Project 2

Waheeb Algabri, William Berritt, Kossi Akplaka

2024-07-12

# Loading libraries

library(caret)  
library(readxl)  
library(dplyr)  
library(car)  
library(xgboost)  
library(e1071)   
library(mice)  
library(fastDummies)  
library(reshape2)  
library(openxlsx)

# Project Summary

You are given a simple data set from a beverage manufacturing company. It consists of 2,571 rows/cases of data and 33 columns / variables. Your goal is to use this data to predict PH (a column in the set). Potential for hydrogen (pH) is a measure of acidity/alkalinity, it must conform in a critical range and therefore it is important to understand its influence and predict its values. This is production data. pH is a KPI, Key Performance Indicator.

The objective of this project is to predict the pH value using the data provided

# Data loading and exploration

We’ve uploaded the data into GitHub for reproducibility

# Load the dataset  
StudentData<- read.csv(  
 "https://raw.githubusercontent.com/waheeb123/Data-624/main/Projects/Project-2/StudentData%20-%20TO%20MODEL.csv")  
  
head(StudentData)

## Brand.Code Carb.Volume Fill.Ounces PC.Volume Carb.Pressure Carb.Temp PSC  
## 1 B 5.340000 23.96667 0.2633333 68.2 141.2 0.104  
## 2 A 5.426667 24.00667 0.2386667 68.4 139.6 0.124  
## 3 B 5.286667 24.06000 0.2633333 70.8 144.8 0.090  
## 4 A 5.440000 24.00667 0.2933333 63.0 132.6 NA  
## 5 A 5.486667 24.31333 0.1113333 67.2 136.8 0.026  
## 6 A 5.380000 23.92667 0.2693333 66.6 138.4 0.090  
## PSC.Fill PSC.CO2 Mnf.Flow Carb.Pressure1 Fill.Pressure Hyd.Pressure1  
## 1 0.26 0.04 -100 118.8 46.0 0  
## 2 0.22 0.04 -100 121.6 46.0 0  
## 3 0.34 0.16 -100 120.2 46.0 0  
## 4 0.42 0.04 -100 115.2 46.4 0  
## 5 0.16 0.12 -100 118.4 45.8 0  
## 6 0.24 0.04 -100 119.6 45.6 0  
## Hyd.Pressure2 Hyd.Pressure3 Hyd.Pressure4 Filler.Level Filler.Speed  
## 1 NA NA 118 121.2 4002  
## 2 NA NA 106 118.6 3986  
## 3 NA NA 82 120.0 4020  
## 4 0 0 92 117.8 4012  
## 5 0 0 92 118.6 4010  
## 6 0 0 116 120.2 4014  
## Temperature Usage.cont Carb.Flow Density MFR Balling Pressure.Vacuum PH  
## 1 66.0 16.18 2932 0.88 725.0 1.398 -4.0 8.36  
## 2 67.6 19.90 3144 0.92 726.8 1.498 -4.0 8.26  
## 3 67.0 17.76 2914 1.58 735.0 3.142 -3.8 8.94  
## 4 65.6 17.42 3062 1.54 730.6 3.042 -4.4 8.24  
## 5 65.6 17.68 3054 1.54 722.8 3.042 -4.4 8.26  
## 6 66.2 23.82 2948 1.52 738.8 2.992 -4.4 8.32  
## Oxygen.Filler Bowl.Setpoint Pressure.Setpoint Air.Pressurer Alch.Rel Carb.Rel  
## 1 0.022 120 46.4 142.6 6.58 5.32  
## 2 0.026 120 46.8 143.0 6.56 5.30  
## 3 0.024 120 46.6 142.0 7.66 5.84  
## 4 0.030 120 46.0 146.2 7.14 5.42  
## 5 0.030 120 46.0 146.2 7.14 5.44  
## 6 0.024 120 46.0 146.6 7.16 5.44  
## Balling.Lvl  
## 1 1.48  
## 2 1.56  
## 3 3.28  
## 4 3.04  
## 5 3.04  
## 6 3.02

The training dataset has 33 columns including a categorical variable Brand Code and other predictors of the pH value.

