Final Project

Lewris Mota

2024-11-27

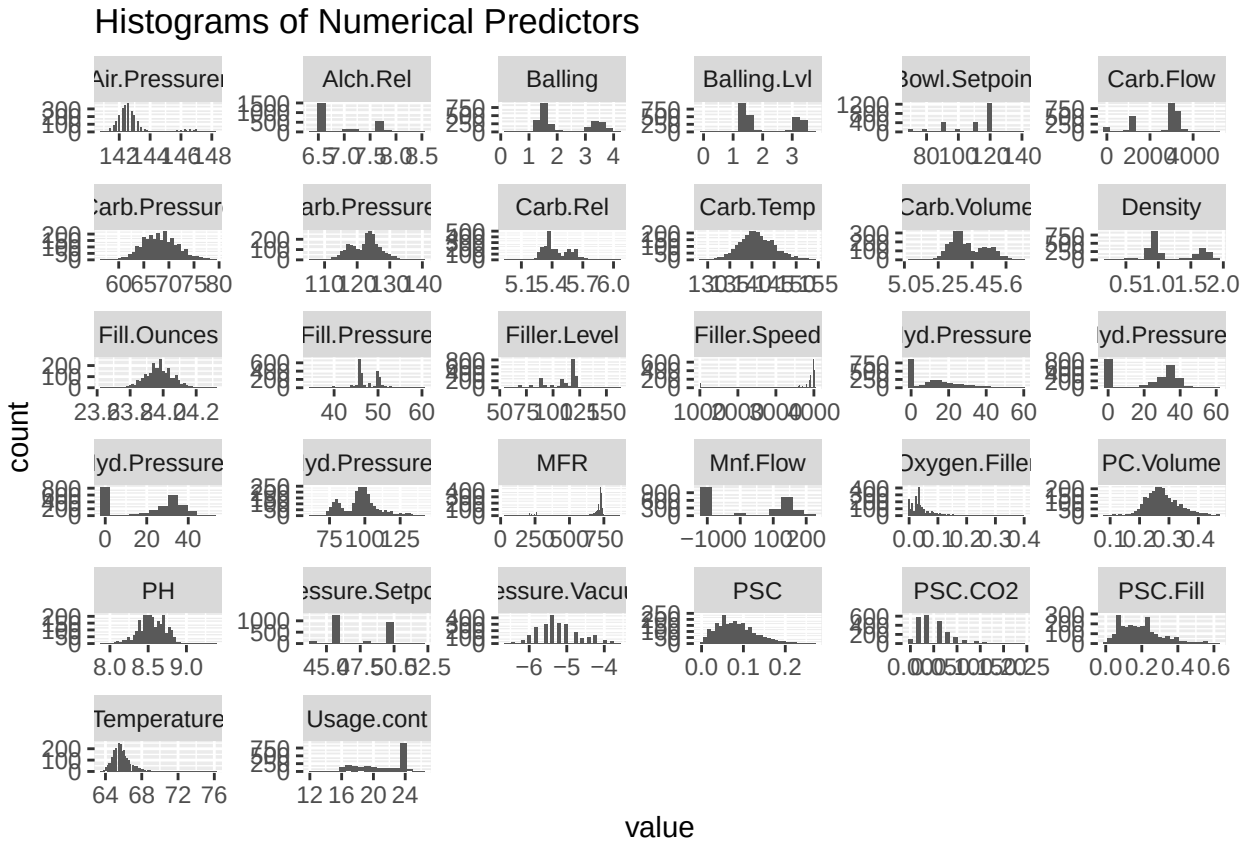
```
manufacturing_tr <- read_csv("imputed_test_data.csv",show_col_types=F)
# manufacturing_test <- read_xlsx("StudentEvaluation.xlsx")

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

Data Preparation The preprocessing process begins by identifying features with near-zero variance, which provide little predictive value, and excluding them to streamline the dataset. Next, numeric features are selected, excluding categorical variables, to calculate a correlation matrix. Features with high correlations above a 0.75 threshold are identified, as they may introduce multicollinearity and reduce model stability. These highly correlated features are then removed to create a refined dataset with reduced dimensionality and improved suitability for modeling. This approach ensures that the remaining features are informative and contribute to robust model performance.

The following predictors will be excluded:

Features Visuals

