Predicting Diagnosis of Early-Stage Alzheimer's Disease

Dataset: Alzheimer's Disease Dataset

Luke Ham

8 December 2024

Abstract

Alzheimer's is a disease with no known cure that causes a progressive memory decline—leading to dementia, overall cognitive function decline, and hindrance to one's ability to function in everyday situations (Public Health Agency of Canada, 2017). Consequently, it is important to be able to diagnose Alzheimer's in its early stages. In this project, we developed a predictive classification model using the Random Forest machine learning algorithm to determine whether an individual is diagnosed with Alzheimer's using a dataset containing patient medical information. Following data preprocessing and model tuning, this Random Forest algorithm achieved high accuracy and reliability for predicting whether a patient is diagnosed with Alzheimer's.

Problem Statement

Progressive age-related diseases that affect cognitive health pose a significant burden on hospitals and individuals. Alzheimer's disease, a progressive neurodegenerative disorder, is one of the leading causes of dementia, and diagnosed patients lose their cognitive abilities and memory over time (Government of Canada, 2024). Moreover, as of yet, there is no cure for Alzheimer's. Research indicates that the leading causes of Alzheimer's include several predictors such as age, genetics, health issues, family history, lifestyle factors, and environmental factors. These risk factors differ from person to person and thus determining the cause of dementia is often difficult. Furthermore, the article highlights specific risk factors such as cardiovascular disease, high blood pressure, diabetes, and physical activity (National Institute on Aging, 2023). These predictors are included in the dataset used in this project. Since scientists and doctors are not aware of the exact causes of Alzheimer's, it is vitally important that Alzheimer's early diagnosis rates are improved using the data on these risk factors. Although there is no cure, the development of dementia can be delayed through early treatment which gives patients a higher quality of life (Rasmussen & Langerman, 2019).

The random forest machine learning algorithm was chosen as the primary statistical approach to build a prediction model for the Alzheimer's classification task. Using a dataset containing the health data of 1504 patients, this project aims to use a random forest approach to train a predictive model that can accurately and reliably predict whether an individual is diagnosed with Alzheimer's. The health data used to train this model contains information regarding the patient's demographic, lifestyle, medical history, clinical measurements, cognitive and functional assessments, and symptoms. With this study, we hope to show how machine learning can help with early Alzheimer's disease identification.

The medical problem that this project is motivated to address, is the negative consequences and cognitive harm that result from the typical late diagnosis of Alzheimer's. Since initial symptoms of Alzheimer's are subtle, the disease often goes undetected until severe symptoms are present. Therefore, this project attempts to help detect and predict early diagnosis of Alzheimer's by identifying high-risk patients using the risk factors associated with Alzheimer's and statistical machine learning methods. Overall, the research question is: Can a random forest machine learning model effectively perform binary classification and predict whether or not an individual is diagnosed with Alzheimer's Disease at an early stage based on demographic, behavioral, and clinical predictors?

These problems are both relevant and important to the world, as Alzheimer's disease is the most common form of dementia, affecting approximately 750,000 Canadians and 47.5 million people globally (Government of Canada, 2024; Public Health Agency of Canada, 2017). Despite being the ninth leading cause of death in Canada in 2022, there is no known cure. Thus, this predictive model aims to assist doctors in identifying individuals at high risk of Alzheimer's early, enabling timely treatment to slow disease progression and alleviate symptoms.

Statistical Analyses: EDA - Training Data Analysis and Preprocessing

The original training dataset has 1504 observations of 35 variables with no missing values and obvious outliers. We removed 2 patient labeling predictors called PatientID and DoctorInCharge that are irrelevant to the prediction. Meanwhile, the outcome variable is binary, indicating the diagnosis status for Alzheimer's. 0 indicates a negative diagnosis of Alzheimer's, and 1 indicates a positive diagnosis of Alzheimer's. There are 972 negatives and 532 positives in our training set, suggesting some imbalance between the two groups. As a result, our resulting training dataset has 32 predictors and the Diagnosis response variable.

As previously discussed, several factors influence the likelihood of being diagnosed(National Institute on Aging, 2023); therefore, we conducted a variable analysis to better understand the factors. We have also calculated the correlation matrix and found that there are generally low correlations among numeric variables, indicating no severe multicollinearity. Cholesterol-related variables, with coefficients around -0.04 to -0.05, are moderately correlated due to their shared measurement focus. While Diagnosis is not included in the multicollinearity assessment, some variables show relevance to the outcome. MMSE score, functional assessment score, and activities of daily living score have stronger negative correlations with Diagnosis (-0.24, -0.37, and -0.33, respectively). In contrast, memory complaints and behavioral problems show positive correlations (0.3 and 0.24), suggesting potential predictive value.

MMSE: Arevalo-Rodriguez et al. concluded that MMSE alone is insufficient for identifying patients at risk of Alzheimer's due to inconsistent diagnostic accuracy. MMSE is categorized into four levels of cognitive impairment: normal, mild, moderate, and severe (Guidelines & Protocols Advisory Committee et al., 2014). Most participants fall into the moderate and severe categories. Patients diagnosed with Alzheimer's tend to have lower MMSE scores, with a lower median score compared to non-diagnosed (Figure 1). Figure 2 shows that normal MMSE scores strongly indicate the absence of Alzheimer's, while severe scores are closely linked to the disease.