# Data exploration and cleaning

* Average PH in the beverage

Let’s calculate the average PH in the dataset

# Calculate mean of 'PH' column, handling NA values  
mean\_PH <- mean(StudentData$PH, na.rm = TRUE)  
  
# Print the mean pH  
print(paste("The mean pH after handling NA values is:", mean\_PH))

## [1] "The mean pH after handling NA values is: 8.54564861706272"

The average pH of an beverage is 8.54. Water naturally varies between about 6.5 and 8.5 on the pH scale. Bottled waters labeled as alkaline can be 8 and 9.

* Different brands of beverage

Categories in Brand Code and Average pH

# Convert 'Brand Code' to factor (categorical)  
StudentData\_clean$Brand.Code <- as.factor(StudentData\_clean$Brand.Code)

## Error in eval(expr, envir, enclos): object 'StudentData\_clean' not found

# Calculate average pH for each category in 'Brand Code'  
brand\_pH\_avg <- StudentData\_clean %>%  
 group\_by(`Brand.Code`) %>%  
 summarise(avg\_pH = mean(PH, na.rm = TRUE))

## Error in eval(expr, envir, enclos): object 'StudentData\_clean' not found

brand\_pH\_avg

## Error in eval(expr, envir, enclos): object 'brand\_pH\_avg' not found

We have 4 different brands of beverage being manufactured and have some missing data in the Brand Code column

* Summary of the data

summary(StudentData)

## Brand.Code Carb.Volume Fill.Ounces PC.Volume   
## Length:2571 Min. :5.040 Min. :23.63 Min. :0.07933   
## Class :character 1st Qu.:5.293 1st Qu.:23.92 1st Qu.:0.23917   
## Mode :character Median :5.347 Median :23.97 Median :0.27133   
## Mean :5.370 Mean :23.97 Mean :0.27712   
## 3rd Qu.:5.453 3rd Qu.:24.03 3rd Qu.:0.31200   
## Max. :5.700 Max. :24.32 Max. :0.47800   
## NA's :10 NA's :38 NA's :39   
## Carb.Pressure Carb.Temp PSC PSC.Fill   
## Min. :57.00 Min. :128.6 Min. :0.00200 Min. :0.0000   
## 1st Qu.:65.60 1st Qu.:138.4 1st Qu.:0.04800 1st Qu.:0.1000   
## Median :68.20 Median :140.8 Median :0.07600 Median :0.1800   
## Mean :68.19 Mean :141.1 Mean :0.08457 Mean :0.1954   
## 3rd Qu.:70.60 3rd Qu.:143.8 3rd Qu.:0.11200 3rd Qu.:0.2600   
## Max. :79.40 Max. :154.0 Max. :0.27000 Max. :0.6200   
## NA's :27 NA's :26 NA's :33 NA's :23   
## PSC.CO2 Mnf.Flow Carb.Pressure1 Fill.Pressure   
## Min. :0.00000 Min. :-100.20 Min. :105.6 Min. :34.60   
## 1st Qu.:0.02000 1st Qu.:-100.00 1st Qu.:119.0 1st Qu.:46.00   
## Median :0.04000 Median : 65.20 Median :123.2 Median :46.40   
## Mean :0.05641 Mean : 24.57 Mean :122.6 Mean :47.92   
## 3rd Qu.:0.08000 3rd Qu.: 140.80 3rd Qu.:125.4 3rd Qu.:50.00   
## Max. :0.24000 Max. : 229.40 Max. :140.2 Max. :60.40   
## NA's :39 NA's :2 NA's :32 NA's :22   
## Hyd.Pressure1 Hyd.Pressure2 Hyd.Pressure3 Hyd.Pressure4   
## Min. :-0.80 Min. : 0.00 Min. :-1.20 Min. : 52.00   
## 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: 86.00   
## Median :11.40 Median :28.60 Median :27.60 Median : 96.00   
## Mean :12.44 Mean :20.96 Mean :20.46 Mean : 96.29   
## 3rd Qu.:20.20 3rd Qu.:34.60 3rd Qu.:33.40 3rd Qu.:102.00   
## Max. :58.00 Max. :59.40 Max. :50.00 Max. :142.00   
## NA's :11 NA's :15 NA's :15 NA's :30   
## Filler.Level Filler.Speed Temperature Usage.cont Carb.Flow   
## Min. : 55.8 Min. : 998 Min. :63.60 Min. :12.08 Min. : 26   
## 1st Qu.: 98.3 1st Qu.:3888 1st Qu.:65.20 1st Qu.:18.36 1st Qu.:1144   
## Median :118.4 Median :3982 Median :65.60 Median :21.79 Median :3028   
## Mean :109.3 Mean :3687 Mean :65.97 Mean :20.99 Mean :2468   
## 3rd Qu.:120.0 3rd Qu.:3998 3rd Qu.:66.40 3rd Qu.:23.75 3rd Qu.:3186   
## Max. :161.2 Max. :4030 Max. :76.20 Max. :25.90 Max. :5104   
## NA's :20 NA's :57 NA's :14 NA's :5 NA's :2   
## Density MFR Balling Pressure.Vacuum   
## Min. :0.240 Min. : 31.4 Min. :-0.170 Min. :-6.600   
## 1st Qu.:0.900 1st Qu.:706.3 1st Qu.: 1.496 1st Qu.:-5.600   
## Median :0.980 Median :724.0 Median : 1.648 Median :-5.400   
## Mean :1.174 Mean :704.0 Mean : 2.198 Mean :-5.216   
## 3rd Qu.:1.620 3rd Qu.:731.0 3rd Qu.: 3.292 3rd Qu.:-5.000   
## Max. :1.920 Max. :868.6 Max. : 4.012 Max. :-3.600   
## NA's :1 NA's :212 NA's :1   
## PH Oxygen.Filler Bowl.Setpoint Pressure.Setpoint  
## Min. :7.880 Min. :0.00240 Min. : 70.0 Min. :44.00   
## 1st Qu.:8.440 1st Qu.:0.02200 1st Qu.:100.0 1st Qu.:46.00   
## Median :8.540 Median :0.03340 Median :120.0 Median :46.00   
## Mean :8.546 Mean :0.04684 Mean :109.3 Mean :47.62   
## 3rd Qu.:8.680 3rd Qu.:0.06000 3rd Qu.:120.0 3rd Qu.:50.00   
## Max. :9.360 Max. :0.40000 Max. :140.0 Max. :52.00   
## NA's :4 NA's :12 NA's :2 NA's :12   
## Air.Pressurer Alch.Rel Carb.Rel Balling.Lvl   
## Min. :140.8 Min. :5.280 Min. :4.960 Min. :0.00   
## 1st Qu.:142.2 1st Qu.:6.540 1st Qu.:5.340 1st Qu.:1.38   
## Median :142.6 Median :6.560 Median :5.400 Median :1.48   
## Mean :142.8 Mean :6.897 Mean :5.437 Mean :2.05   
## 3rd Qu.:143.0 3rd Qu.:7.240 3rd Qu.:5.540 3rd Qu.:3.14   
## Max. :148.2 Max. :8.620 Max. :6.060 Max. :3.66   
## NA's :9 NA's :10 NA's :1

