Final Project

Matthew Tillmawitz, Heleine Fouda, Marjete Vucinaj, Lewris Mota, Kim

2024-12-15

Contents

1. Introduction

Recent regulatory changes emphasize the importance of having a comprehensive understanding of manufacturing processes and their impact on product quality. At ABC Beverage, pH levels are a critical parameter for ensuring consistency and maintaining product standards. This analysis aims to identify and quantify the factors driving pH variability while developing a predictive model that meets these new regulatory standards. By leveraging advanced analytical methods and predictive modeling techniques, the analysis provides data-driven insights to ensure compliance and enhance understanding of the production process. The scope of this analysis includes data preprocessing, the identification of predictive factors, and the development and validation of a robust predictive model for pH levels. The focus is on exploring relationships within the manufacturing data to uncover insights that can drive better decision-making. By implementing this structured approach, the findings will provide actionable insights into the key drivers of pH variability and a reliable framework for prediction.

Understanding and predicting pH levels is crucial not only for meeting regulatory standards but also for maintaining operational excellence. Accurate prediction will enable more efficient quality control, reducing variability and ensuring that ABC Beverage consistently delivers high-quality products. The results of this analysis will empower the company to address compliance needs while optimizing its manufacturing process, underscoring the importance of data-driven decision-making in the production environment.

2. Dataset Overview

The dataset comprises observations collected from a beverage production line, capturing information about carbonation levels, filling processes, environmental conditions, and quality control metrics. Each row represents a single production instance, and each column corresponds to a specific variable measured or controlled during the process.

```
# Create a dataframe in R
variables <- data.frame(
  Feature = c(
    "Brand Code", "Carb Volume", "Fill Ounces", "PC Volume", "Carb Pressure",
    "Carb Temp", "PSC", "PSC Fill", "PSC CO2", "Mnf Flow", "Carb Pressure1",
    "Fill Pressure", "Hyd Pressure1", "Hyd Pressure2", "Hyd Pressure3", "Hyd Pressure4",
    "Filler Level", "Filler Speed", "Temperature", "Usage cont", "Carb Flow",
    "Density", "MFR", "Balling", "Pressure Vacuum", "PH", "Oxygen Filler",
    "Bowl Setpoint", "Pressure Setpoint", "Air Pressure", "Alch Rel", "Carb Rel",
    "Balling Lvl"
  ),
```

```
  Description = c(
    "Unique identifier for the product's brand.",
    "Volume of carbon dioxide dissolved in the product.",
    "Volume of liquid dispensed into each container.",
    "Process control volume for monitoring liquid levels.",
    "Pressure level during carbonation.",
    "Temperature during carbonation for CO2 solubility.",
    "Process Setpoint Control for maintaining parameter targets.",
    "Filling setpoint under controlled conditions.",
    "Setpoint for CO2 levels during carbonation.",
    "Manufacturing flow rate for liquid or gas.",
    "Secondary carbonation pressure reading.",
    "Pressure applied during filling operations.",
    "Hydraulic pressure reading 1 for machine operation.",
    "Hydraulic pressure reading 2 for machine operation.",
    "Hydraulic pressure reading 3 for machine operation.",
    "Hydraulic pressure reading 4 for machine operation.",
    "Measurement of product level in containers.",
    "Speed of the filling machine or process.",
    "Temperature of the process environment.",
    "Container usage or consumption metrics.",
    "Flow rate of CO2 during carbonation.",
    "Density of the product for consistency monitoring.",
    "Mass flow rate of the material through the system.",
    "Sugar concentration level measured by the Balling scale.",
    "Vacuum pressure in the system.",
    "Acidity level of the product.",
    "Oxygen levels in the filling process.",
    "Target setpoint for intermediate container levels.",
    "Desired pressure level in the process.",
    "Air pressure measurement in the system.",
    "Alcohol release or related parameter.",
    "Carbonation release or related measurement.",
    "Sugar concentration level in the final product."
  ),
  Type = c(
    "Categorical", "Numerical", "Numerical", "Numerical", "Numerical",
    "Numerical", "Numerical", "Numerical", "Numerical", "Numerical",
    "Numerical", "Numerical", "Numerical", "Numerical", "Numerical",
    "Numerical", "Numerical", "Numerical", "Numerical", "Numerical",
    "Numerical", "Numerical", "Numerical", "Numerical", "Numerical",
    "Numerical", "Numerical", "Numerical", "Numerical", "Numerical",
    "Numerical", "Numerical", "Numerical"
  ),
  stringsAsFactors = FALSE
)

variables |> kable(caption = "Beverage Manufacture Process Features") |> kable_styling() |>  kable_class
```