Physical Activity: By plotting a histogram, the physical activity data shows no obvious skew, with participants covering all activity levels, reflecting diverse lifestyle behaviors. Meanwhile, physical activity levels are nearly identical across the two groups, with similar medians and standard deviations. This suggests that physical activity may not differ meaningfully between individuals with and without Alzheimer's in our dataset, and it likely lacks strong predictive power for distinguishing between these groups in the training data.

Blood Pressure: Figure 3 suggests that both systolic and diastolic blood pressures have identical medians between the two diagnosis groups, indicating a minimal difference in blood pressure distribution between patients with and without Alzheimer's in our dataset, offering some insight for the later model.

Diabetes: Figure 4 shows the number of diabetes cases by diagnosis status. Most participants do not have Alzheimer's, regardless of diabetes status. Additionally, more individuals without Alzheimer's have diabetes compared to those with the disease. However, these findings should be interpreted cautiously due to the dataset's imbalance, with about twice as many participants without Alzheimer's as those with the disease.

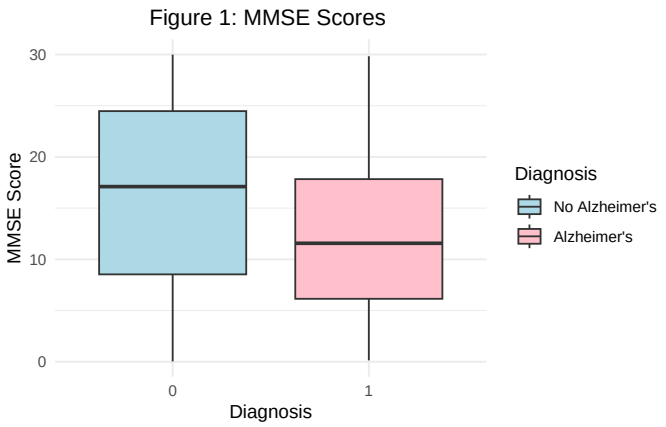


Figure 1: MMSE Scores

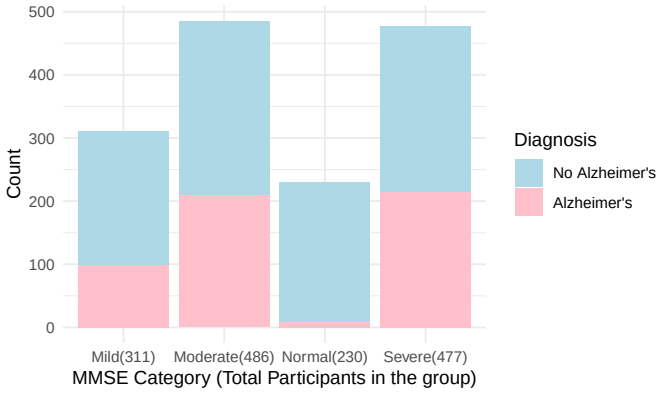


Figure 2: Distribution of MMSE by Diagnosis

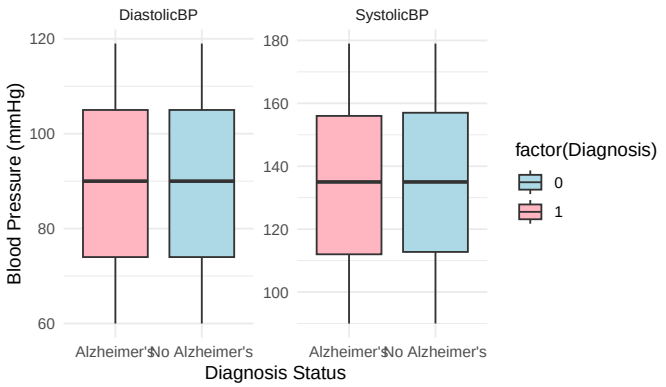


Figure 3: Systolic and Diastolic Blood Pressure by Diagnosis

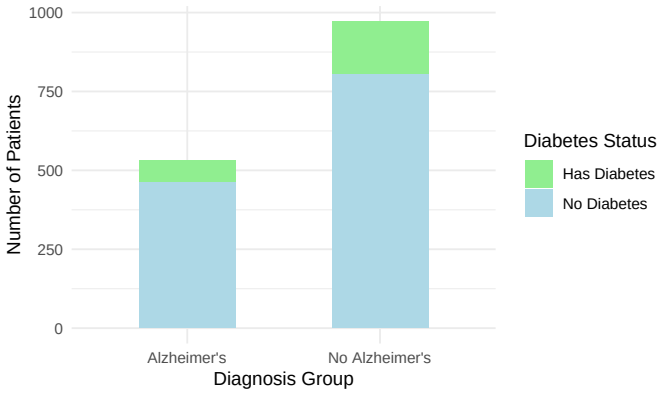


Figure 4: Count of Diabetes by Diagnosis

Statistical Analyses: Random Forest Model

We chose to use a Random Forest machine learning algorithm as it performs well in classification settings. It is also extremely flexible as it can handle both quantitative and qualitative features, and it is often the model of choice in real-life situations where patients' medical information is used to diagnose diseases. A Random Forest model is an advanced machine learning algorithm combining the use of multiple decision trees to reduce variance, prevent over-fitting, and improve prediction accuracy for classification all without increasing bias. The model was primarily chosen because regular decision trees are prone to over-fitting which random forests deal with by creating a large number of decision trees that are decorrelated and trained differently. Then the decision outputs of these decision trees are combined and averaged for the final classification. Similar to bagging, random forest models create decision trees on bootstrap training samples. However, unlike bagging, random forests are not affected by the correlation among decision trees since random forests decorrelate the trees which reduces variance when the trees are averaged. Decorrelation is achieved when building the classification trees of a random forest, each time a split in an individual tree is considered, we randomly select a subset of $mtry$ predictors as split candidates from the full set of predictors. The number of predictors $mtry$ is a hyperparamter that is less than the total number of predictors, otherwise, you would simply be implementing bagging. This subset of $mtry$ predictors is then considered for the split when building each individual tree. In other words, each node is split using the predictor that results in the best split of that subset of $mtry$ predictors. In classification, the best split is one that minimizes Gini impurity. This reduces tree correlation as strong predictors do not dominate every tree as they may not even be considered at each split due to the random subset selection of predictors.