```
manufacturing_tr |> select(-all_of(brand_code_col)) |>
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
  geom_histogram(binwidth = bins_cal) +
  facet_wrap(~key, scales = "free") +
  ggtitle("Histograms of Numerical Predictors")
```




```
manufacturing_tr  |> select(-all_of(brand_code_col)) |>
  keep(is.numeric) |>
  gather() |>
  ggplot(aes(value)) +
  geom_boxplot(outlier.colour="red", outlier.shape=8,outlier.size=4) +
  facet_wrap(~key, scales = 'free',  ncol = 6) +
  ggtitle("Boxplots of Numerical Predictors")
```

```
manufacturing_tr <- beverage_train_uncorr |> select(-all_of(brand_code_col)) |> keep(is.numeric) |> piv

ggplot( beverage_pivot, aes(x = Value, y = PH)) +

  geom_point() +
  facet_wrap(~Variable, scales = "free_x",ncol = 3) +
  theme_minimal()
```

Data Train/Test Split The dataset with no inter-correlated features was split into training (75%) and testing (25%) subsets using random sampling. This approach ensures that the model is trained on a diverse and representative subset of the data while reserving an independent set for validating its performance.

```
set.seed(8675309)

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

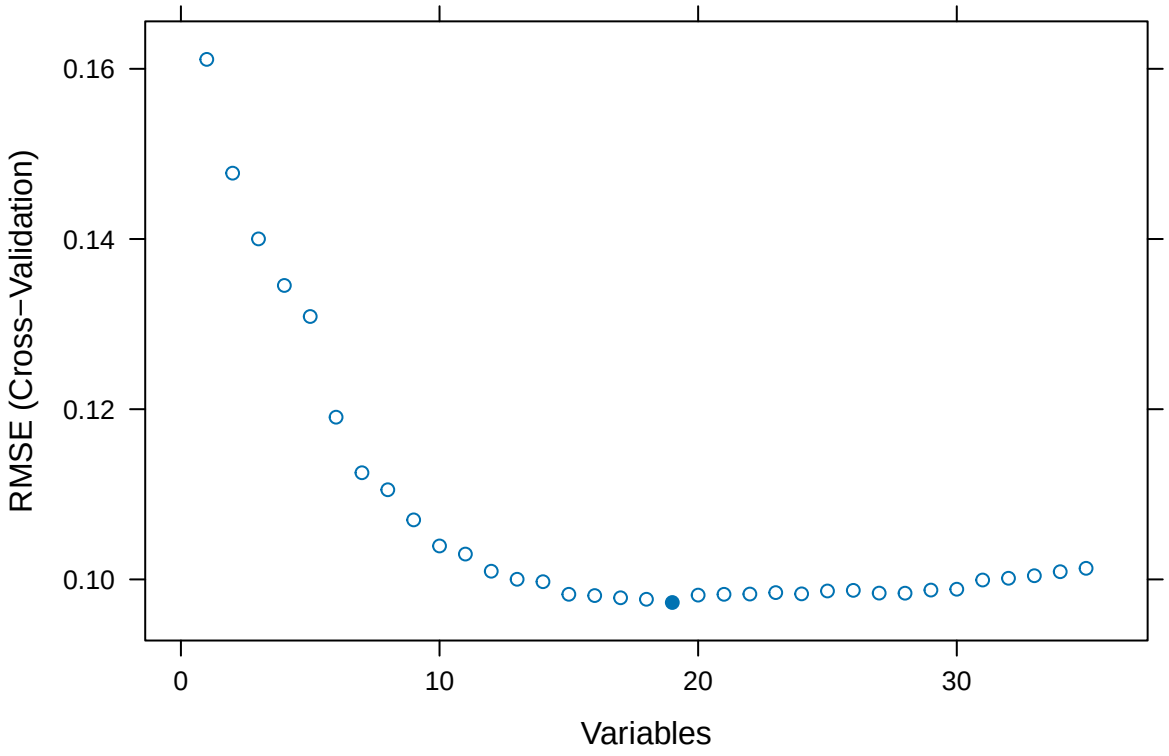
```
beverage_man_train <- manufacturing_tr[trainIndex, ]
beverage_man_test <- manufacturing_tr[-trainIndex, ]
```