Table 1: Beverage Manufacture Process Features

Feature	Description	Type

Brand Code	Unique identifier for the product's brand.	Categorical
Carb Volume	Volume of carbon dioxide dissolved in the product.	Numerical
Fill Ounces	Volume of liquid dispensed into each container.	Numerical
PC Volume	Process control volume for monitoring liquid levels.	Numerical
Carb Pressure	Pressure level during carbonation.	Numerical
Carb Temp	Temperature during carbonation for CO2 solubility.	Numerical
PSC	Process Setpoint Control for maintaining parameter targets.	Numerical
PSC Fill	Filling setpoint under controlled conditions.	Numerical
PSC CO2	Setpoint for CO2 levels during carbonation.	Numerical
Mnf Flow	Manufacturing flow rate for liquid or gas.	Numerical
Carb Pressure1	Secondary carbonation pressure reading.	Numerical
Fill Pressure	Pressure applied during filling operations.	Numerical
Hyd Pressure1	Hydraulic pressure reading 1 for machine operation.	Numerical
Hyd Pressure2	Hydraulic pressure reading 2 for machine operation.	Numerical
Hyd Pressure3	Hydraulic pressure reading 3 for machine operation.	Numerical
Hyd Pressure4	Hydraulic pressure reading 4 for machine operation.	Numerical
Filler Level	Measurement of product level in containers.	Numerical
Filler Speed	Speed of the filling machine or process.	Numerical
Temperature	Temperature of the process environment.	Numerical
Usage cont	Container usage or consumption metrics.	Numerical
Carb Flow	Flow rate of CO2 during carbonation.	Numerical
Density	Density of the product for consistency monitoring.	Numerical
MFR	Mass flow rate of the material through the system.	Numerical
Balling	Sugar concentration level measured by the Balling scale.	Numerical
Pressure Vacuum	Vacuum pressure in the system.	Numerical
PH	Acidity level of the product.	Numerical
Oxygen Filler	Oxygen levels in the filling process.	Numerical
Bowl Setpoint	Target setpoint for intermediate container levels.	Numerical
Pressure Setpoint	Desired pressure level in the process.	Numerical
Air Pressure	Air pressure measurement in the system.	Numerical
Alch Rel	Alcohol release or related parameter.	Numerical
Carb Rel	Carbonation release or related measurement.	Numerical
Balling Lvl	Sugar concentration level in the final product.	Numerical

The variables play a crucial role in capturing and monitoring various aspects of the production process, directly impacting product quality and operational efficiency:

- **Carbonation Variables**: These variables (e.g., `Carb Volume`, `Carb Pressure`, and `Carb Temp`) ensure the beverage achieves the desired level of fizziness and retains CO2 effectively. They are critical for meeting product specifications and customer satisfaction.
- **Filling Variables**:
  Variables like `Fill Ounces`, `PC Volume`, and `Filler Speed` ensure accurate and consistent product volume in containers, minimizing waste and maintaining packaging standards.
- **Quality Control Metrics**: Metrics such as `Density`, `Balling`, and `PSC` monitor the chemical and physical properties of the beverage, including sugar concentration, density, and carbonation, which significantly influence the product's pH balance. Maintaining the appropriate pH ensures flavor stability, microbial safety, and overall product quality.
- **Process Control Variables**: Variables like `PSC CO2` and `PSC` act as setpoints to maintain optimal operational conditions, reducing variability and enhancing production reliability.

