Report
You did a very good job. This is a very professional report, an executive report. Little things make the difference. ToC was good, could be better. Some groups offered rich references which added credibility to their work. You had some easily correctable things that you did not tale the time to correct, like things pointed out by spell check.

You can argue and it might seem like small things, but small things make the difference. Right or wrong, perception of your work counts more in the data science profession than anything. Yes, you must know your stuff, that is a given. It does not differentiate you. And most times you are dealing with people that cannot accurately gauge your technical acumen (moreover, in the real world, actuals are difficult to account for – economic or extraneous variation OR was it your modeling aptitude). This was meant to be a report to executive leaders and you did a very nice job.

Technical
Good model selection and comparison that produced great results.
MAPE Accuracy was 0.82 great, best submitted was 0.79.

Report Grade – 96 (80% of grade)

MAPE Grade 98 (20% of grade)

Final Grade – 96.4 !!

Data 624 Predictive Analytics
Project #2
Group 1: Keith Colella, Bridget Boakye, Daniel Craig

TABLE OF CONTENTS

TABLE OF CONTENTS 1

EXECUTIVE SUMMARY **1**
INTRODUCTION **2**
EXPLORATORY DATA ANALYSIS (EDA) **3**
Primary Observations 7
MODELING **8**
VARIABLE IMPORTANCE **10**
Strong Variables plotted Against PH Values: 11
CONCLUSION **12**
APPENDIX A: Python Comparison **13**
APPENDIX B: Links to Implementation **14**
ADDITIONAL PLOTS **15**

EXECUTIVE SUMMARY

The ability to turn data, and in this case, coefficients into insights and then profit is what every business desires. In this project, we analyze the dataset of a beverage manufacturing company, BMC, to predict one of its key performance indicators, pH. Accurate prediction of pH values is essential for maintaining the desired range of acidity/alkalinity. After robust modeling of the data provided using the most up-to-date techniques available, we identify that the RandomForest (RF) model provides the best predictions of PH. The analysis modeled across many different techniques to derive knowledge about different variables to provide insight on potential variables for change. Variables across models were broken into three groups; Strong/Common, Strong/Uncommon, and Weak/Common. These groups were chosen due to their relevance in potential cost savings or product quality improvement by optimizing their values with potential actions for each group. The Strong/Common variables in Oxy Filler, Hyd Pressure 1 - 3, Bowl Setpoint, Filler level, and Mnf Flow that have the strongest abilities to control PH outcome and are the most likely to provide reliable and significant changes to PH. Strong/Uncommon variables are potentially as powerful as the Strong/Common variables, but less reliable. Weak/Common variables are reliable, but not as powerful in impacting PH. Optimizing their values as a group may show significant results, but should be considered mainly as a cost savings approach in replacement of other variables. For instance, if Mnf Flow is hard to control or costly to implement, reducing cost here to reinvest in another more easily controllable variable may provide cost savings with Im equitable product quality. For more details, please reference the Variable Importance section. This insight should help BMC continue to make beverages that satisfy customers and cements its market position.

INTRODUCTION

The dataset provided to us by the client contains 2,571 observations and 33 variables. pH is a critical measure of acidity/alkalinity and serves as a key performance indicator (KPI) for the company. Accurate prediction of pH values is essential to maintain the desired range of acidity/alkalinity.

We assigned the data to three analysts. One performed the exploratory data analysis and the others followed a similar protocol to fit models and evaluate the results. This final deliverable consists of an evaluation of several models and predictions for pH.

The general protocol for predicting pH was as follows:

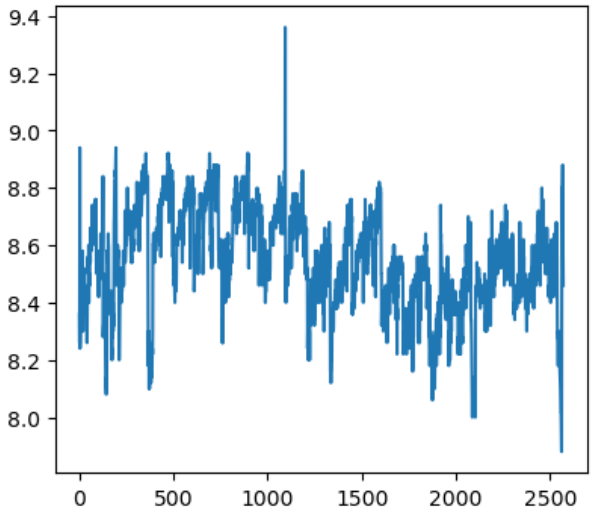
1. Explore the data to understand the variables and their relationship to the target variable (pH) and each other. In this step, we consider issues such as missing values, skewness, collinearity, and multicollinearity.
2. Preprocess the data to ensure that the assumptions for models are observed. As part of preprocessing, we split the data 70/30 into training and testing sets, so that we have out-of-sample data to assess model performance. Transformations were performed as needed.
3. Models were tuned across a grid of options and a final tuning chosen to be compared against other models.
4. Models were tested against a test set and a final model selected by comparing MAPE and R^2 values.
5. Predictions were generated using several models for comparison, but only one submitted for evaluation.

