Final Report

Yuhan Zhang

2025-03-15

1 Abstract

This report explores the application of Random Forest classification model on a dataset that contains 360 numerical features to predict if the patient has Alzheimer's disease The dataset has 339 observations, with label AD if the patient has Alzheimer's disease or C if the patient does not have it. After data preprocessing and feature selection, a Random Forest model was trained and evaluated on the data.

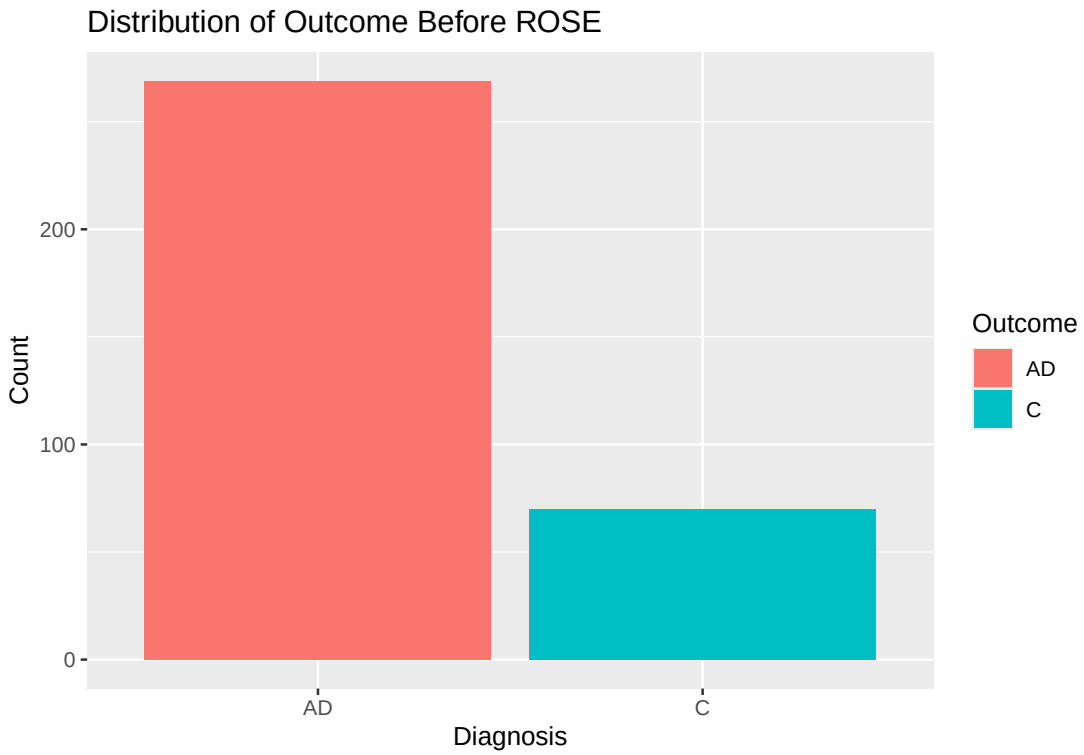
2 Introduction

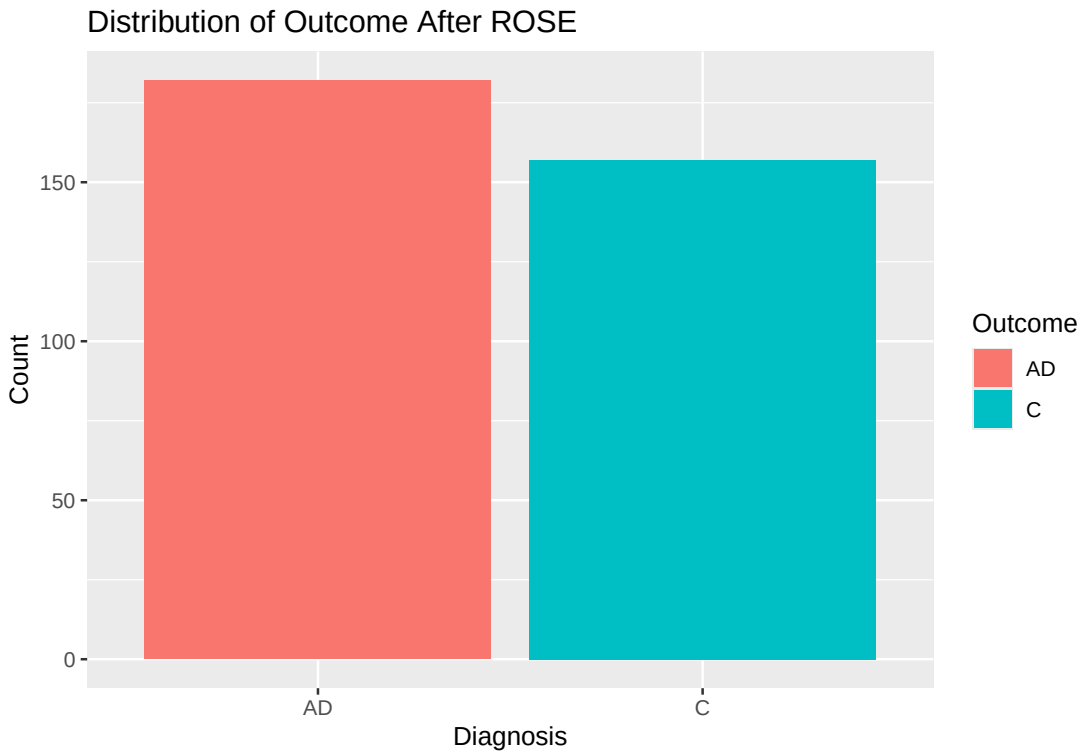
Alzhermer's disease is a progressive neurodegenerative disorder that affects cognitive function and memory. Since 1907 Alois Alzheimer found the case of the first case of Alzheimer's disease, it has been a century while this is still a concern for human society. The dataset that is used in this report contains variables which describe the brain activity and whether the patient has Alzheimer's disease or not.

3 Data

The dataset consists of 339 samples and 360 numerical features extracted from MRI scans, describing brain activity and structural measurements. The Outcome column indicates whether a patient has Alzheimer's disease (AD) or not (C).

However, the dataset is highly imbalanced: there are 269 observations in the AD class and only 70 observations in the C class. This imbalance can lead to a biased model, where the model prefers the majority class (AD) and classify all the observations to the minority class (C) to have higher accuracy.





To address this issue, the Random Over-Sampling Examples (ROSE) method was applied. ROSE generates synthetic samples for the minority class based on kernel density estimation rather than simply duplicating or removing existing samples. This helps create a more balanced data while still preserving the statistical distribution of the original data.[2] After using this technique, the total number of observations is still 339, with the number of observation in class AD is 182, and the number of observation in class C is 157.

4 Model Selection and Training

Given the high number of features, Random Forest is selected as the primary classification model due to its ability to handle high-dimensional data, resistance to overfitting, and feature importance evaluation.

Random Forest is a group of decision trees, where each tree is trained on a random subset of the training data. The random forest usually started with Bootstrapped Sampling, where each decision tree is trained on a random subset of the data; feature randomization, since only a random subset of the features is considered for each the tree developing; majority voting, in which the classification is determined by the majority vote across all trees.

The dataset is split into training and testing set. The training set contains 80% of the data, and testing set contains the rest of the data. The Random Forest model is fitted in R, with parameter `ntree` equals 500, which is the number of trees, and `mtry` equals 30, which is the number of parameters in each split.