The typical choice of the number of predictors $mtry$, at each split, is the square root of the total number of predictors. Furthermore, after a specified number of decision trees are built on bootstrap training samples with subsets of candidate features at each split, the random forest aggregates all of the trees and their predictions. For classification, a majority vote is used so the class that is voted the most often for by each tree is chosen as the final prediction. The number of decision trees used in a random forest is also a hyperparameter, $ntree$.

Justification and Reasoning for Random Forests Model

We initially considered several other types of models such as linear discriminant analysis, bagging, support vector machine, and logistic regression. However, none of the resulting predictive models were reliable classifiers with good predictive accuracy for diagnosing Alzheimer's. Every model had a prediction accuracy of around 0.88. Ultimately, when compared to these other models, the random forest model easily outperformed them in prediction accuracy. Our choice to use the random forest algorithm is justified by the many advantages that the model offers relative to the dataset used in the project as well as the project's goals.

Since our goal is to diagnose patients with Alzheimer's, we must be able to predict the diagnosis of each patient accurately as both a false positive and false negative diagnosis are very negative. The random forest model is ideal for this project since it is known for its strong predictive performance and flexibility. Next, as seen in EDA, the predictors in this dataset are generally weakly correlated and do not have strong linear relationships, so there is no multicollinearity issue. This is beneficial for random forests, as it handles weakly correlated features well and can detect complex interactions without needing strong linear relationships. Furthermore, a random forest model is ideal as it can capture non-linear relationships without assuming linearity, avoiding bias from oversimplification. It handles both categorical predictors (e.g., symptoms, ethnicity) and continuous predictors (e.g., age) seamlessly, without requiring dummy variables, making it well-suited for the diverse data in this project.

Also, the random forest model is an enhanced extension of both bagging and single decision trees, combining their strengths while addressing their limitations. Decision trees are versatile and interpretable but often suffer from high variance and overfitting. Bagging improves prediction accuracy by aggregating predictions from multiple trees created through random sampling, yet it remains vulnerable to overfitting due to correlations between trees. Random forests overcome this issue by decorrelating the trees, reducing variance, and minimizing the risk of overfitting. This makes random forests robust, effectively handling both weak and strong predictors, and ideal for this study.

In the introduction, it is mentioned that the exact causes of Alzheimer's are unknown. Given this context, we want to use all of the variables that are associated with the disease in our project model. The random forest model is justified in this situation as it allows us to build the model without feature selection since it is not affected by weak and strong predictors. Ultimately, the use of a random forest model is the best choice and further justified given the context of the project and the fact that the model is not prone to over-fitting with good prediction accuracy since we wanted to create a model that can reliably predict and detect Alzheimer's.

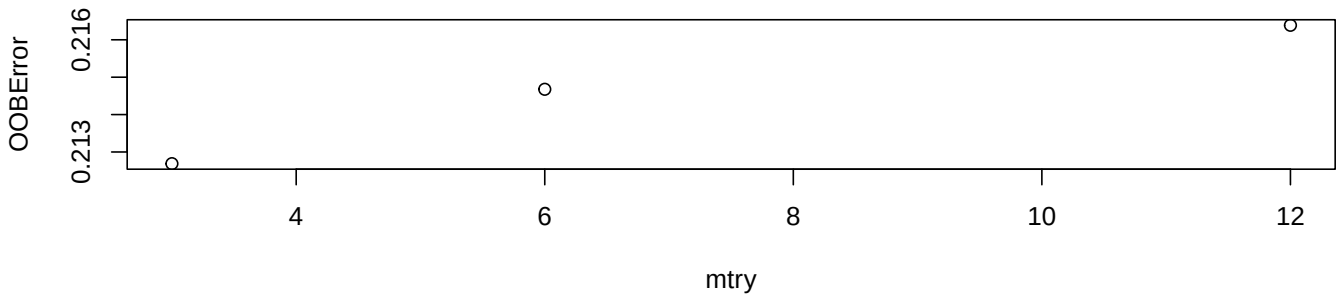
Model Training

The first step in training the model was preprocessing the dataset by checking for missing values and removing irrelevant features. With the exact causes of Alzheimer's unknown, we chose not to do further feature selection since all of the remaining predictors were identified as potential risk factors for Alzheimer's according to our research.

