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)
library(xtable)
```

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/Stu

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

```r
# Calculate mean of 'PH' column, handling NA values
mean_PH <- mean(StudentData$PH, na.rm = TRUE)
```

```r
# 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

```r
# 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
```

```r
# 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
```

```r
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

```r
xtable(summary(StudentData))
```

```
## % latex table generated in R 4.3.1 by xtable 1.8-4 package
## % Sat Jul 13 13:41:04 2024
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlllllllllllllllllllllllllllll}
##   \hline
##  &  Brand.Code &  Carb.Volume &  Fill.Ounces &   PC.Volume & Carb.Pressure &   Carb.Temp &      PSC &
##   \hline
## X & Length:2571        & Min.   :5.040   & Min.   :23.63   & Min.   :0.07933   & Min.   :57.00   & Mi
##   X.1 & Class :character   & 1st Qu.:5.293   & 1st Qu.:23.92   & 1st Qu.:0.23917   & 1st Qu.:65.60
##   X.2 & Mode  :character   & Median :5.347   & Median :23.97   & Median :0.27133   & Median :68.20
##   X.3 &  & Mean   :5.370   & Mean   :23.97   & Mean   :0.27712   & Mean   :68.19   & Mean   :141.1
##   X.4 &  & 3rd Qu.:5.453   & 3rd Qu.:24.03   & 3rd Qu.:0.31200   & 3rd Qu.:70.60   & 3rd Qu.:143.8
##   X.5 &  & Max.   :5.700   & Max.   :24.32   & Max.   :0.47800   & Max.   :79.40   & Max.   :154.0
##   X.6 &  & NA's   :10   & NA's   :38   & NA's   :39   & NA's   :27   & NA's   :26   & NA's   :33   &
##    \hline
## \end{tabular}
## \end{table}
```

The summary of the StudentData dataset provides a comprehensive overview of key metrics used in beverag
The presence of missing values in some variables, such as pH and oxygen filler, indicates potential data

- 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 0.090     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

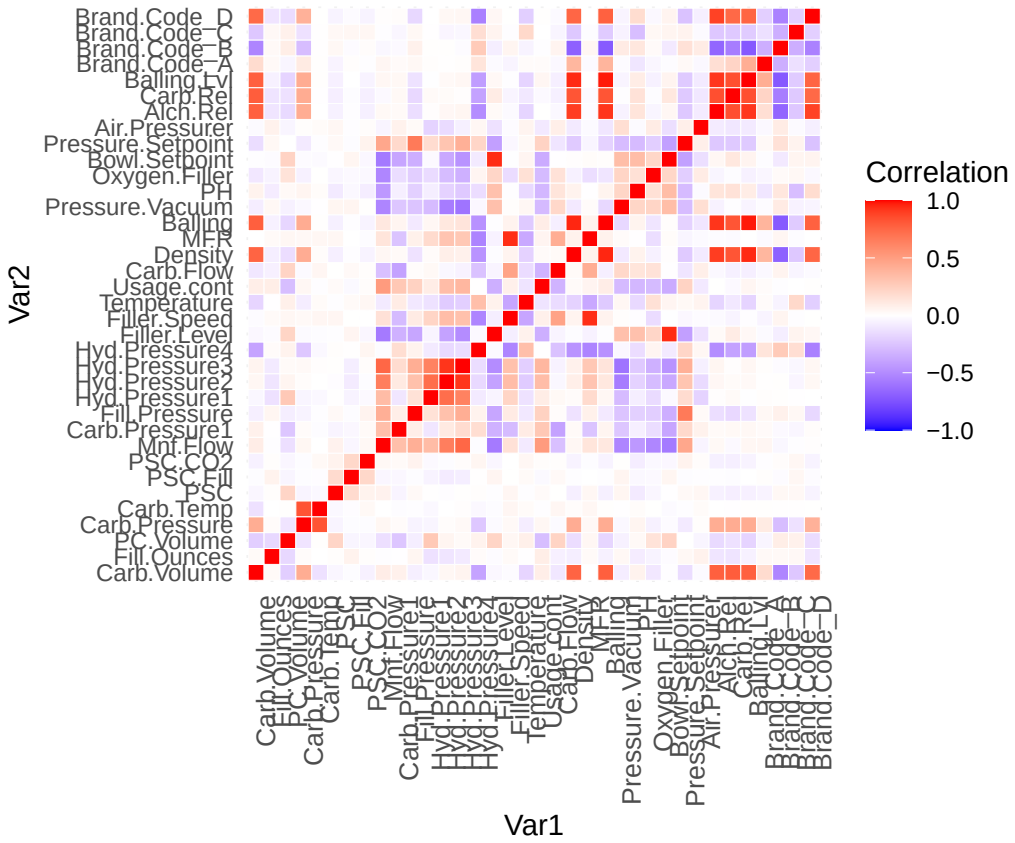
First, we will compute the correlation and visualize the correlation matrix

- 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.