The summary of the StudentData dataset provides a comprehensive overview of key metrics used in beverage manufacturing. It indicates that variables like carbonation volume, fill ounces, and pressures (both carbonation and hydraulic) span a range from minimum to maximum values, suggesting variability in production conditions.  
The presence of missing values in some variables, such as pH and oxygen filler, indicates potential data gaps that could affect the accuracy of the model.

* Missing values

colSums(is.na(StudentData))

## Brand.Code Carb.Volume Fill.Ounces PC.Volume   
## 0 10 38 39   
## Carb.Pressure Carb.Temp PSC PSC.Fill   
## 27 26 33 23   
## PSC.CO2 Mnf.Flow Carb.Pressure1 Fill.Pressure   
## 39 2 32 22   
## Hyd.Pressure1 Hyd.Pressure2 Hyd.Pressure3 Hyd.Pressure4   
## 11 15 15 30   
## Filler.Level Filler.Speed Temperature Usage.cont   
## 20 57 14 5   
## Carb.Flow Density MFR Balling   
## 2 1 212 1   
## Pressure.Vacuum PH Oxygen.Filler Bowl.Setpoint   
## 0 4 12 2   
## Pressure.Setpoint Air.Pressurer Alch.Rel Carb.Rel   
## 12 0 9 10   
## Balling.Lvl   
## 1

# Data Cleaning

* Cleaning the data using the mice package

MICE (Multivariate Imputation by Chained Equations) is a robust technique for handling missing data in datasets. It imputes missing values by modeling each variable with missing data as a function of other variables, preserving relationships and uncertainty through multiple imputations.