We then wanted to find out which hyperparameters would lead to the highest predictive and most robust random forest model. The two hyperparameters we wished to tune were *mtry*-the number of predictors considered at each split and *ntree*-the number of decision trees used in the model. In order to decide what values we wished to set for our hyperparameters, we decided to use the entire training dataset to train our model as we are short of data with only 1504 patient observations, rather than randomly splitting the training data into a training and validation set with a split such as 70/30. For training a Random Forest model, we do not need to create a separate validation set to calculate the test error. To explain, the validation set approach is not needed for our model since random forests use a bootstrap approach to fit decision trees, where each bootstrap dataset is a subset of the data and does not contain one-third (1/3) of the original observations. Therefore, trees are trained on a subset of around 2/3 of the original data. Since one-third of the original observations are not in each bootstrap dataset, these observations are not used to fit and train each decision tree. These observations are called out-of-bag (OOB) observations that can be used to estimate the test error and evaluate the accuracy of the model. Considering all of this, we will be using out-of-bag (OOB) error with the OOB observations to not only tune the hyperparameters of our model, but also to evaluate the misclassification error rate and accuracy of the model.

We used the *tuneRF* function to find the *mtry* value with the least error rate. Typically *mtry* is the set equal to the square root of the total number of predictors. In our model, this is the square root of 32: $\sqrt{32} = 5.66$. Therefore we began with a *mtry* value of 6 in the *tuneRF* function. As seen in Figure 5, the *tuneRF* function states that the *mtry* value of 3 has the lowest OOB error rate of 0.2126878, however, the three *mtry* values 3, 6, and 12 all have very minimal changes in OOB error. The *mtry* value of 6 has an OOB error rate of 0.2146769 and the *mtry* value of 12 has an OOB error rate of 0.2163881. Instead of relying on strictly *tuneRF*, we decided to check the OOB error rate of each random forest model trained with the three different *mtry* values. After training each model, it was evident that the *mtry* value of 12 resulted in the best prediction accuracy and lowest OOB error rate of 4.19% for diagnosing Alzheimer's. Therefore, to train our random forest model, we chose 12 as our *mtry* hyperparameter—meaning that at each split, we are randomly selecting 12 predictors as candidates.

Figure 5: Out−of−Bag Error Rate for Each Value of the mtry Hyperparameter



The other hyperparameter that needs to be tuned in our random forest model is *ntree*, which defines the number of decision trees that are grown in the random forest model. Generally, more trees result in better performance and better prediction accuracy. As more trees are used in the random forest model, the variance and error rate are expected to decrease whereas the prediction accuracy increases. However, at a certain point, there will be minimal improvements to the error rate, while computational costs continue to rise. Ultimately, we set *ntree* to 1500 since this level of computational cost is manageable, while still attaining high predictive accuracy. We confirmed this experimentally as well by testing our model with different *ntree* values such as 500 and 1000. The prediction accuracy was much more better and reliable when the number of trees increased to 1500.

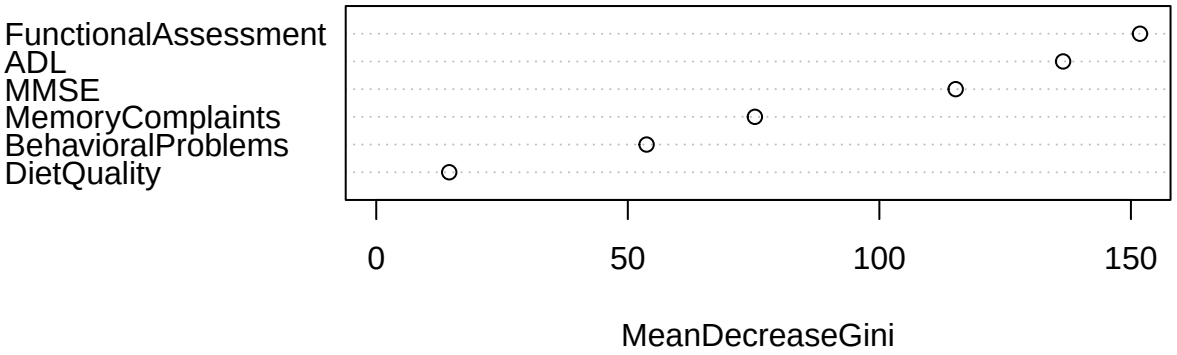
After we tuned the *mtry* and *ntree* hyperparameters, the randomForest function was used to create and train a random forest model using the training dataset and the hyperparameters that we chose. In the end, our random forest model was trained using 1500 decision trees ($ntree = 1500$) and at each split for each decision tree, 12 predictors were randomly sampled as candidates from the entire set of predictors ($mtry = 12$). The resulting trained random forest model is seen in Figure 6.