3. Exploratory Data Analysis

4. Data Preparation

5. Model Development

This section outlines the methodology for building and implementing the predictive model. It includes details on data preprocessing, feature selection, hyperparameter tuning, and the modeling framework used. By leveraging advanced machine learning techniques, the objective is to create a robust, accurate, and generalizable model that captures the relationships between key variables and the target outcome.

```
manufacturing_tr <- read.csv("https://raw.githubusercontent.com/MarjeteV/data624/refs/heads/main/imputed

colnames(manufacturing_tr) <- gsub(" ", "_", colnames(manufacturing_tr))
brand_code_col <- c("Brand.CodeA","Brand.CodeB","Brand.CodeC","Brand.CodeD")
```

5.1 Support Vector Machine This section outlines the training and tuning of a Support Vector Machine with Gaussian Radial Basis Function Kernel to predict pH levels, leveraging the model's ability to handle non-linear relationships. The SVM model was selected due to its flexibility in capturing complex patterns in the data, making it well-suited for the task. Using a Gaussian Radial Basis Function (RBF) kernel, the SVM transforms input data into a higher-dimensional space by computing the similarity between data points. This transformation enables the model to find an optimal decision boundary or regression function in the new space, effectively capturing non-linear relationships between predictors and the target variable. This approach is particularly valuable for addressing the variability observed in pH levels, where intricate interactions among features may influence the outcome.

5.2 Support Vector Machine Model Preprocessing The dataset is split into training (75%) and testing (25%) subsets using random sampling to ensure the model is trained on a representative and diverse portion of the data while reserving an independent set for unbiased performance validation. Feature selection is applied prior to fitting a Radial Support Vector Machine (SVM) model to enhance model performance and efficiency. Recursive Feature Elimination (RFE) is employed as a robust method to identify the most relevant predictors by systematically ranking features based on their importance and iteratively removing those with minimal contribution. The dataset includes a categorical variable, Brand Code, which is transformed using target encoding. Target encoding replaces each category with the mean of the target variable for that category, allowing the relationship between the categorical variable and the target to be represented numerically. This transformation ensures compatibility with the RFE process by converting the categorical variable into a format that can be used effectively in feature selection. A Random Forest (rfFuncs) is used to evaluate feature importance, with performance assessed through 5-fold cross-validation to ensure the reliability and robustness of the selected feature subset.

