Business Data Analytics

Decision Trees Project

By
Group-1

Sreenija Dharma

Bal Thirupathi Guddati

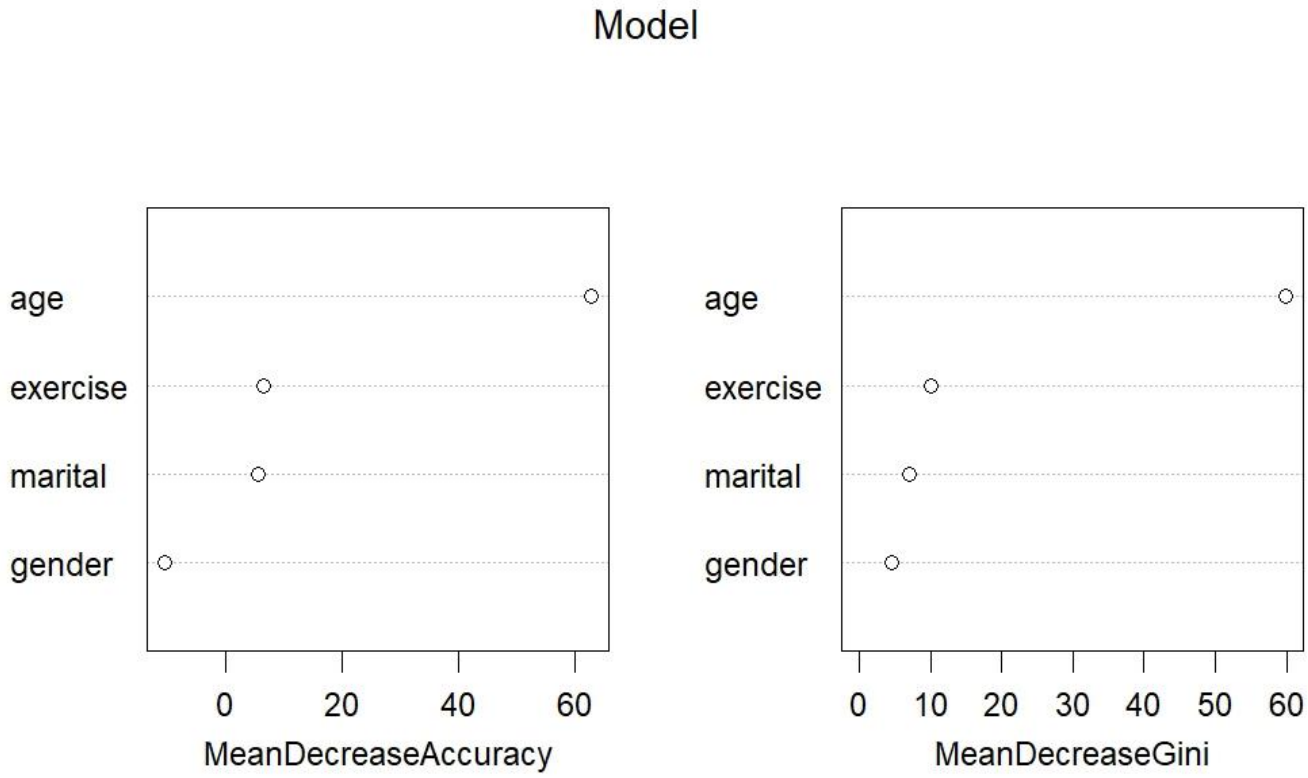
Sandeep Chanda

Professor: Santiago Aguirre

BIS581

Class:22450889

a. Analysis of the predictors with higher importance for predicting oatmeal preference



Predictors	Mean Decrease Gini
Age	**59.89**
Exercise	**10.02**
Marital	**6.97**
Gender	**4.46**

Predictors	Mean Decrease Accuracy
Age	**62.75**
Exercise	**6.60**
Marital	**5.49**
Gender	**-10.41**

Analysis:

From the analysis, it is observed that age is the most significant factor for the prediction of breakfast preference, with the highest mean decrease accuracy and mean decrease Gini, whose values are 62.75 and 59.89, respectively. This indicates there is a strong relationship between age and breakfast preference. Exercise has a mean decrease accuracy of 6.60 and a mean decrease Gini of 10.02, which indicates that predictions of breakfast preferences are slightly influenced by exercise habits.

