Final Project

Lewris Mota

2024-11-27

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

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

Support Vector Machine with Gaussian Radial Basis Function Kernel

This section outlines the training and tuning of a Support Vector Machine (SVM) regression model 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.

Support Vector Machine Model Preprocessing

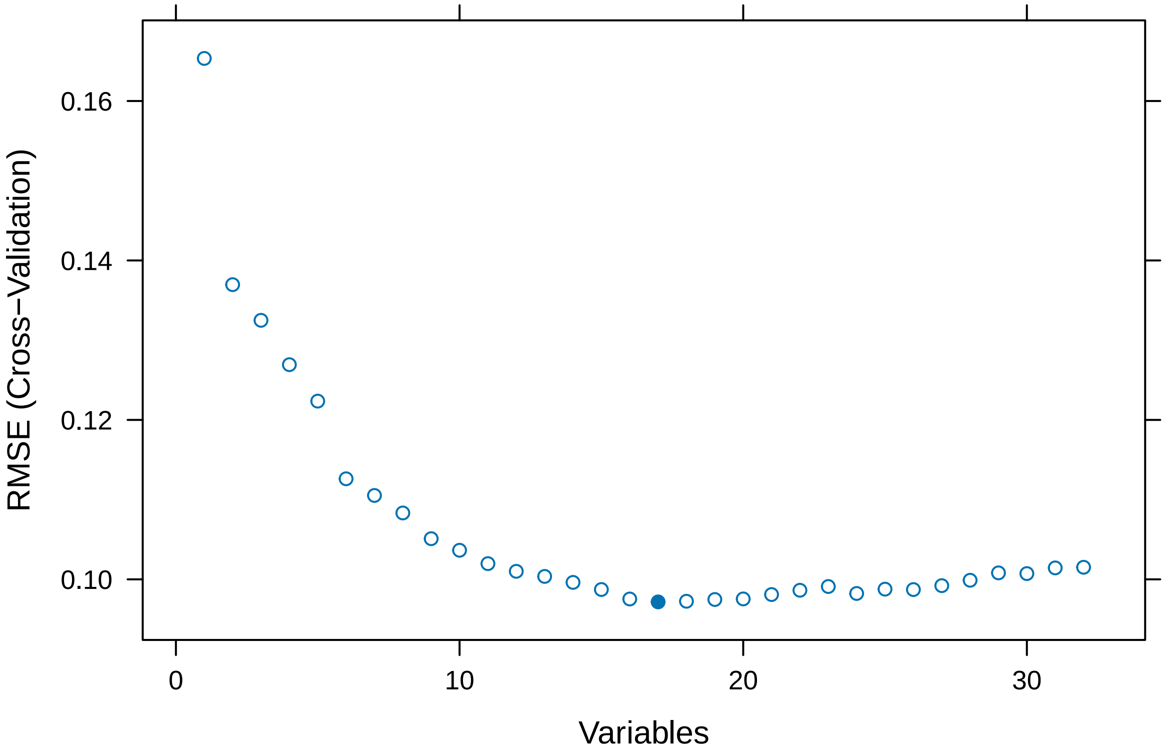
The dataset is split into training (75%) and testing (25%) subsets using random sampling to ensure the model is trained on a representative and diverse portion of the data while reserving an independent set for unbiased performance validation. Feature selection is applied prior to fitting a Radial Support 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, ]
```