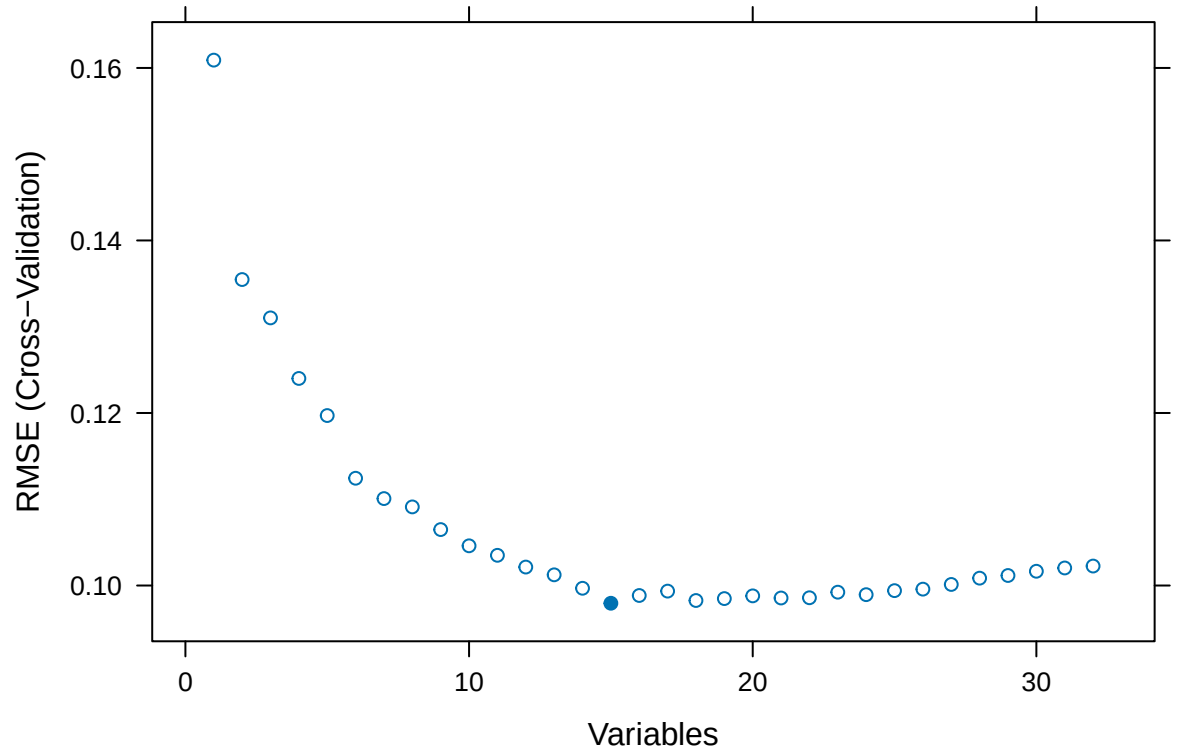
```
set.seed(8675309)

trainIndex <- sample(1:nrow(manufacturing_tr), size = 0.75 * nrow(manufacturing_tr))

beverage_man_train <- manufacturing_tr[trainIndex, ]
beverage_man_test <- manufacturing_tr[-trainIndex, ]
```

```
rfe_dataset <- beverage_man_train %>%
  rowwise() %>%
  mutate(brand_code = case_when(
    Brand.CodeA == 1 ~ "A",
    Brand.CodeB == 1 ~ "B",
```

```
    Brand.CodeC == 1 ~ "C",
    Brand.CodeD == 1 ~ "D",
    TRUE ~ NA_character_
  )) %>%
  ungroup() %>% group_by(brand_code) %>%
  mutate(brand_code_encoded = mean(PH)) |> ungroup() |>
  select(-c(Brand.CodeA, Brand.CodeB, Brand.CodeC, Brand.CodeD,"brand_code"))
```

```
set.seed(8675309)
ignore_col <- c("PH")

control <- rfeControl(functions = rfFuncs, method = "cv", number = 5,
                      allowParallel = T)

# Perform RFE
rfe_results <- rfe(
    rfe_dataset |> select(-all_of(ignore_col)),
  rfe_dataset$PH,
  sizes = c(1:length(rfe_dataset)),
  rfeControl = control
)
```

```
plot(rfe_results)
```