Figure 6: The Trained Random Forest Model Summary

Attribute	Value
Call	randomForest(formula = Diagnosis ~ ., data = tdata, ntree = 1500, mtry = 12)
Type of Random Forest	Classification
Number of Trees	1500
Number of Variables Tried at Each Split (mtry)	12
OOB Estimate of Error Rate	4.19%

Model Analysis

Figure 7: Top 6 Variable Importance



Variables with a higher mean decrease in Gini index score are more important for accurately predicting the diagnosis of Alzheimer's. The plot in Figure 7 suggests that FunctionalAssessment and Activities of Daily Living score (ADL) are the 2 most significant predictors that impact the diagnosis outcome, indicating that they provide valuable information for distinguishing between positive and negative cases. In contrast, the remaining variables not shown in the graph are less important to this prediction task as their mean decrease in Gini index is closer to 0. These low-importance predictors do not need to be removed since random forests are resistant to over-fitting due to the nature of the model. By choosing the best split from a random subset of predictors at each split of every decision tree, even extremely strong predictors in the model such as FunctionalAssessment are not considered as candidates in a large number of splits in each decision tree. As a result, the trees will not be similar to each other and therefore will not suffer from over-fitting.

Evaluation of Classification Model Performance

To evaluate the performance of our random forest model in performing binary classification of Alzheimer's disease, several metrics were chosen such as accuracy and precision. These metrics are appropriate because it is essential that we correctly diagnose the patient with Alzheimer's since a false diagnosis, whether it is a false positive or a false negative can be detrimental to the patient. A false positive diagnosis could leave the patient and their families worried and stressed while a false negative diagnosis would allow the Alzheimer's disease to go unnoticed and continue to develop, destroying the patient's cognitive abilities and memory over time. We used the OOB estimates to evaluate our model's performance.

To begin, the first metric we used to evaluate the model's accuracy was the overall classification accuracy—the proportion of correctly classified patients. To calculate the classification accuracy, we used a confusion matrix. The confusion matrix is a table that shows and compares the predicted values to actual values—displaying how many patients were misclassified. The confusion matrix can be seen in Figure 8 and by comparing the predicted diagnosis classifications and the real diagnosis of the patients in the dataset, we found that the model produced an overall classification accuracy rate score of 0.9581 or 95.81%. This classification accuracy score was calculated as $1441/1504 = 0.9581$ (the correctly classified observations were $948 + 493 = 1441$ and the total observations were 1504).

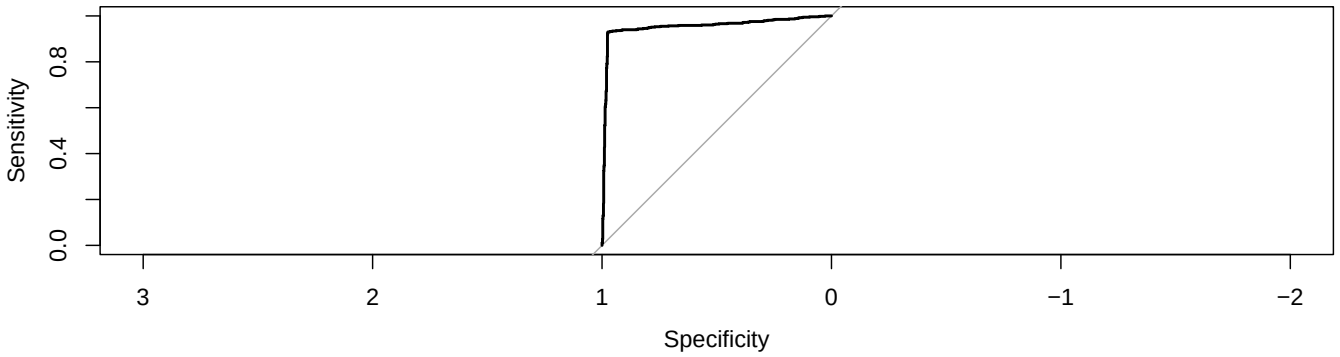
Figure 8: Confusion Matrix of Trained Random Forest Model

	Predicted: No Alzheimer's	Predicted: Alzheimer's	Class Error
Actual: No Alzheimer's	948	24	0.0246914
Actual: Alzheimer's	39	493	0.0733083

Furthermore, according to our trained model in Figure 6, the OOB estimate of the model's error rate is 4.19%, which corresponds to the calculated classification accuracy score from the confusion matrix: 0.9581.

Another metric that was chosen to evaluate our random forest model was analyzing the ROC curve. As shown in Figure 9, this ROC curve's proximity to the top-left corner indicates a high sensitivity and low false positive rate—showing that the model is effective in distinguishing between positive and negative Alzheimer's diagnosis. Furthermore, the high area under the curve (AUC) value of 0.9541 suggests that the model is highly reliable for this prediction task and that the overall performance of this random forest classifier is good.

Figure 9: ROC Curve for Random Forest



Area under the curve: 0.9541

Results

Ultimately, our trained random forest model has an OOB error rate of 4.19% and a classification accuracy of 0.9581.