```r
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

```r
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

```r
cat("Linear Regression R-squared:", rsq_lm, "\n")
```

```
## Linear Regression R-squared: 0.4226781
```

```r
cat("XGBoost R-squared:", rsq_xgb, "\n")
```

```
## XGBoost R-squared: 0.6961308
```

```r
cat("Support Vector Regression R-squared:", rsq_svr, "\n")
```

```
## Support Vector Regression R-squared: 0.5796009
```

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

```r
# 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 0.15919449 0.037069836 0.02771911
## 2: Oxygen.Filler 0.06310481 0.064568693 0.04276663
## 3:      Alch.Rel 0.05537701 0.030549265 0.02006336
## 4:  Brand.Code_C 0.05383211 0.007695543 0.00343189
## 5:    Usage.cont 0.05360539 0.050419496 0.03907075
```

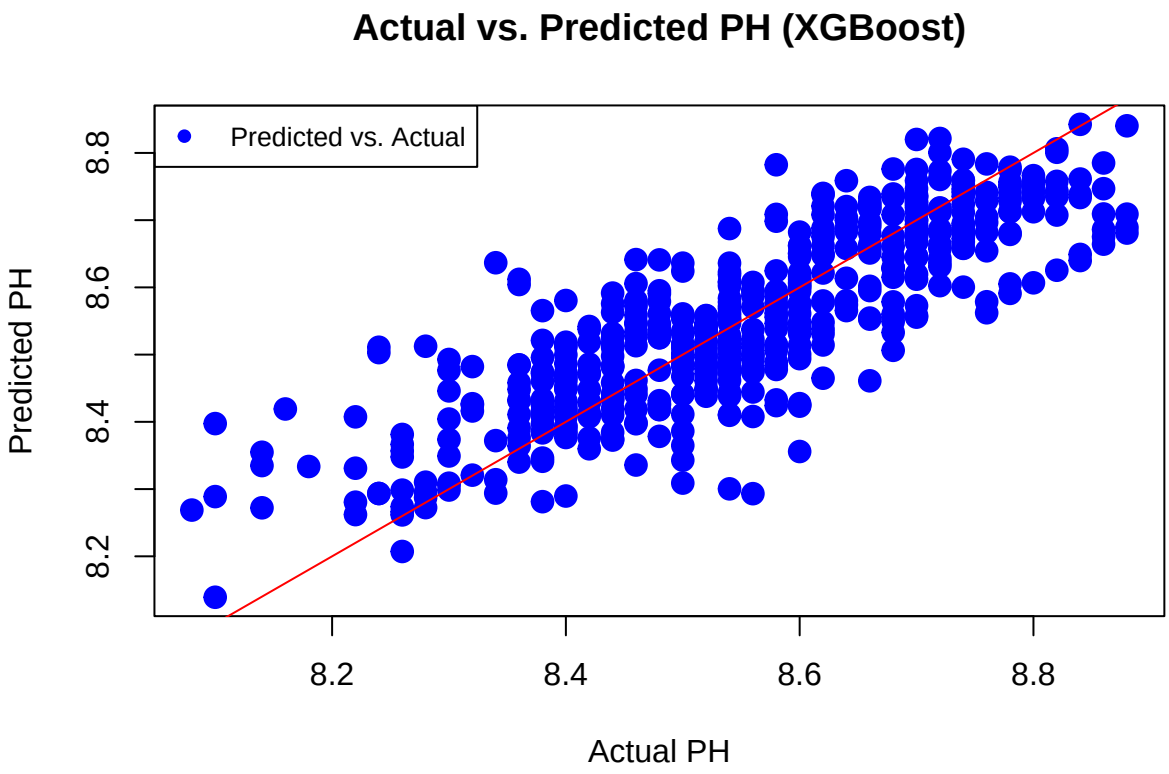
Features like Mnf.Flow, Brand Codes, Oxygen Filler, Pressure are the highest predictors of the pH's value.

- Comparing actual vs. predicted values

```r
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,
```



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/StudentE
```

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)
```