In comparison to age and exercise, both the mean decrease accuracy and the mean decrease Gini of marital status are significantly lower, at 5.49 and 6.97, respectively. The low mean decrease Gini (4.46) and mean decrease accuracy (-10.41) indicate that gender alone not consider as a significant predictor in predicting breakfast preferences

b. Analysis of prediction accuracy (confusion matrix).

Confusion matrix for the training data set:

Pred train	Cereal Bar	Cereals	Oatmeal	% Accuracy
Cereal Bar	63	37	4	60.57
Cereals	71	117	42	50.86
Oatmeal	28	85	168	59.78

Confusion matrix for the validation data set:

Pred valid	Cereal Bar	Cereals	Oatmeal	%Accuracy
Cereal Bar	28	16	3	59.57
Cereals	31	59	16	55.66
Oatmeal	10	25	76	68.46

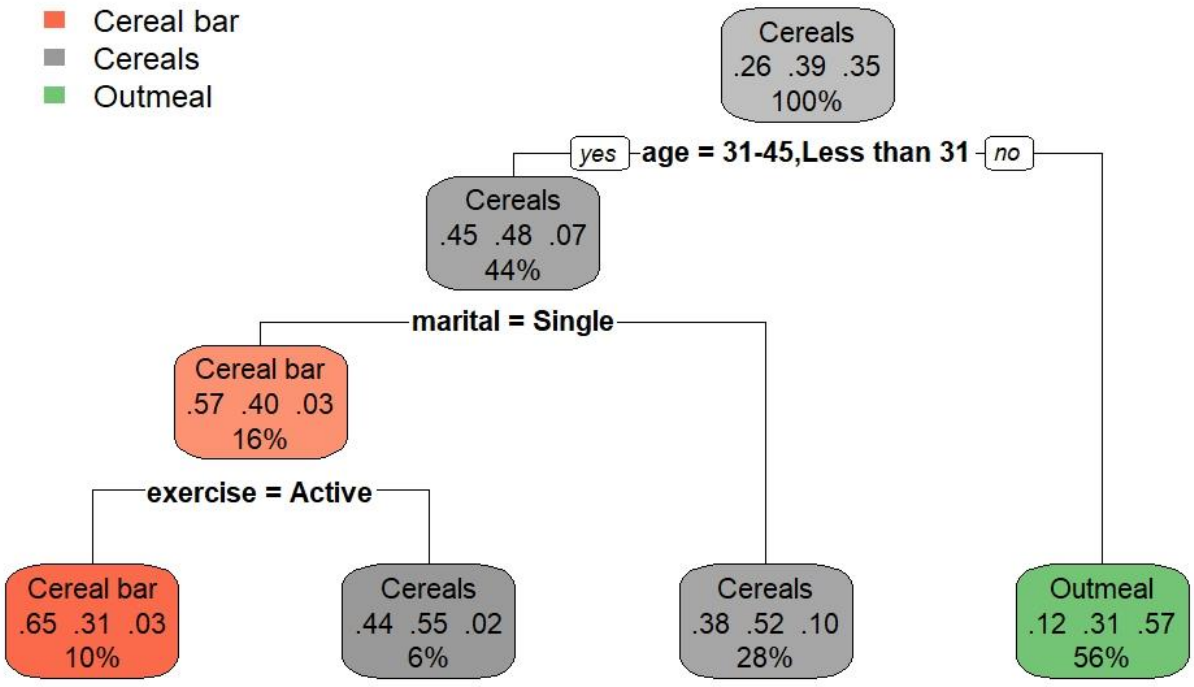
Analysis:

The model seems to perform more consistently on cereal bars, with a difference of only 1% accuracy between training data set and validation data set. There's a slight improvement in accuracy for cereals (4.79%) on the validation set, suggesting the model might generalize better for this category. The most significant improvement is in the oatmeal classification (8.68% increase), indicating the model performs better on unseen data for this category. The model shows moderate overall accuracy approximately 61.74% on the validation set, with some variation in class-specific performance. The slightly higher validation accuracy compared to training accuracy suggests the model might not be severely overfitting.

c. Recommendation for the company regarding the marketing campaign for promoting the sales of the oatmeal product.