# Feature engineering

After dealing with the missing values in the dataset, we will create dummies varibles for the categorical variable. This enhance the interpretability of the model results. Each dummy variable represents a specific category, making it easier to understand the impact of different categories on the prediction outcome.

* Create dummy variables for ‘Brand Code’

# Create dummies variables  
StudentData\_clean2 <- fastDummies::dummy\_cols(StudentData\_clean,  
 remove\_first\_dummy = TRUE)  
  
# Remove the Brand Code column  
StudentData\_clean2 <- subset(StudentData\_clean2, select = - Brand.Code)  
  
# Print the 5 elements of the data  
head(StudentData\_clean2)

## Carb.Volume Fill.Ounces PC.Volume Carb.Pressure Carb.Temp PSC PSC.Fill  
## 1 5.340000 23.96667 0.2633333 68.2 141.2 0.104 0.26  
## 2 5.426667 24.00667 0.2386667 68.4 139.6 0.124 0.22  
## 3 5.286667 24.06000 0.2633333 70.8 144.8 0.090 0.34  
## 4 5.440000 24.00667 0.2933333 63.0 132.6 0.114 0.42  
## 5 5.486667 24.31333 0.1113333 67.2 136.8 0.026 0.16  
## 6 5.380000 23.92667 0.2693333 66.6 138.4 0.090 0.24  
## PSC.CO2 Mnf.Flow Carb.Pressure1 Fill.Pressure Hyd.Pressure1 Hyd.Pressure2  
## 1 0.04 -100 118.8 46.0 0 0  
## 2 0.04 -100 121.6 46.0 0 0  
## 3 0.16 -100 120.2 46.0 0 0  
## 4 0.04 -100 115.2 46.4 0 0  
## 5 0.12 -100 118.4 45.8 0 0  
## 6 0.04 -100 119.6 45.6 0 0  
## Hyd.Pressure3 Hyd.Pressure4 Filler.Level Filler.Speed Temperature Usage.cont  
## 1 0 118 121.2 4002 66.0 16.18  
## 2 0 106 118.6 3986 67.6 19.90  
## 3 0 82 120.0 4020 67.0 17.76  
## 4 0 92 117.8 4012 65.6 17.42  
## 5 0 92 118.6 4010 65.6 17.68  
## 6 0 116 120.2 4014 66.2 23.82  
## Carb.Flow Density MFR Balling Pressure.Vacuum PH Oxygen.Filler  
## 1 2932 0.88 725.0 1.398 -4.0 8.36 0.022  
## 2 3144 0.92 726.8 1.498 -4.0 8.26 0.026  
## 3 2914 1.58 735.0 3.142 -3.8 8.94 0.024  
## 4 3062 1.54 730.6 3.042 -4.4 8.24 0.030  
## 5 3054 1.54 722.8 3.042 -4.4 8.26 0.030  
## 6 2948 1.52 738.8 2.992 -4.4 8.32 0.024  
## Bowl.Setpoint Pressure.Setpoint Air.Pressurer Alch.Rel Carb.Rel Balling.Lvl  
## 1 120 46.4 142.6 6.58 5.32 1.48  
## 2 120 46.8 143.0 6.56 5.30 1.56  
## 3 120 46.6 142.0 7.66 5.84 3.28  
## 4 120 46.0 146.2 7.14 5.42 3.04  
## 5 120 46.0 146.2 7.14 5.44 3.04  
## 6 120 46.0 146.6 7.16 5.44 3.02  
## Brand.Code\_A Brand.Code\_B Brand.Code\_C Brand.Code\_D  
## 1 0 1 0 0  
## 2 1 0 0 0  
## 3 0 1 0 0  
## 4 1 0 0 0  
## 5 1 0 0 0  
## 6 1 0 0 0

# Outlier Detection and Removal

* Remove outliers in the data

We defined a function remove\_outliers\_specific to remove outliers based on the interquartile range Q3 and Q1 for the PH column.