```r
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"))
```

```r
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
)
```

```r
plot(rfe_results)
```



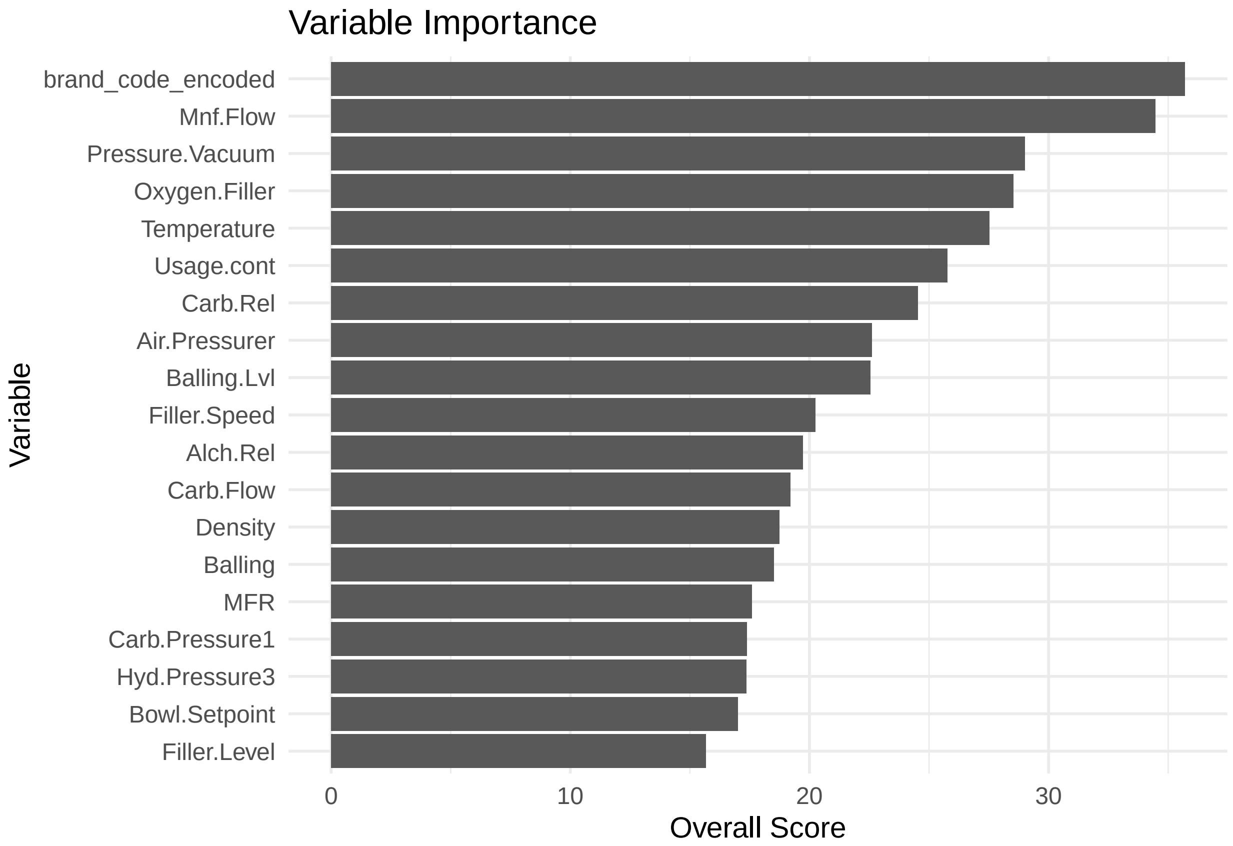
```
rfeMaxR2 <- max(rfe_results$resample[,"RMSE"])
rfeMinR2 <- min(rfe_results$resample[,"RMSE"])

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

Table 1: Recursive Feature Elimination Results

	Variables	RMSE	Rsquared	MAE	Resample
17	17	0.1109911	0.6013809	0.0750401	Fold1
49	17	0.0977572	0.7132663	0.0705884	Fold2
81	17	0.0934207	0.7290961	0.0687313	Fold3
113	17	0.0873011	0.7266080	0.0661555	Fold4
145	17	0.0963918	0.7350628	0.0704511	Fold5

The plot generated from the Recursive Feature Elimination (RFE) process shows that the model achieves low RMSE values with the 17-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 0.0873011 to 0.1109911, 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 17-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.

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.