Some additional metrics of our model can be discerned such as the training error rate, false positive rate, and false negative rate. According to the confusion matrix in Figure 8, the training error rate of our model is (39 + 24) / 1504 = 0.0419 * 100 = 4.19%. The false positive rate (the fraction of patients who do not have Alzheimer's but are classified as having Alzheimer's) can be calculated as (39 / 987) = 0.0395 * 100 = 3.95%. The false negative rate (fraction of patients who do have Alzheimer's but are classified as not having Alzheimer's) is (24 / 517) = 0.0464 * 100 = 4.64%. As stated before, the false positive and false negative rates are important in the context of this project, as identifying high-risk patients of Alzheimer's and diagnosing them at an early stage is very sensitive to errors as the stakes are so high. Encouragingly the false positive and false negative rates for our model are low, which indicates that the model is reliable and accurate in predicting whether a patient has Alzheimer's. Similarly, the confusion matrix provides metrics that can be calculated such as sensitivity which is the true positive rate, and specificity which is the true negative rate. For our model, the sensitivity is 0.9605 because 948/987 = 0.9605 and the specificity is 0.9536 because 493/517 = 0.9536. Since both the sensitivity and specificity are high, it indicates that our random forest model is accurate in diagnosing a patient with Alzheimer's and viable for medical use.

After training our predictive model with the training data, we used the model to predict the diagnosis of Alzheimer's for the 645 patients in the test data set. The resulting predictive accuracy was 0.95195 (rank #33) in the public score Kaggle competition which indicates a good predictive performance for diagnosis. Our model only scored marginally worse in the final Kaggle competition, with a predictive accuracy of 0.91666 (rank #72).

Conclusion

The primary goal of this project was to develop a predictive classification model capable of accurately diagnosing early-stage Alzheimer's disease using patient medical data. Accordingly, after training our predictive random forest model, it was then used to predict the diagnosis of 645 patients in the test data. Through the implementation of a Random Forest machine learning algorithm, we achieved a high-performing predictive model with an accuracy of 0.91666 or 91.666% for diagnosing Alzheimer's on the test dataset in the Kaggle competition. Our final ranking in the competition was rank 72. The model's low

false positive and false negative rates are particularly significant in the medical field, where misdiagnoses can have severe consequences. Furthermore, the high sensitivity and specificity values as well as the high overall classification accuracy indicate that the model is effective at correctly identifying both patients with and without the disease. Overall, our trained model is reliably accurate in detecting Alzheimer's disease at an early stage.

Our findings suggest that machine learning models like the Random Forest classifier can play a crucial role in supporting clinical decisions. By providing a tool for early diagnosis of Alzheimer's, healthcare providers can initiate interventions sooner, potentially slowing disease progression and improving patient outcomes.

In conclusion, the Random Forest model developed in this project demonstrates significant potential for aiding in the early detection of Alzheimer's disease. By leveraging varied medical data, such models can enhance diagnostic processes and contribute to better patient care.

Discussion

First, although we found that a Random Forest model was very well suited to the data as it achieved a very high level of predictive accuracy, this effectiveness could be because there is no precise test for Alzheimer's and the disease is not well understood, so effective proxies for indications of the disease do not exist. It is also likely our dataset is quite noisy. Thankfully Random Forest models are robust to noise because the predictions from each decision tree are averaged out, and is unlikely that noisy data influences every decision tree.

One weakness of our Random Forest approach is that it is difficult to interpret the influence each variable has on the model. While we can say which variables are the most significant using Gini scores, it is difficult to understand the underlying logic of each decision. By using a feature selection method to achieve a more parsimonious model, we would make the model inherently more interpretable. This interpretability may be beneficial to healthcare professionals as the model would indicate which risk factors are most important in the development of Alzheimer's.

Another limitation of our project was the imbalanced output variable in our training dataset. As seen in the EDA, there were around twice more patients without Alzheimer's in the dataset. With less data on the class of patients with Alzheimer's, the imbalanced dataset may have caused our model to be biased towards the no Alzheimer's diagnosis class.

As for potential future directions, to tackle the problem of interpretability, we could first perform a lasso regression to reduce redundant or unhelpful features. We could also conduct a Principal Component Analysis, which may provide insights about the features, possibly revealing unnecessary ones. Additionally, implementing different decision tree models would make sense given the success of the random forest approach. We believe that a boosting model may lead to better prediction results. In boosting, each new tree is built sequentially while focusing on correcting the errors of the previous tree, thus the final model can be well-tailored and capture complex relationships. Boosting has also been shown to achieve a better overall bias-variance tradeoff. By implementing boosting and improving the dataset by collecting more data, our project could be improved.

References

Arevalo-Rodriguez, I., Smailagic, N., Figuls, M. R. I., Ciapponi, A., Sanchez-Perez, E., Giannakou, A., Pedraza, O. L., Cosp, X. B., & Cullum, S. (2015). Mini-Mental State Examination (MMSE) for the detection of Alzheimer's disease and other dementias in people with mild cognitive impairment (MCI). Cochrane Library. https://doi.org/10.1002/14651858.cd010783.pub2

Classification of the Alzheimer's Disease. Kaggle. (n.d.). https://www.kaggle.com/competitions/classification-of-the-alzheimers-disease/data

Government of Canada, S. C. (2024, January 12). Alzheimer's Awareness Month. https://www.statcan.gc.ca/o1/en/plus/5374-alzheimers-awareness-month

Guidelines & Protocols Advisory Committee, Molloy, D. W., M. D., Alemayehu, E., & Roberts, R. (2014). Cognitive Impairment – Recognition, diagnosis and management in Primary care: Standardized Mini-Mental State Examination. https://www2.gov.bc.ca/assets/gov/health/practitioner-pro/bc-guidelines/cogimpsmmse.pdf