SVM Model Preprocessing Before fitting a Radial Support Vector Machine (SVM) model, it is essential to focus on the most relevant features to enhance model performance and efficiency. Recursive Feature Elimination (RFE) is a robust feature selection method that systematically identifies these features. The process involves training a model with all features, ranking their importance, and iteratively removing the least impactful ones until the optimal subset is found. This ensures that the model is not influenced by redundant or irrelevant predictors.

For this model, a Random Forest (rfFuncs) will be used to evaluate feature importance and the model performance will be assessed using 5-fold cross-validation.

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

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

# Perform RFE
rfe_results <- rfe(
    # beverage_man_train |> select(-all_of(c("PH","MFR","Filler.Speed","Oxygen.Filler"))),
    beverage_man_train |> select(-all_of(c("PH"))),
  beverage_man_train$PH,
  sizes = c(1:length(beverage_man_train)),  # Test different subset sizes
  rfeControl = control
)
```

```
plot(rfe_results)
```



The plot of variables against RMSE shows that the model achieves a low RMSE values with the 19-feature subset.

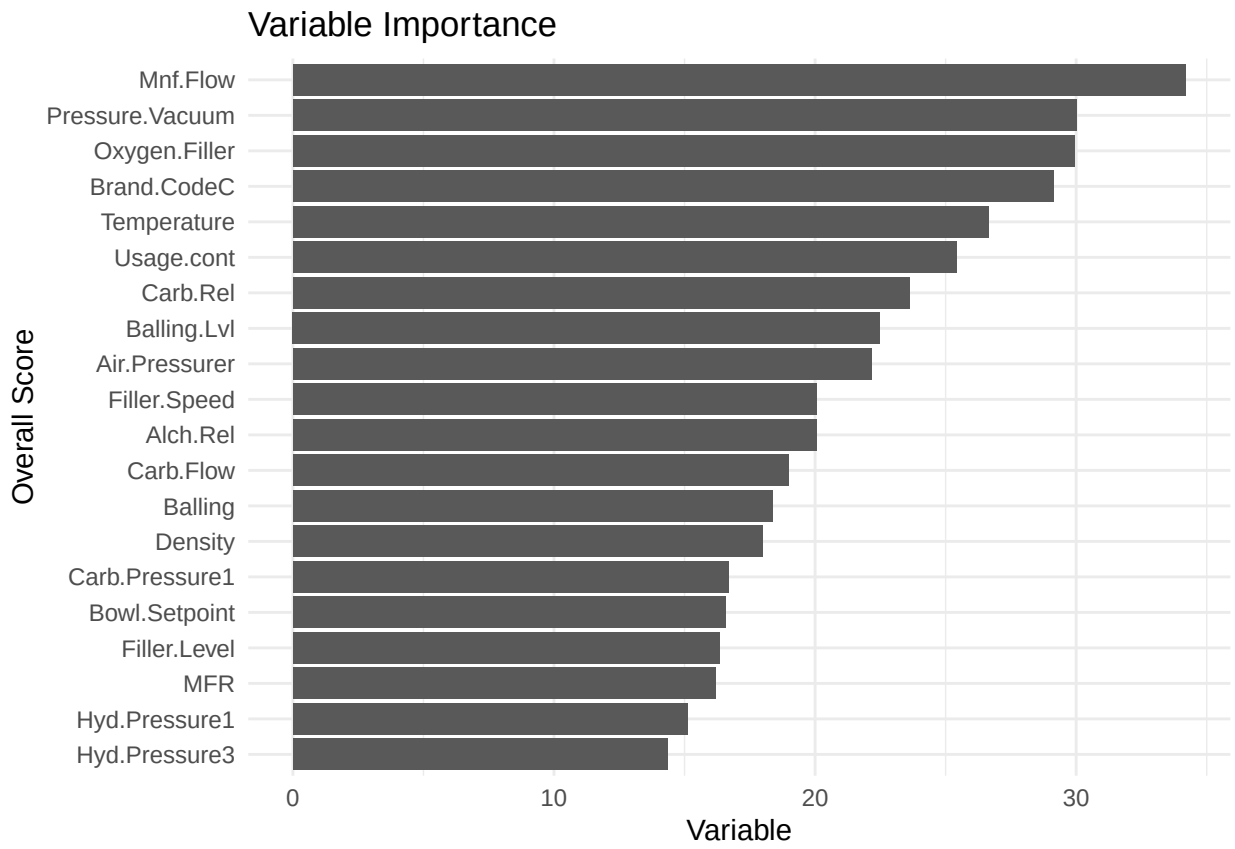
```
rfe_results$resample |> kable() |> kable_styling() |>  kable_classic()
```

	Variables	RMSE	Rsquared	MAE	Resample
19	19	0.1105660	0.6053546	0.0744996	Fold1
54	19	0.0977113	0.7124464	0.0703980	Fold2
89	19	0.0938515	0.7288807	0.0695308	Fold3
124	19	0.0883916	0.7195909	0.0672496	Fold4
159	19	0.0959100	0.7393462	0.0700071	Fold5

The results are consistent across the 5 folds, with similar values for RMSE, R^2, and MAE in each iteration. The low RMSE and MAE values indicate that the model is making accurate and stable predictions. The R^2 values, ranging from 0.54 to 0.64, suggest that the model explains a reasonable portion of the variance in the target variable. These consistent performance metrics across folds indicate that the model is robust and generalizes well, making the feature subset of 19 variables a reliable choice for prediction. Overall, the model appears to perform effectively with this subset, supporting its use for further analysis.