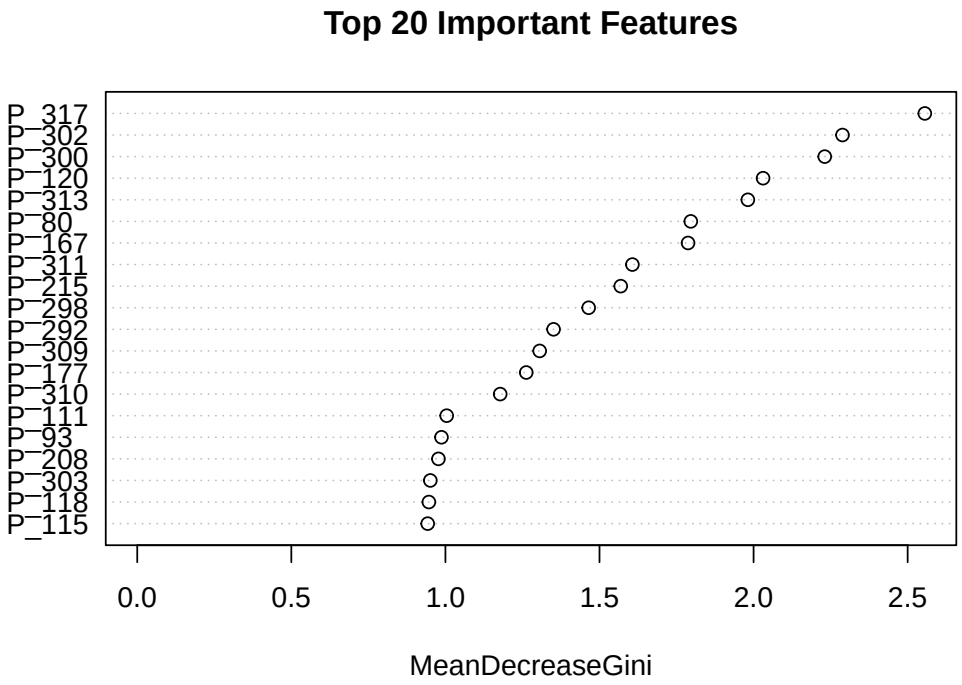
5 Results

5.1 Important Features

The summary of the selected Random Forest model, and the confusion matrix on training set is:

```
##
## Call:
##  randomForest(formula = Outcome ~ ., data = train_data, ntree = 500,      mtry = 30)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 30
##
##         OOB estimate of  error rate: 15.5%
## Confusion matrix:
##     AD   C class.error
## AD 129  16   0.1103448
## C   26 100   0.2063492
```

The important features from Random Forest model are below:



The confusion matrix on the testing set is:

```
##     actual
## pred AD  C
##   AD 33  2
##   C   4 29
```

The accuracy on the training set is 0.845, and on the testing set is 0.912. This shows that the Random Forest model has a good performance on both the training set and testing set.

6 Conclusion

This study applied the Random Forest algorithm to classify the AD_360 dataset, achieving high accuracy and interpretability. The feature importance analysis identified key predictive variables, paving the way for future refinements.

7 Appendix

7.1 Code

```
set.seed(1)
data <- read.csv("AD_360_training.csv")
data$Outcome <- as.factor(data$Outcome)
# apply ROSE to make the two cases balanced
new_data <- ROSE(Outcome ~., data)$data

train_idx <- sample(1:nrow(new_data), 0.8 * nrow(new_data))
train_data <- new_data[train_idx, ]
test_data <- new_data[-train_idx, ]

rf_model <- randomForest(Outcome ~ ., data = train_data, ntree = 100,
                         mtry = 30)

predictions <- predict(rf_model, test_data)

accuracy <- sum(predictions == test_data$Outcome) / nrow(test_data)
```

8 Reference

[1] Fischl, B., & Dale, A. M. (2000). Measuring the thickness of the human cerebral cortex from magnetic resonance images. *Proceedings of the National Academy of Sciences, 97*(20), 11050–11055. https://doi.org/10.1073/pnas.200033797

[2] Lunardon, N., Menardi, G., & Torelli, N. (2014). ROSE: a Package for Binary Imbalanced Learning. *The R Journal, 6*(1), 79. https://doi.org/10.32614/rj-2014-008