National Institute on Aging. (2023, April 5). Alzheimer's Disease Fact Sheet . https://www.nia.nih.gov/health/alzheimers-and-dementia/alzheimers-disease-fact-sheet

Public Health Agency of Canada. (2017, September 21). Dementia in Canada, including Alzheimer's disease. https://www.canada.ca/en/public-health/services/publications/diseases-conditions/dementia-highlights-canadian-chronic-disease-surveillance.html

Randomforest: Classification and Regression with Random Forest. RDocumentation. (n.d.). https://www.rdocumentation.org/packages/randomForest/versions/4.7-1.2/topics/randomForest

Rasmussen, J., & Langerman, H. (2019). Alzheimer's Disease – Why We Need Early Diagnosis. Degenerative Neurological and Neuromuscular Disease, 9, 123–130. https://doi.org/10.2147/DNND.S228939

Appendix

This is the code to reproduce our project results.

```
library(caret)
library(randomForest)
library(datasets)
library(tidyverse)
library(pROC)
library(corrplot)

summary(train_data)
colSums(is.na(train_data))
```

```
# Diagnosis
diagnosis_table <- data.frame(
  Status = c("No Alzheimer's",
             "With Alzheimer's"),
  Number_of_Participants = c(972, 532)
)
kable(diagnosis_table, caption = "Summary Table for Diagnosis")
```

```
# correlation matrix

numeric_vars <- sapply(train_data, is.numeric)
tdata_numeric <- train_data[, numeric_vars]

cor_matrix <- cor(tdata_numeric, use = "complete.obs")

corrplot(cor_matrix, method = "color", type = "upper", tl.col = "black", tl.srt = 45, addCoef.col =
```

```
# MMSE
# Create MMSE category based on the score ranges
MMSE_data <- train_data %>%
  mutate(
    MMSE_Category = case_when(
      MMSE > 25  ~ "Normal(230)",
      MMSE > 19 & MMSE <= 25 ~ "Mild(311)",
      MMSE > 9 & MMSE <= 19 ~ "Moderate(486)",
      MMSE >= 0 & MMSE <= 9  ~ "Severe(477)",
      TRUE ~ NA_character_
    )
  )

MMSE_boxplot <- ggplot(MMSE_data, aes(x = factor(Diagnosis), y = MMSE, fill = factor(Diagnosis)))
  geom_boxplot(outlier.colour = "red") +
  labs(
    title = "MMSE Scores ",
    x = "Diagnosis",
    y = "MMSE Score",
    fill = "Diagnosis"
  ) +
  scale_fill_manual(values = c("lightblue", "pink"), labels = c("No Alzheimer's", "Alzheimer's"))
  theme_minimal()

MMSE_barplot <- ggplot(MMSE_data, aes(x = MMSE_Category, fill = factor(Diagnosis))) +
  geom_bar(position = "stack") +
  labs(
```

```
    title = "Distribution of MMSE by Diagnosis",
    x = "MMSE Category (Total Participants in the group)",
    y = "Count",
    fill = "Diagnosis"
  ) +
  scale_fill_manual(
    values = c("lightblue", "pink"),
    labels = c("No Alzheimer's", "Alzheimer's")
  ) +
  theme_minimal()

MMSE_boxplot <- MMSE_boxplot +
  theme(plot.title = element_text(hjust = 0.5))

MMSE_barplot <- MMSE_barplot +
  theme(plot.title = element_text(hjust = 0.5))

grid.arrange(MMSE_boxplot, MMSE_barplot, ncol = 2,widths = c(35, 65))
plot2 <- MMSE_boxplot
plot6<- MMSE_barplot
```

```
blood_train_data <- train_data %>%
  mutate(Diagnosis_Group = case_when(
      Diagnosis == 0 ~ "No Alzheimer's",
      Diagnosis == 1 ~ "Alzheimer's"
    ))

# Summary table for SystolicBP and DiastolicBP by Diagnosis
blood_summary_table <- blood_train_data %>%
  group_by(Diagnosis_Group) %>%
  summarize(
    SystolicBP_Mean = mean(SystolicBP, na.rm = TRUE),
    SystolicBP_Median = median(SystolicBP, na.rm = TRUE),
    DiastolicBP_Mean = mean(DiastolicBP, na.rm = TRUE),
    DiastolicBP_Median = median(DiastolicBP, na.rm = TRUE))
kable(blood_summary_table, caption = "Summary Table for Blood Pressure")

long_data <- blood_train_data %>%
  pivot_longer(cols = c(SystolicBP, DiastolicBP), names_to = "variable", values_to = "value")
ggplot(long_data, aes(x = factor(Diagnosis_Group), y = value, fill = factor(Diagnosis))) +
  geom_boxplot(outlier.colour = "red") +
  facet_wrap(~variable, scales = "free") +
  labs(
    title = "Comparison of Systolic and Diastolic Blood Pressure by Diagnosis",
    x = "Diagnosis Status",
    y = "Blood Pressure (mmHg)"
  ) +
  scale_fill_manual(values = c("lightblue", "lightpink")) +
  theme_minimal()

plot4 <- ggplot(long_data, aes(x = factor(Diagnosis_Group), y = value, fill = factor(Diagnosis)))
  geom_boxplot(outlier.colour = "red") +
  facet_wrap(~variable, scales = "free") +
  labs(