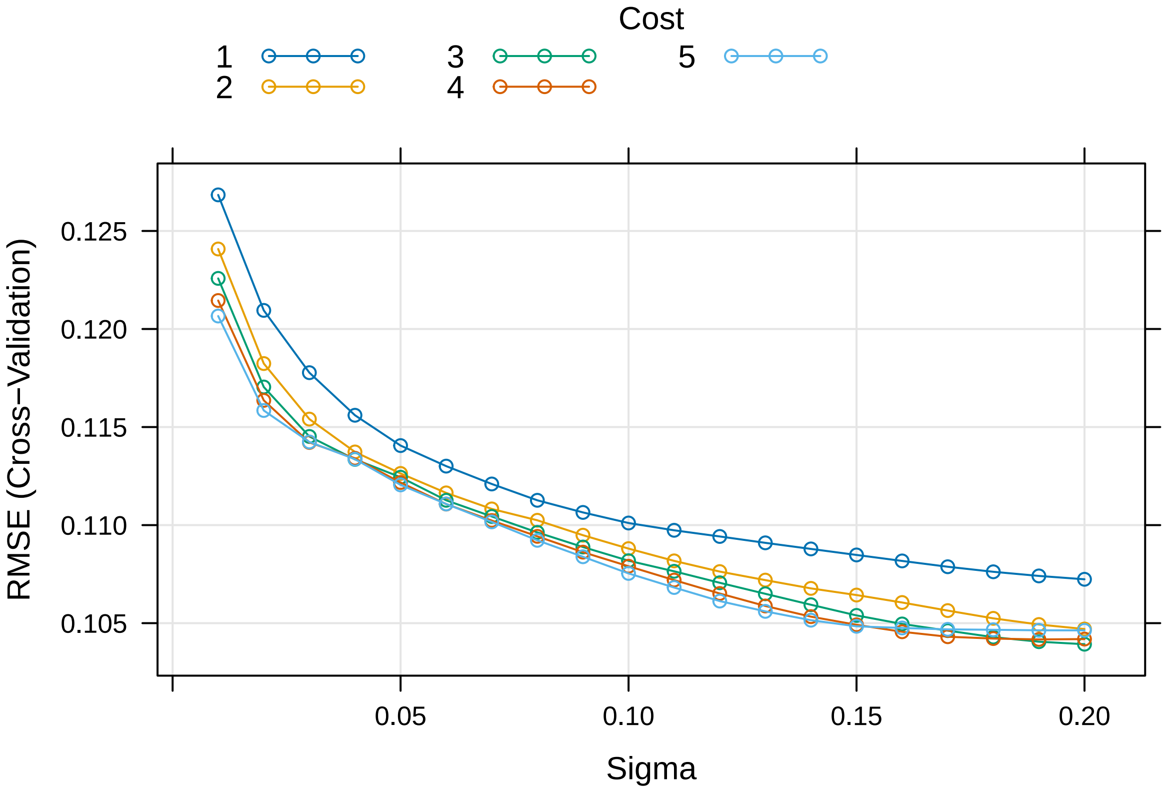
```r
set.seed(8675309)

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))
```

```r
svmRTuned$finalModel
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: eps-svr  (regression)
##  parameter : epsilon = 0.1  cost C = 3
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.2
##
```

```
## Number of Support Vectors : 1496
##
## Objective Function Value : -932.4143
## Training error : 0.098148
```

```
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 2: SVM Training Set Evaluation Metrics

sigma	C	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
0.2	3	0.1039269	0.6380536	0.0730553	0.0122489	0.0809546	0.006001

The results of the cross-validated SVM model with a sigma value of 0.2 and a cost parameter (C) of 3 demonstrate robust performance. The model achieves a low RMSE of 0.1039269 and an MAE of 0.0730553, indicating accurate and consistent predictions on the scaled data. The R-squared value of 0.6380536 suggests the model explains approximately 7.3055277 of the variance in the target variable, showing moderate explanatory power. The standard deviations for RMSE (0.0122489), R-squared (0.0809546), and MAE (0.006001) 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.

```
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 3: SVM Test Set Evaluation Metrics

RMSE	Rsquared	MAE
0.1034722	0.6297776	0.0729414

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, (0.1039269), is nearly identical to the test RMSE, (0.1039269), suggesting stable predictive accuracy. Similarly, the MAE values are close, with (0.0730553) in training and (0.0730553) on the test set, reflecting consistent error magnitudes. The R-squared value shows a minor decrease from (0.6380536) in training to (0.6380536) 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.