```
# Import libraries
# install.packages("fastDummies")
# install.packages("car")
# install.packages("GGally")
# install.packages("corrplot")
library(car)
library(fastDummies)
library(dplyr)
library(ggplot2)
library(corrplot)
library(GGally)
library(tree)
library(class)

# Import data
# getwd()
# setwd("C:/")
alz_data <- read.csv("Alzheimer.csv")
summary(alz_data)
str(alz_data)


# Data processing
# Remove Nulls
alz_data <- na.omit(alz_data)
colSums(is.na(alz_data))

# Convert M.F dummies M = 1 F = 0
alz_data$M.F <- ifelse(alz_data$M.F == "M", 1, 0)
names(alz_data)[names(alz_data) == 'M.F'] <- 'Male'

# Convert Group dummies Demented = 1 UnDemented = 0, drop Converted
alz_data <- alz_data[alz_data$Group != "Converted", ]
alz_data$Group <- ifelse(alz_data$Group == "Demented", 1, 0)

# Convert SES and CDR into categorical variables
alz_data <- dummy_cols(alz_data, select_columns = 'SES',
                       remove_first_dummy = TRUE)
alz_data <- dummy_cols(alz_data, select_columns = 'CDR',
                       remove_first_dummy = TRUE)

# Remove old SES and CDR
alz_data <- alz_data[, !(names(alz_data) %in% c("SES", "CDR"))]
# summary(alz_data)
# str(alz_data)

# Export the cleaned data
# write.csv(alz_data, "Alzheimer_Final.csv", row.names=FALSE)

# Looking at the correlations b/t variables
cor(alz_data)
corrplot(cor(alz_data), method = "circle", type = "upper",
         order = "hclust", tl.col = "black", tl.srt = 45, tl.cex = 0.7)

# Looks like need to address multicollinearity issue
# Let's perform the Variance Inflation Factor (VIF) analysis
model <- lm(Group ~ ., data = alz_data)
vif_values <- vif(model)
print(vif_values)

# Look at the values of eTIV and ASF! They are highly collinear!
# Removing ASF as it is just normalized brain volume (head size), but eTIV is
# the estimated volume of the brain, meningitis (membrane layers), and
# cerebrospinal fluid from an MRI.
alz_data <- alz_data[, -which(names(alz_data) == "ASF")]
str(alz_data)

# Now making sure necessary cols are factors
cols_to_convert <- c("Group", "Male", "SES_2", "SES_3", "SES_4", "SES_5",
                     "CDR_0.5", "CDR_1", "CDR_2")
alz_data[cols_to_convert] <- lapply(alz_data[cols_to_convert], as.factor)


# Now let's take a look at the processed data (Descriptive & Visualizations)
str(alz_data)
summary(alz_data) # much more 0s then 1s, females than males

# Another nice way package and function:
# install.packages("psych")
# library(psych)
# describe(alz_data)

# Bar plot for a categorical column 'Group'
ggplot(alz_data, aes(x = as.factor(Group))) +
  geom_bar() +
  xlab("Group") +
  geom_text(stat = 'count', aes(label = ..count..), vjust = -0.5)

# Bar plot for a categorical column 'Male'
ggplot(alz_data, aes(x = Male)) +
  geom_bar() +
  xlab("Male") +
  geom_text(stat = 'count', aes(label = ..count..), vjust = -0.5)

# Histogram for a numeric column named 'Age'
ggplot(alz_data, aes(x = Age)) +
  geom_histogram(binwidth = 3, fill = "blue", color = "black")

# Histogram for a numeric column named 'EDUC'
ggplot(alz_data, aes(x = EDUC)) +
  geom_histogram(binwidth = 3, fill = "blue", color = "black")

# Boxplot for numeric column 'Age' grouped by a categorical column 'Group'
ggplot(alz_data, aes(x = as.factor(Group), y = Age)) +
  geom_boxplot() +
  xlab("Group")


# Logistic Regression Analysis
# All-in Logistic Regression
reg_all <- glm(Group ~ ., data = alz_data, family = "binomial")
summary(reg_all)