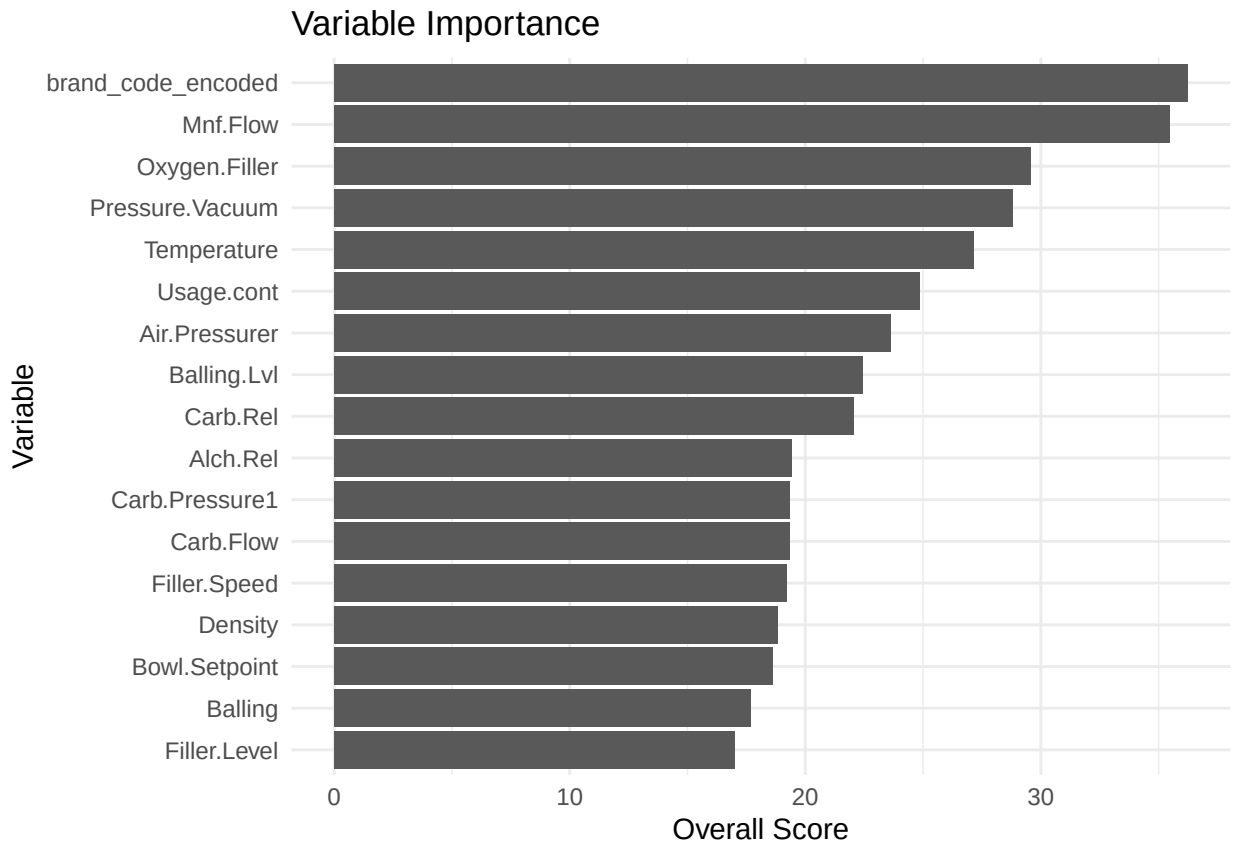
```
rfeMaxR2 <- max(rfe_results$resample[,"RMSE"])
rfeMinR2 <- min(rfe_results$resample[,"RMSE"])

rfe_results$resample |> kable(caption = "Recursive Feature Elimination Results") |> kable_styling() |>
```

Table 2: Recursive Feature Elimination Results

	Variables	RMSE	Rsquared	MAE	Resample
15	15	0.1003364	0.6547349	0.0729730	Fold1
47	15	0.0903284	0.7080149	0.0664764	Fold2
79	15	0.1015991	0.6714153	0.0743822	Fold3
111	15	0.1016013	0.7066444	0.0725293	Fold4
143	15	0.0958548	0.6906337	0.0677522	Fold5