    title = "Comparison of Systolic and Diastolic Blood Pressure by Diagnosis",
    x = "Diagnosis Status",
```

```
    y = "Blood Pressure (mmHg)"
  ) +
  scale_fill_manual(values = c("lightblue", "lightpink")) +
  theme_minimal()
```

```
Diab_train_data <- train_data %>%
  mutate(
    Diabetes_count = case_when(
      Diabetes == 0 ~ "No Diabetes",
      Diabetes == 1 ~ "Has Diabetes"
    ),
    Diagnosis_Group = case_when(
      Diagnosis == 0 ~ "No Alzheimer's",
      Diagnosis == 1 ~ "Alzheimer's"
    )
  )

#table(Diab_train_data$Diabetes_count)

Dia_summary_table <- Diab_train_data %>%
  group_by(Diagnosis_Group, Diabetes_count) %>%
  summarize(Count = n(), .groups = "drop")
#print(Dia_summary_table)

Dia_summary_table <- data.frame(
  Patient_Status = c("Alzheimer's and Has Diabetes",
                     "Alzheimer's and No Diabetes",
                     "No Alzheimer's and Has Diabetes",
                     "No Alzheimer's and No Diabetes"),
  Number_of_Patients = c(71, 461, 169, 803)
)
kable(Dia_summary_table, caption = "Number of Diabetes Cases by Diagnosis Status")


ggplot(Diab_train_data, aes(x = Diagnosis_Group, fill = Diabetes_count)) +
  geom_bar(position = "stack", width = 0.5) +
  labs(
    title = "Count of Diabetes by Diagnosis",
    x = "Diagnosis Group",
    y = "Number of Patients",
    fill = "Diabetes Status"
  ) +
  scale_fill_manual(
    values = c("lightgreen", "lightblue")
  ) +
  theme_minimal()

plot5 <-ggplot(Diab_train_data, aes(x = Diagnosis_Group, fill = Diabetes_count)) +
  geom_bar(position = "stack", width = 0.5) +
  labs(
    title = "Count of Diabetes by Diagnosis",
    x = "Diagnosis Group",
    y = "Number of Patients",
    fill = "Diabetes Status"
  ) +
  scale_fill_manual(
```

```
    values = c("lightgreen", "lightblue")
  ) +
  theme_minimal()
```

```
grid.arrange(
  plot2, plot6,
  plot3, plot5,
  plot4,  # First row  # Second row
  ncol = 2,            # Number of columns
  top = "Combined Graphs"  # Optional: Add a title
)
```

```
data <- read.csv("train.csv")
data2 <- read.csv("test.csv")

# Remove unneeded variables from training data
tdata <- select(data, -DoctorInCharge, -PatientID)

# Remove unneeded variables from test data
test_data <- select(data2, -DoctorInCharge, -PatientID)

# Create Diagnosis outcome variable in test data set
test_data$Diagnosis = ""

# Make outcome Diagnosis a factor in original training data
tdata$Diagnosis <- as.factor(tdata$Diagnosis)
test_data$Diagnosis <- as.factor(test_data$Diagnosis)

str(tdata)
str(test_data)
```

```
# Choosing mtry number for random forest model
set.seed(1)
#tuning mtry (number of variables to randomly sample as candidates at each split)
# Choosing best mtry value for random forest model
# Mtry is the number of variables that are randomly sampled as candidates from the set of predict
# Starting mtry value is 6 as that is the square root of the total number of predictors (32)
tune_mtry <- tuneRF(tdata[,-32], tdata[,32], mtryStart = 6, stepFactor = 0.5, plot = TRUE,
                    ntreeTry = 1500, trace = TRUE, improve = 0.01)
```

```
# Create random forest model
set.seed(5)
# Create Random Forest Model for Alzheimers Data
# Choose number of trees (ntree) as 1500 since higher number of trees results in more accurate pr
# Choose mtry as 12 as from testing, 12 gives the best prediction score. 12 is double the square
rf_model <- randomForest(Diagnosis~., data=tdata, ntree=1500, mtry=12)

# Display Random Forest Model
print(rf_model)
```

```
# Variable Importance
#Feature Importance Analysis
importance(rf_model)

#Variable importance plot
varImpPlot(rf_model)
```

```r
#Top 10 Variable Importance
varImpPlot(rf_model, sort = T, n.var = 10, main = "Variable Importance")
```

```r
# ROC Curve Analysis
roc<-roc(tdata$Diagnosis,rf_model$votes[,2])
plot(roc, main = "ROC Curve for Random Forest")
auc(roc)
```

```r
# Make Predictions Using Model
# Create Prediction
prediction1 <- predict(rf_model, test_data)
```

```r
# For prediction submission
# Create data frame of patient ID and Predicted Diagnosis for submission
submit_prediction <- data.frame(PatientID=data2$PatientID,
                                Diagnosis=prediction1)

# Export csv for submission
write.csv(submit_prediction, "forest_prediction.csv", row.names = FALSE)
```