- From the analysis, we observed that age is a significant factor in predicting breakfast preferences, so the company should customize ads for different age groups and notify them how important oatmeal is for a healthy lifestyle, which helps the company attract a larger audience and increase their sales.
- Highlight oatmeal's health benefits, especially for active people.
- Educate people about oatmeal's benefits and versatility. They could show different ways you can make oatmeal taste good and how you can eat it for breakfast in lots of ways.
- The company should produce the oatmeal products in different flavors so that they can increase sales of the products.

d. Decision tree graphic in RStudio



R code

```
#attach the data file to Rstudio

attach(Cerealdata_2024)


#Install Packages

install.packages("rpart.plot")

install.packages("randomForest")


# Load necessary libraries

library(randomForest)

library(rpart.plot)


# Convert categorical variables to factors

Cerealdata_2024$age <- as.factor(Cerealdata_2024$age)

Cerealdata_2024$gender <- as.factor(Cerealdata_2024$gender)

Cerealdata_2024$marital <- as.factor(Cerealdata_2024$marital)

Cerealdata_2024$exercise <- as.factor(Cerealdata_2024$exercise)

Cerealdata_2024$breakfast <- as.factor(Cerealdata_2024$breakfast)




# Creating training and validation sets
```

```
# Split data into training and validation sets
#Training Set: Validation Set =70:30 (random)
set.seed(200)
train_index <- sample(nrow(Cerealdata_2024), 0.7 * nrow(Cerealdata_2024), replace = FALSE)
trainset <- Cerealdata_2024[train_index,]
validset <- Cerealdata_2024[-train_index,]



#Create random forest model and evaluate prediction accuracy


# Create a random forest model
Model <- randomForest(breakfast ~ . , data = trainset, importance = TRUE)

# Analysis of predictors importance
print("Mean Decrease Accuracy:")
importance_acc <- importance(Model, type = 1)
print(importance_acc)

print("Mean Decrease Gini:")
importance_gini <- importance(Model, type = 2)
print(importance_gini)



# Predicting on training set
```

```
predtrain <- predict(Model, trainset, type = "class")

# Checking classification accuracy on training set
train_conf_matrix <- table(predtrain, trainset$breakfast)
print("Confusion Matrix for Training Set:")
print(train_conf_matrix)


# Predicting on validation set
predvalid <- predict(Model, validset, type = "class")

# Checking classification accuracy on validation set
valid_conf_matrix <- table(predvalid, validset$breakfast)
print("Confusion Matrix for Validation Set:")
print(valid_conf_matrix)

# Calculate accuracy on validation set
accuracy <- mean(predvalid == validset$breakfast)
print(paste("Accuracy on Validation Set:", accuracy))

#to check important variables
importance(Model)
varImpPlot(Model)
```

```
#To predict the values
#create data frame for new data
new_data <- data.frame(
  age = c("More than 60"),
  exercise   = c("Active"),
  marital = c("single"),
  gender = c("men"))

print(new_data)

# Provide hypothetical values for marital and gender
new_data$marital <- factor("Single", levels = levels(trainset$marital))
new_data$gender <- factor("Men", levels = levels(trainset$gender))

#convert relevant columns to factors if they are categorical in the original data
new_data$age <- factor (new_data$age, levels = levels (trainset$age))
new_data$gender <- factor (new_data$gender,levels = levels(trainset$gender))
new_data$marital <- factor (new_data$marital,levels = levels(trainset$marital))
new_data$exercise <- factor (new_data$exercise, levels = levels(trainset$exercise))

#make predictions using the trained random forest model
predicted_values <- predict(model1, new_data)

#print or use the predicted values as needed
print(predicted_values)
```

```
#Decision tree graphic


# Train a decision tree model
tree_model <- rpart(breakfast ~ ., data = Cerealdata_2024, method = "class")


# Plot the decision tree
rpart.plot(tree_model)
```