# Function to remove rows containing outliers based on IQR for a specific column  
remove\_outliers\_specific <- function(df, column\_name) {  
 # Copy the dataframe to avoid modifying the original  
 df\_clean <- df  
   
 # Calculate quartiles and IQR for the specified column  
 Q1 <- quantile(df[[column\_name]], 0.25)  
 Q3 <- quantile(df[[column\_name]], 0.75)  
 IQR\_val <- Q3 - Q1  
   
 # Define the lower and upper bounds for outliers  
 lower\_bound <- Q1 - 1.5 \* IQR\_val  
 upper\_bound <- Q3 + 1.5 \* IQR\_val  
   
 # Identify rows with outliers in the specified column  
 outliers <- df[[column\_name]] < lower\_bound | df[[column\_name]] > upper\_bound  
   
 # Subset the dataframe to keep only rows without outliers for the specified column  
 df\_clean <- df[!outliers, ]  
   
 return(df\_clean)  
}  
  
# Remove outliers specifically from the PH column  
StudentData\_clean3 <- remove\_outliers\_specific(StudentData\_clean2, "PH")

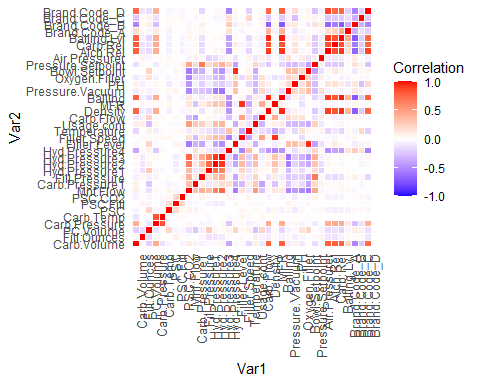
pH outside the quartile range Q3 and Q1 have been dropped. There were 18 outliers in the data that were removed.

# Data Exploration

First, we will compute the correlation and visualize the correlation matrix

* Correlation and visualization

# Calculate the correlation matrix  
correlation\_matrix <- cor(StudentData\_clean3)  
  
# Convert the correlation matrix to long format for plotting  
cor\_melted <- melt(correlation\_matrix)  
  
# Plot the heatmap  
ggplot(cor\_melted, aes(Var1, Var2, fill = value)) +  
 geom\_tile(color = "white") +  
 scale\_fill\_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0,  
 limit = c(-1, 1), space = "Lab", name="Correlation") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 1, size = 10, hjust = 1)) +  
 coord\_fixed()



There are high collinearities between the variables Carb Pressure, Hyd Pressure3, Balling Lvl, and Carb Rel.

* Remove collinearities

Let’s remove the collinearities to improve the model performance

StudentData\_clean3 <- StudentData\_clean3[, !colnames(StudentData\_clean3) %in% "Carb Pressure"]  
StudentData\_clean3 <- StudentData\_clean3[, !colnames(StudentData\_clean3) %in% "Hyd Pressure3"]  
StudentData\_clean3 <- StudentData\_clean3[, !colnames(StudentData\_clean3) %in% "Balling Lvl"]  
StudentData\_clean3 <- StudentData\_clean3[, !colnames(StudentData\_clean3) %in% "Carb Rel"]

# Model Building

We split the data 80% to 20% between training and testing set

set.seed(123) # for reproducibility  
trainIndex <- createDataPartition(StudentData\_clean3$PH, p = 0.8, list = FALSE)  
trainData <- StudentData\_clean3[trainIndex, ]  
testData <- StudentData\_clean3[-trainIndex, ]

* Linear regression model

Linear models provide straightforward interpretation of coefficients. Each predictor’s coefficient indicates the strength and direction of its relationship with the dependent variable. This makes it easy to understand the impact of each predictor on the outcome.

set.seed(123)  
# Linear Regression  
model\_lm <- lm(PH ~ ., data = trainData)  
predictions\_lm <- predict(model\_lm, newdata = testData)  
rsq\_lm <- cor(predictions\_lm, testData$PH)^2

* XGBoost model

XGBoost have better predictive accuracy and ability to handle complex relationships. Its ensemble learning approach iteratively improves model performance by sequentially correcting errors, making it effective at capturing non-linear relationships between predictors and pH levels.