# Look at the Residual deviance of 0! The all-in Logistic Regression indicates
# a perfect fit! Not good not good not good!

# When you think about the Clinical Dementia Rating (CDR variable), this literally
# is the assessment of if the individual has Alzheimer or not, so a CDR of 0
# means exactly undemented, and having higher levels like 1 or 2 is almost certain that
# individual are demented. This really undermines other variables' correlation with
# our response variable. (It might be the reason why our all-in regression has
# a perfect fit!)

# Therefore let's run a Logistic Regression without CDRs
# Now we get meaningful coefs. from our regression.
reg_2 <- glm(Group ~ Male + Age + EDUC + MMSE + eTIV + nWBV
             + SES_2 + SES_3 + SES_4 + SES_5, data = alz_data,
            family = "binomial")
summary(reg_2)
anova(reg_2, reg_all)
# Again the resid. dev. of 0 shows the perfect fit of reg_all.


# Here the age variable does not make sense since the greatest known risk factor
# for Alzheimer's and other dementias is increasing age - After age 65, the risk
# of Alzheimer's doubles every five years, according to the Alzheimer's Association.

# To understand what is going on:
table(alz_data$Group)
alz_data %>% filter(Group == "0" & Age > 70) %>% summarise(Count = n())

# Visualize the relationship between Age and the likelihood of getting 0s and 1s
ggplot(alz_data, aes(x = Age, fill = as.factor(Group))) +
  geom_bar() +
  labs(x = "Age", y = "Count", fill = "Group") +
  theme_minimal()

# We can clearly see that there are more 0s in higher age groups than 1s which
# gives more weight and the false interpretation that aging might decrease the
# risk of getting this disease.

# In fact, Women have a greater risk of developing dementia during their lifetime.
# Around twice as many women have Alzheimer's disease – the most common type of
# dementia – compared to men, according to Alzheimer's Society in the U.K.
# The main reason for this greater risk is because women live longer than men
# and old age is the biggest risk factor for this disease.
# So the coefficient of the Male predictor is also false against the real life.

# Create a bar plot that shows the count of 0s and 1s for each gender in the dataset.
ggplot(alz_data, aes(x = as.factor(Male), fill = as.factor(Group))) +
  geom_bar(position = "dodge", stat = "count") +
  labs(x = "Gender", y = "Count", fill = "Group") +
  scale_x_discrete(labels = c("Female", "Male")) +
  theme_minimal()

# Same problem. Much more 0s than 1s in female group and more 1s than 0s in male
# group which leads to the false conclusion.


# Now let's think about the interpretation of other variables
summary(reg_2)
summary(alz_data)
# MMSE (Integer, Mini Mental State Examination): This variable is a cognitive
# screening tool. Lower MMSE scores may indicate cognitive impairment and are
# associated with an increased risk of Alzheimer's.

# eTIV (Integer, Estimated Total Intracranial Volume): The estimated volume of
# the brain, meningitis (membrane layers), and Cerebrospinal fluid from an MRI.

# nWBV (Float, Normalized Whole Brain Volume): Volume within the cranium (white
# and gray matter - tissues and fibers), potentially useful in studying structural
# changes related to cognitive decline.