```
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 = "Variable",
    y = "Overall Score"
  )  + theme_minimal()
```




```
best_predictors <- predictors(rfe_results)
```

SVM with Gaussian Radial Basis kernel function This section focuses on implementing and tuning a Support Vector Machine (SVM) model with a Gaussian Radial Basis Function (RBF) kernel. The model is trained on the selected predictors, with the target variable being the PH levels. To enhance model performance, preprocessing steps such as centering, scaling, and spatial sign transformation are applied. Hyperparameter tuning is conducted using a grid search approach, exploring a range of values for the kernel width (sigma) and cost parameter (C). The training process is guided by cross-validation to ensure robust evaluation and optimal parameter selection. This methodology aims to develop a highly accurate and generalizable model for predicting the target variable.

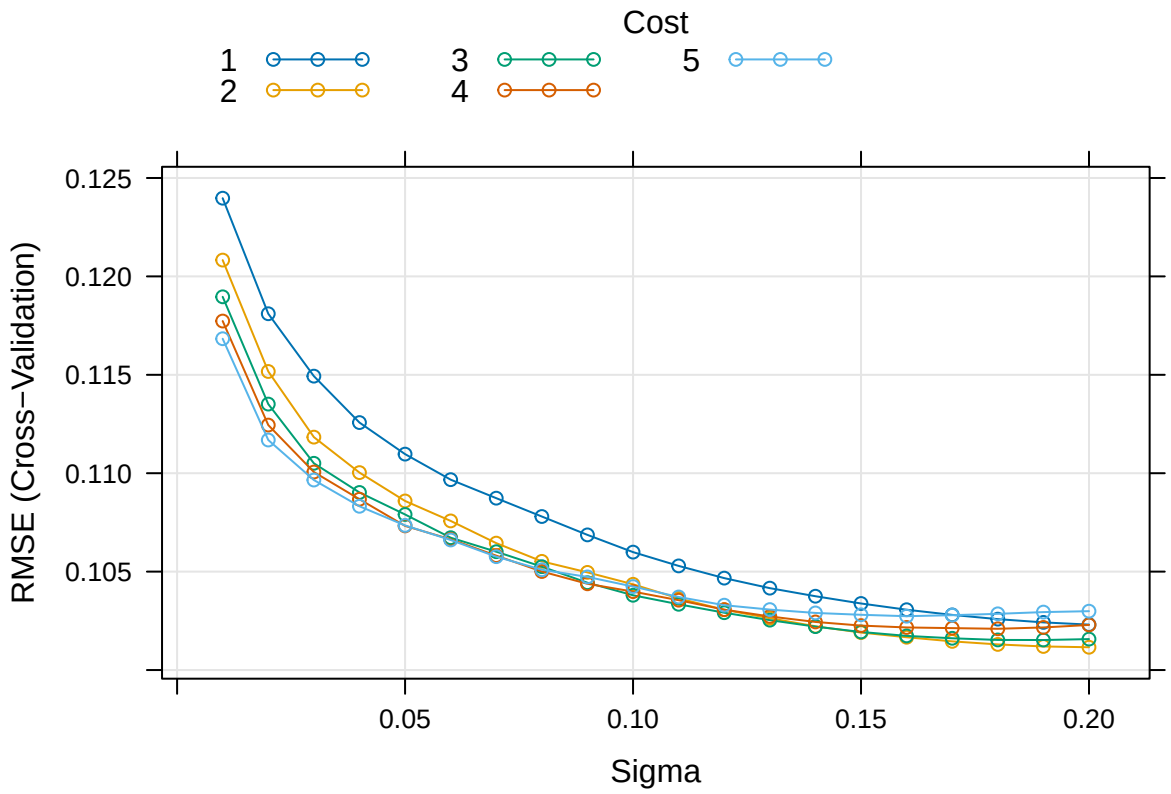
```
set.seed(8675309)

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

                y=beverage_man_train$PH,
                   method = "svmRadial",
                   preProcess = c("center", "scale","spatialSign"),
                    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 = 2
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.2
##
## Number of Support Vectors : 1471
##
## Objective Function Value : -663.7952
## Training error : 0.103177
```

```
plot(svmRTuned)
```




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

svmResults <- svmRTuned$results |> filter(sigma == sigmaParam & C==cParam)
svmResults |> kable() |> kable_styling() |>  kable_classic()
```

sigma	C	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
0.2	2	0.1011584	0.6557155	0.0711389	0.010064	0.0741996	0.005035

The results of the cross-validated SVM model with a sigma value of 0.2 and a cost parameter (C) of 2 demonstrate robust performance. The model achieves a low RMSE of 0.1011584 and an MAE of 0.0711389, indicating accurate and consistent predictions on the scaled data. The R-squared value of 0.6557155 suggests the model explains approximately 7.1138871 of the variance in the target variable, showing moderate explanatory power. The standard deviations for RMSE (0.010064), R-squared (0.0741996), and MAE (0.005035) 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.

```
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() |> kable_styling() |>  kable_classic()
```

RMSE	Rsquared	MAE
0.1013199	0.641751	0.0711864

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, (0.1011584), is nearly identical to the test RMSE, (0.1011584), suggesting stable predictive accuracy. Similarly, the MAE values are close, with (0.0711389) in training and (0.0711389) on the test set, reflecting consistent error magnitudes. The R-squared value shows a minor decrease from (0.6557155) in training to (0.6557155) 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.

The metrics from this model will be used as a benchmark for comparison with other models to identify the best-performing approach for the dataset.