The plot generated from the Recursive Feature Elimination (RFE) process shows that the model achieves low RMSE values with the 15-feature subset, indicating strong predictive performance during feature selection. Across the 5-fold cross-validation, the results are consistent, with RMSE, R^2, and MAE values remaining stable across iterations. The low RMSE and MAE confirm the model's accuracy and stability, while R^2 values, ranging from 0.0903284 to 0.1016013, suggest that the model has the potential to explain a reasonable portion of the variance in the target variable. These consistent metrics highlight the robustness of the model and validate the 15-feature subset as a reliable choice for prediction.

```
varImportance <- varImp(rfe_results)
rfe_importance_df <- data.frame(Variable= rownames(varImportance),
                                Overall=varImportance$Overall )

rfe_importance_df |> ggplot( aes(y = reorder(Variable, +Overall), x = Overall)) + geom_bar(stat = "iden
    title = "Variable Importance",
    x = "Overall Score",
    y = "Variable"
  )  + theme_minimal()
```




```
imp_predictors <- predictors(rfe_results)
best_predictors <- append(brand_code_col,imp_predictors[imp_predictors != "brand_code_encoded"])
```

The Recursive Feature Elimination (RFE) results provide a clear ranking of predictors, emphasizing their contributions to the model's performance. The top-ranked variable, brand_code_encoded, underscores the importance of capturing brand-specific patterns through target encoding, as it strongly correlates with the target variable. Among the operational process variables, Mnf.Flow, Oxygen.Filler, Pressure.Vacuum, and Temperature stand out, reflecting their critical role in driving variability in the target. These variables highlight the influence of flow rates, oxygen levels, pressure conditions, and environmental factors in the manufacturing process, which are essential for maintaining product quality.

Additional contributors, such as Usage.cont, Carb.Rel, and Balling.Lvl, showcase the importance of operational and compositional properties in fine-tuning predictions. While variables like Filler.Speed, Carb.Flow, and Balling rank lower, they still provide valuable supplementary information about the operational flow and composition dynamics.

Features such as Density, Bowl.Setpoint, and Filler.Level rank among the lowest, indicating limited direct impact or indirect influence through higher-ranked variables. Similarly, Carb.Pressure1 and Hyd.Pressure3 show minimal importance, suggesting their contribution to the target is captured by other operational metrics.

The RFE results highlight a robust combination of categorical, operational, and compositional variables, with brand_code_encoded and key operational factors leading the rankings. The brand_code_encoded variable, while ranked as the most significant, will be provided to the final model in its one-hot encoded format to ensure compatibility with the Support Vector Machine (SVM) regression model.

5.3 Support Vector Machine Model Setup This section details the implementation and tuning of a Support Vector Machine (SVM) regression model with a Gaussian Radial Basis Function (RBF) kernel to predict pH levels in the manufacturing process. The model is trained on a carefully selected set of predictors, which will be preprocessed using centering and scaling techniques. These preprocessing steps ensure that all variables contribute proportionally to the model and meet the requirements for SVM, which is sensitive to the magnitude of features.

To optimize model performance, hyperparameter tuning was conducted using a systematic grid search approach. This process explored a range of values for two critical parameters: the kernel width (sigma) and the regularization parameter (C). The kernel width (sigma) was varied from 0.01 to 0.2 in increments of 0.01, controlling the locality of the RBF kernel and determining how far its influence extends in the feature space. The regularization parameter (C) was tested over a range of 1 to 5 in increments of 1, balancing the trade-off between minimizing errors on the training data and maintaining a simpler, more generalizable model. Together, these parameters define the flexibility of the regression function and its ability to manage prediction deviations.