* Support Vector Regression model

Support Vector Machines (SVMs) are another powerful technique for predicting beverage pH based on other variables. SVMs are effective in scenarios where the relationship between predictors and the outcome (pH) is not necessarily linear and can exhibit complex patterns. SVMs work by finding the optimal hyperplane to predict continuous outcomes. SVMs can handle datasets with a small number of samples

set.seed(123)  
# Support Vector Regression (SVR)  
model\_svr <- svm(PH ~ ., data = trainData)  
predictions\_svr <- predict(model\_svr, newdata = testData)  
rsq\_svr <- cor(predictions\_svr, testData$PH)^2

* Interpret the result

Print R-squared values for comparison

cat("Linear Regression R-squared:", rsq\_lm, "\n")

## Linear Regression R-squared: 0.4137368

cat("XGBoost R-squared:", rsq\_xgb, "\n")

## XGBoost R-squared: 0.6760822

cat("Support Vector Regression R-squared:", rsq\_svr, "\n")

## Support Vector Regression R-squared: 0.5696495

The R-squared values provided for the different models—Linear Regression (0.41), XGBoost (0.68), and Support Vector Regression (0.57) reflect how well each model explains the variability in predicting beverage pH based on other variables.

R-squared values highlight the superior predictive performance of XGBoost over linear regression and SVR in modeling beverage pH based on other variables, emphasizing its ability to handle complex relationships and improve model accuracy.

* Most important predictors of pH using XGBoost

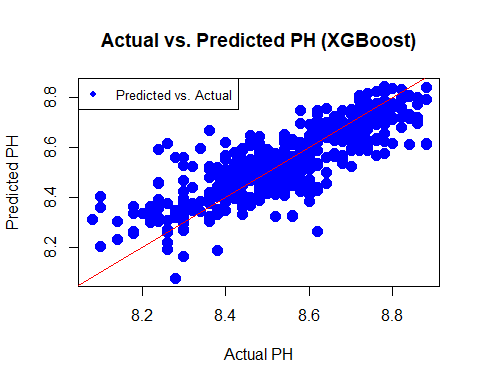
# Extract feature importance  
importance\_scores <- xgb.importance(model = model\_xgb)  
  
# Print the top 5  
head(importance\_scores, 5)

## Feature Gain Cover Frequency  
## 1: Mnf.Flow 0.15852103 0.037455992 0.029472596  
## 2: Oxygen.Filler 0.05586902 0.039379422 0.039038263  
## 3: Brand.Code\_C 0.04874221 0.006395774 0.004653568  
## 4: Carb.Pressure1 0.04747544 0.054596354 0.044984488  
## 5: Alch.Rel 0.04672519 0.035933603 0.020682523

Features like Mnf.Flow, Brand Codes, Oxygen Filler, Pressure are the highest predictors of the pH’s value.

* Comparing actual vs. predicted values

plot(testData$PH, predictions\_xgb,   
 main = "Actual vs. Predicted PH (XGBoost)",  
 xlab = "Actual PH",  
 ylab = "Predicted PH",  
 col = "blue",  
 pch = 19,  
 cex = 1.5)  
  
# Add a diagonal line to show perfect predictions  
abline(0, 1, col = "red")  
  
# Add legend  
legend("topleft", legend = "Predicted vs. Actual", col = c("blue", "red"), pch = c(19, NA), lty = c(NA, 1), cex = 0.8)



# Scoring using the best prediction

Load the data

# Load necessary libraries  
library(readxl) # For reading Excel files  
  
# Read new data from Excel file  
data <- read.csv("https://raw.githubusercontent.com/waheeb123/Data-624/main/Projects/Project-2/StudentEvaluation-%20TO%20PREDICT.csv")

Replace null values in Brand Code with the most common brand

brand\_counts <- table(data$Brand.Code)  
most\_common\_brand <- names(which.max(brand\_counts))  
data$Brand.Code[is.na(data$Brand.Code)] <- most\_common\_brand

Let’s impute the data using the package mice

Remove collinearities

submission\_df <- submission\_df[, !colnames(submission\_df) %in% "Carb Pressure"]  
submission\_df <- submission\_df[, !colnames(submission\_df) %in% "Hyd Pressure3"]  
submission\_df <- submission\_df[, !colnames(submission\_df) %in% "Balling Lvl"]  
submission\_df <- submission\_df[, !colnames(submission\_df) %in% "Carb Rel"]