# Testing the accuracy of the model
# With a dataset of only 317 observations, splitting the data into training and
# testing sets, especially in a 50-50 ratio, might limit the amount of data
# available for training your model, potentially affecting its performance
# and generalizability.

# So let's do the 10-Fold Cross-validation which is a powerful technique for small
# datasets.
library(caret)

# Define the control parameters for cross-validation
ctrl <- trainControl(method = "cv", number = 10)

# Fit logistic regression model using cross-validation
fit <- train(Group ~  Male + Age + EDUC + MMSE + eTIV + nWBV
             + SES_2 + SES_3 + SES_4 + SES_5, data = alz_data, method = "glm",
             family = "binomial", trControl = ctrl)

# Predicted classes from cross-validated logistic regression model
predicted_classes <- predict(fit, type = "raw")

# Create a confusion matrix
confusion_matrix <- confusionMatrix(predicted_classes, alz_data$Group)
confusion_matrix
# (177 + 99) / 317

# Extract accuracy, precision, recall, and F1 score from the confusion matrix
accuracy <- confusion_matrix$overall['Accuracy']
accuracy # 0.8706625 looks fantastic

precision <- confusion_matrix$byClass['Pos Pred Value']
precision

recall <- confusion_matrix$byClass['Sensitivity']
recall

f1_score <- confusion_matrix$byClass['F1']
f1_score # 0.8962025 is wonderful

# T.B.Cont...?
# Do we wanna do Bootstrapping to generate more data to build our model...?
# Probably not because the dataset collection itself is not a good
# representation of our real-world...?


# Decision Trees
# We also employ a tree model to assess the relative performance between the
# tree model and logistic regression,
# and herein lies our findings.
tree1 <- tree(Group~ Male + Age + EDUC + MMSE + eTIV + nWBV +
                SES_2 + SES_3 + SES_4 + SES_5, data = alz_data)
summary(tree1)
plot(tree1)
text(tree1, pretty = 1, cex = 0.5)

# Based on the tree plot, we observe that MMSE is the primary factor in the
# initial split, followed by nWBV and eTIV, which aligns with the conclusion
# drawn from the logistic regression analysis. However, the tree appears to be
# overfitting due to excessive branching, leading to redundancy in some of the
# leaf nodes.
# For instance, both eTIV < 1652.5 branches result in the same outcome (0).
# To address this issue, we need to employ pruning techniques to refine the tree
# and improve its generalization capability.
#  ------
prune1 <- prune.misclass(tree1)
names(prune1)
#
# Plot the results of the prune
#
plot(prune1)
plot(prune1$size, prune1$dev, xlab = "Size of Tree",
     ylab = "Deviation")
```

```
# As we can see the relation between size and misclass, When the size is 5, the
# misclassification rate is 40%.
# However, as the size approaches 16, the misclassification rate decreases to
# approximately 22%.
# Therefore, we select 16 as the optimal size since the misclassification rate
# no longer decreases.
prune.tree <- prune.misclass(tree1, best = 16)
summary(prune.tree)

# Residual mean deviance:  0.3653 = 109.9 / 301
# Misclassification error rate: 0.0694 = 22 / 317

prune.tree
plot(prune.tree)
text(prune.tree, pretty = 1, cex = 0.7)
# Boom!! The prune tree works really good!
# Having no repeated branches and clear separation in each branch indicates that
# the model is effectively partitioning the data based on the selected variables,
# which is generally desirable in a classification tree model.


# Use cross-validation to train a decision tree model
fit_tree <- train(Group ~  Male + Age + EDUC + MMSE + eTIV + nWBV
             + SES_2 + SES_3 + SES_4 + SES_5, data = alz_data, method = "rpart",
            trControl = ctrl)

# Predicted classes from cross-validated logistic regression model
predicted_classes_tree <- predict(fit_tree, type = "raw")

# Create a confusion matrix
confusion_matrix_tree <- table(predicted_classes_tree, alz_data$Group)

# Create a confusion matrix
confusion_matrix_tree
# (180 + 91) / 317

# Extract accuracy, precision, recall, and F1 score from the confusion matrix
accuracy_tree <- confusion_matrix_tree$overall['Accuracy']
accuracy_tree # 0.8549 <- A bit lower than logistic regression

precision_tree <- confusion_matrix_tree$byClass['Pos Pred Value']
precision_tree  # 0.7165

recall_tree <- confusion_matrix_tree$byClass['Sensitivity']
recall_tree  # 0.9010

f1_score_tree <- confusion_matrix_tree$byClass['F1']
f1_score_tree # 0.7997 <- Lower than logistic regression

# Based on this result, it indicates that Logistic regression is better
# than Prune tree on this experiment.

# write.csv(alz_data, "Alzheimer_Final.csv", row.names=FALSE)
```