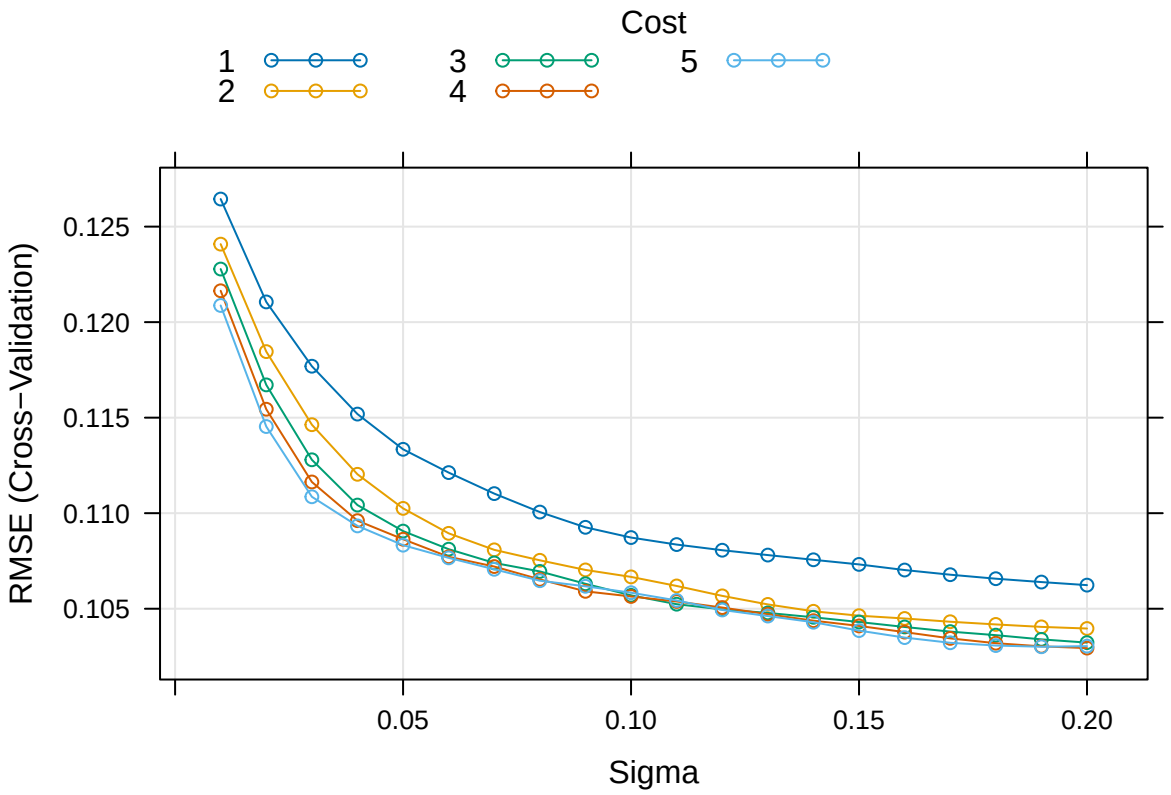
The training process incorporated cross-validation to evaluate model performance and ensure the selected hyperparameters generalize well to unseen data. This approach reduces the risk of overfitting and helps develop a predictive model that is both accurate and robust, meeting regulatory requirements for monitoring and reporting pH variability in the manufacturing process.

```
set.seed(8675309)

svmRTuned <- train(
           x=beverage_man_train |>  select(all_of(best_predictors)),

                y=beverage_man_train$PH,
                   method = "svmRadial",
                   preProcess = c("center", "scale"),
                    tuneGrid = expand.grid(sigma = seq(0.01,0.2,0.01), C = seq(1,5,1)),

                   trControl = trainControl(method = "cv",allowParallel = TRUE))
```

```
svmRTuned$finalModel
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: eps-svr  (regression)
##  parameter : epsilon = 0.1  cost C = 4
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.2
##
## Number of Support Vectors : 1546
##
## Objective Function Value : -1181.85
## Training error : 0.081453
```

```
plot(svmRTuned)
```




```
sigmaParam <- svmRTuned$bestTune[1,"sigma"]
cParam <- svmRTuned$bestTune[1,"C"]

svmResults <- svmRTuned$results |> filter(sigma == sigmaParam & C==cParam)
svmResults |> kable(caption = "SVM Training Set Evaluation Metrics") |> kable_styling() |>  kable_class
```

Table 3: SVM Training Set Evaluation Metrics

sigma	C	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
0.2	4	0.1029398	0.6387648	0.0723842	0.008408	0.0591688	0.0044526

The results of the cross-validated SVM model with a sigma value of 0.2 and a cost parameter (C) of 4 demonstrate robust performance. The model achieves a low RMSE of 0.1029398 and an MAE of 0.0723842, indicating accurate and consistent predictions on the scaled data. The R-squared value of 0.6387648 suggests the model explains approximately 7.2384217 of the variance in the target variable, showing moderate explanatory power. The standard deviations for RMSE (0.008408), R-squared (0.0591688), and MAE (0.0044526) across resampling folds are small, highlighting the stability of the model's performance across different data splits. These metrics indicate that the chosen parameters produce a reliable and well-generalized model for the given training dataset.