Create dummy variables

# Create dummies variables  
submission\_df <- fastDummies::dummy\_cols(submission\_df,  
 remove\_first\_dummy = TRUE)  
  
# Remove the Brand Code column  
submission\_df <- subset(submission\_df, select = - Brand.Code)  
  
# Print the 5 elements of the data  
head(submission\_df)

## Carb.Volume Fill.Ounces PC.Volume Carb.Pressure Carb.Temp PSC PSC.Fill  
## 1 5.480000 24.03333 0.2700000 65.4 134.6 0.236 0.40  
## 2 5.393333 23.95333 0.2266667 63.2 135.0 0.042 0.22  
## 3 5.293333 23.92000 0.3033333 66.4 140.4 0.068 0.10  
## 4 5.266667 23.94000 0.1860000 64.8 139.0 0.004 0.20  
## 5 5.406667 24.20000 0.1600000 69.4 142.2 0.040 0.30  
## 6 5.286667 24.10667 0.2120000 73.4 147.2 0.078 0.22  
## PSC.CO2 Mnf.Flow Carb.Pressure1 Fill.Pressure Hyd.Pressure1 Hyd.Pressure2  
## 1 0.04 -100 116.6 46.0 0 0  
## 2 0.08 -100 118.8 46.2 0 0  
## 3 0.02 -100 120.2 45.8 0 0  
## 4 0.02 -100 124.8 40.0 0 0  
## 5 0.06 -100 115.0 51.4 0 0  
## 6 0.02 -100 118.6 46.4 0 0  
## Hyd.Pressure3 Hyd.Pressure4 Filler.Level Filler.Speed Temperature Usage.cont  
## 1 0 96 129.4 3986 66.0 21.66  
## 2 0 112 120.0 4012 65.6 17.60  
## 3 0 98 119.4 4010 65.6 24.18  
## 4 0 132 120.2 1006 74.4 18.12  
## 5 0 94 116.0 4018 66.4 21.32  
## 6 0 94 120.4 4010 66.6 18.00  
## Carb.Flow Density MFR Balling Pressure.Vacuum PH Oxygen.Filler  
## 1 2950 0.88 727.6 1.398 -3.8 NA 0.022  
## 2 2916 1.50 735.8 2.942 -4.4 NA 0.030  
## 3 3056 0.90 734.8 1.448 -4.2 NA 0.046  
## 4 28 0.74 290.6 1.056 -4.0 NA 0.186  
## 5 3214 0.88 752.0 1.398 -4.0 NA 0.082  
## 6 3064 0.84 732.0 1.298 -3.8 NA 0.064  
## Bowl.Setpoint Pressure.Setpoint Air.Pressurer Alch.Rel Carb.Rel Balling.Lvl  
## 1 130 45.2 142.6 6.56 5.34 1.48  
## 2 120 46.0 147.2 7.14 5.58 3.04  
## 3 120 46.0 146.6 6.52 5.34 1.46  
## 4 120 46.0 146.4 6.48 5.50 1.48  
## 5 120 50.0 145.8 6.50 5.38 1.46  
## 6 120 46.0 146.0 6.50 5.42 1.44  
## Brand.Code\_A Brand.Code\_B Brand.Code\_C Brand.Code\_D  
## 1 0 0 0 1  
## 2 1 0 0 0  
## 3 0 1 0 0  
## 4 0 1 0 0  
## 5 0 1 0 0  
## 6 0 1 0 0

Predict the PH using the XGBoost model

# Convert df\_test to xgb.DMatrix format  
submission\_df\_xgb <- xgb.DMatrix(as.matrix(  
 submission\_df[, -which(names(submission\_df) == "PH")]))  
  
# Predict using the XGBoost model  
predictions\_xgb <- predict(model\_xgb, submission\_df\_xgb)  
  
# Replace PH column in df\_test with predictions  
submission\_df$PH <- predictions\_xgb

Export the data

data$PH <- submission\_df$PH  
  
# Define file path for the Excel file  
file\_path <- "submission\_df\_proj\_2\_624.xlsx"  
  
# Export submission\_df to Excel  
write.xlsx(data, file\_path, rowNames = FALSE)