-disease/overview

Straiton, J. (2019). Predicting Alzheimer's disease. BioTechniques, 67(4), 146–148. https://doi.org/10.2144/btn-2019-0114

Hong, X., Lin, R., Yang, C., Zeng, N., Cai, C., Gou, J., & Yang, J. (2019). Predicting Alzheimer's Disease Using LSTM. IEEE Access, 7, 80893–80901. https://doi.org/10.1109/ACCESS.2019.2919385

Franzmeier, N., Koutsouleris, N., Benzinger, T., Goate, A., Karch, C. M., Fagan, A. M., McDade, E., Duering, M., Dichgans, M., Levin, J., Gordon, B. A., Lim, Y. Y., Masters, C. L., Rossor, M., Fox, N. C., O'Connor, A., Chhatwal, J., Salloway, S., Danek, A., ... Ewers, M. (2020). Predicting sporadic Alzheimer's disease progression via inherited Alzheimer's disease-informed machine-learning. Alzheimer's & Dementia, 16(3), 501–511. https://doi.org/10.1002/alz.12032

Appendix

```
cleanedtrain <- subset(trainingdata, select = -c(PatientID, DoctorInCharge))
```

```
modfull <- glm(Diagnosis ~ .,family = binomial(link = logit), data = cleanedtrain)
summary(modfull)
```

```
#stepwise
modskinny <- step(modfull, direction = "both")
```

```
#modskinnied
summary(modskinny)
```

```
mod_interaction <- glm(Diagnosis ~ MMSE * FunctionalAssessment + MemoryComplaints +
                         BehavioralProblems + ADL,
                         family = binomial(link = "logit"),
                         data = cleanedtrain)
summary(mod_interaction)
```

```
#predict on test
subpredict <- predict(modskinny, newdata = testdata, type = "response")
subpredict_class <- ifelse(subpredict > 0.5, 1, 0)

testdup <- testdata
testdup$Diagnosis <- subpredict_class

finaltest <- subset(testdup, select = c(PatientID, Diagnosis))
```

```
#interaction predict
int_predict <- predict(mod_interaction, newdata = testdata, type = "response")
int_predict_clas <- ifelse(int_predict > 0.5, 1, 0)

int_testdup <- testdata
int_testdup$Diagnosis <- int_predict_clas

finaltest_int <- subset(int_testdup, select = c(PatientID, Diagnosis))
```

```
#Export_to_csv

write.csv(finaltest,"newpredictions.csv", row.names = FALSE)
write.csv(finaltest_int,"newinteractionpredictions.csv", row.names = FALSE)
```

```
#Treemaking

#tempfactormaking
FactorDiag <- factor(ifelse(cleanedtrain$Diagnosis == 0, "No", "Yes"))

factortrain <- data.frame(cleanedtrain, FactorDiag)

factortrain <- subset(factortrain, select = -c(Diagnosis))

tree.mod <- tree(FactorDiag ~ ., factortrain)


tree.mod_nonfact <- tree(Diagnosis ~ ., cleanedtrain)
```

```r
summary(tree.mod)
summary(tree.mod_nonfact)
```

```r
#treeplot
plot(tree.mod)
text(tree.mod, pretty = 0, cex = 0.6)

plot(tree.mod_nonfact)
text(tree.mod_nonfact, pretty = 0)
```

```r
#this tree is pretty good !!!!!!!
set.seed(123)
tree.predict <- predict(tree.mod_nonfact, newdata = testdata)
tree_predict_clas <- ifelse(tree.predict > 0.5, 1, 0)

tree_testdup <- testdata
tree_testdup$Diagnosis <- tree_predict_clas
```

```r
#TRY PRUNE
cv_tree.mod <- cv.tree(tree.mod, FUN = prune.misclass)

names(cv_tree.mod)

cv_tree.mod
```

```r
plot(cv_tree.mod$size, cv_tree.mod$dev, type = "b")
plot(cv_tree.mod$k, cv_tree.mod$dev, type = "b")
```

```r
#prune it, to the nonfactor
prune_treemod <- prune.misclass(tree.mod, best = 8)
plot(prune_treemod)
text(prune_treemod, pretty = 0)

tree_testdup2 <- testdata

tree_testdup2$Diagnosis <- predict(prune_treemod, newdata = testdata, type = "class")

pruned_out <- tree_testdup2 %>% mutate(Diagnosis = case_when(Diagnosis == "Yes" ~ 1,
                                                             Diagnosis == "No" ~ 0)) %>% select(Patient

write.csv(pruned_out,"pruned8.csv", row.names = FALSE)
```

```r
tre.mod.pred <- predict(tree.mod, trainingdata, type = "class")
pruned_acc <- predict(prune_treemod, trainingdata, type = "class")
table(tre.mod.pred, trainingdata$Diagnosis)
table(pruned_acc, trainingdata$Diagnosis)
```

```r
#randomforest/ this worked even better
set.seed(123)
bag.train <- randomForest(FactorDiag ~ ., data = factortrain, ntree = 500, mtry = 32,  importance = TRU
bag.train

predbag.train <- predict(bag.train, newdata = testdata, type = "class")

randomforestout <- testdata
```

```
randomforestout$Diagnosis <- predbag.train

forestout <- randomforestout %>% mutate(Diagnosis = case_when(Diagnosis == "Yes" ~ 1,
                                                              Diagnosis == "No" ~ 0)) %>% select(Patient
write.csv(forestout,"forestout.csv", row.names = FALSE)
```

```
print(bag.train)
varImpPlot(bag.train)
importance(bag.train)
```

```
finaltree_out <- subset(tree_testdup, select = c(PatientID, Diagnosis))

write.csv(finaltree_out,"newtreepredictions.csv", row.names = FALSE)
```