Building on these results, it is essential to evaluate the model's consistency across training and test datasets to ensure its robustness and generalizability. By comparing cross-validation metrics with test set performance, we can assess whether the SVM model maintains its predictive accuracy and explanatory power when applied to unseen data.

```r
svmPred <- predict(svmRTuned,newdata = beverage_man_test |> select(all_of(best_predictors)))

svm_test_post <- postResample(pred = svmPred, obs = beverage_man_test$PH) |> as.data.frame()

svm_test_metrics <- data.frame(RMSE=svm_test_post[1,1],Rsquared=svm_test_post[2,1],MAE=svm_test_post[3,

svm_test_metrics |> kable(caption = "SVM Test Set Evaluation Metrics") |> kable_styling() |>  kable_cla
```

Table 4: SVM Test Set Evaluation Metrics

RMSE	Rsquared	MAE
0.1095147	0.6143043	0.0772482

The SVM model demonstrates consistent performance between the training and test datasets, as indicated by the metrics from cross-validation and post-resample evaluation. The RMSE during training, (0.1029398), is nearly identical to the test RMSE, (0.1029398), suggesting stable predictive accuracy. Similarly, the MAE values are close, with (0.0723842) in training and (0.0723842) on the test set, reflecting consistent error magnitudes. The R-squared value shows a minor decrease from (0.6387648) in training to (0.6387648) on the test set, indicating that the model maintains reasonable explanatory power without significant overfitting. These results suggest that the model generalizes well and is reliable for making predictions on unseen data.

Overall, the SVM model demonstrates moderate predictive performance, with consistent metrics across training and test datasets that highlight its robustness and reliability. While the RMSE and MAE values indicate good predictive accuracy, the R-squared suggests room for improvement in explaining the variance of the target variable. These results establish a solid benchmark for comparison with other models.

6. Model Selection

7. Conclusion

8. Annotated References

1. **Anton Paar Wiki. (n.d.).** *Carbon Dioxide in Beverages.*
   Carbon Dioxide in Beverages
   This resource provides an in-depth understanding of the role of carbon dioxide in beverages, including its effect on carbonation, fizziness, and product quality. It directly supports our analysis of variables like **Carb Flow** and **Carb Pressure**, which influence carbonation and pH levels.

2. **Omega. (n.d.).** *What is pH?*
   What is pH?
   This article explains the concept of pH, its measurement, and its relevance to various industries. It is useful for understanding the chemical principles behind pH variability in beverages and helps contextualize our target variable within the manufacturing process.

3. **Emerson. (n.d.).** *Training Beverage Process Solutions Guide on De-Aeration.*
   Training Beverage Process Solutions Guide on De-Aeration
   This document focuses on de-aeration processes in beverage production, particularly the removal of dissolved oxygen and its impact on carbonation and quality. It directly relates to variables like **Oxygen.Filler** and **Pressure.Vacuum**, providing insights into their operational importance.

4. **Jochamp. (n.d.).** *Carbonated Beverages Manufacturing Process - A Step by Step Guide.*
   Carbonated Beverages Manufacturing Process - A Step by Step Guide

This guide offers a comprehensive overview of the carbonation process in beverage production, including the role of temperature, pressure, and filling speed. It supports our understanding of variables like **Temperature**, **Filler.Speed**, and **Carb Pressure**, helping us interpret their influence on product quality.