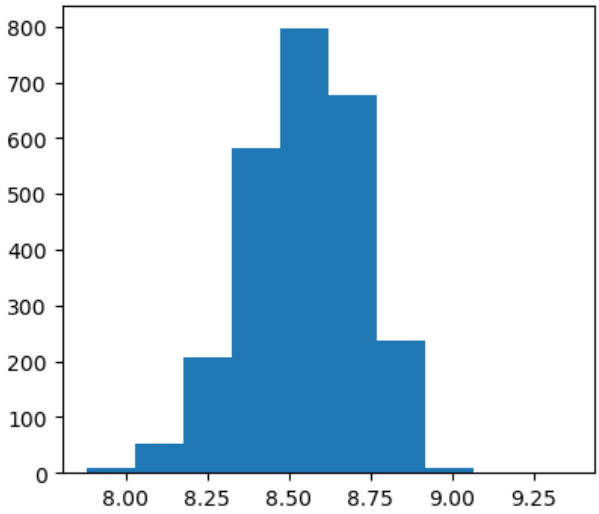
EXPLORATORY DATA ANALYSIS (EDA)

We summarize EDA, along with support visuals, below.

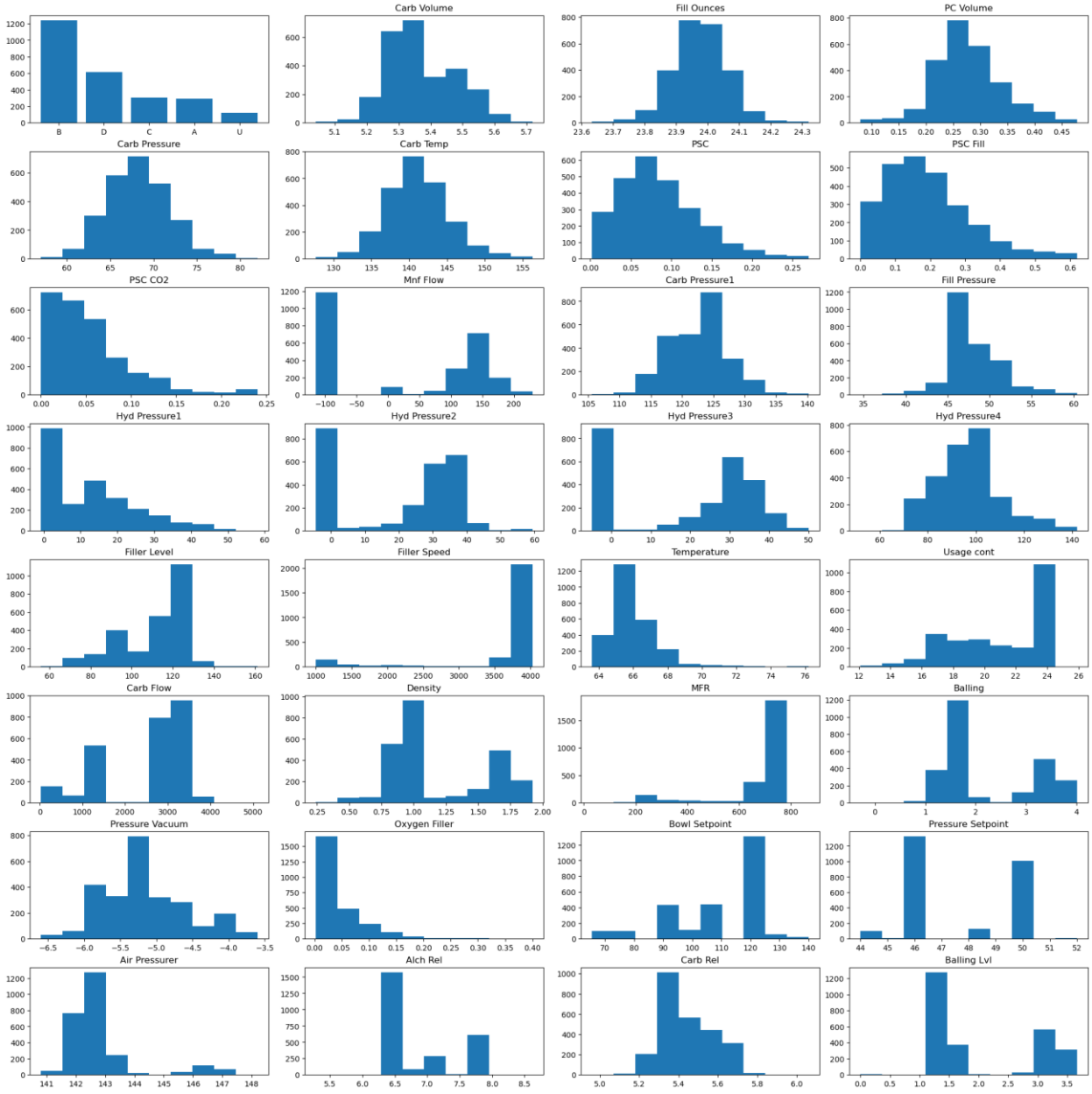
1. Identified and handled missing values in the dataset using Multiple Imputation by Chained Equations (MICE), a standard yet robust approach for handling missing values. MICE used Predictive Mean Matching for numeric values and Logistic Regression to impute categorical variables. MICE attempts to use complete observations to guide the selection of new values. Its ability to handle categorical variables led it to be chosen over KNN Imputation.

2. Explored the target variable (pH) using summary statistics and histograms.

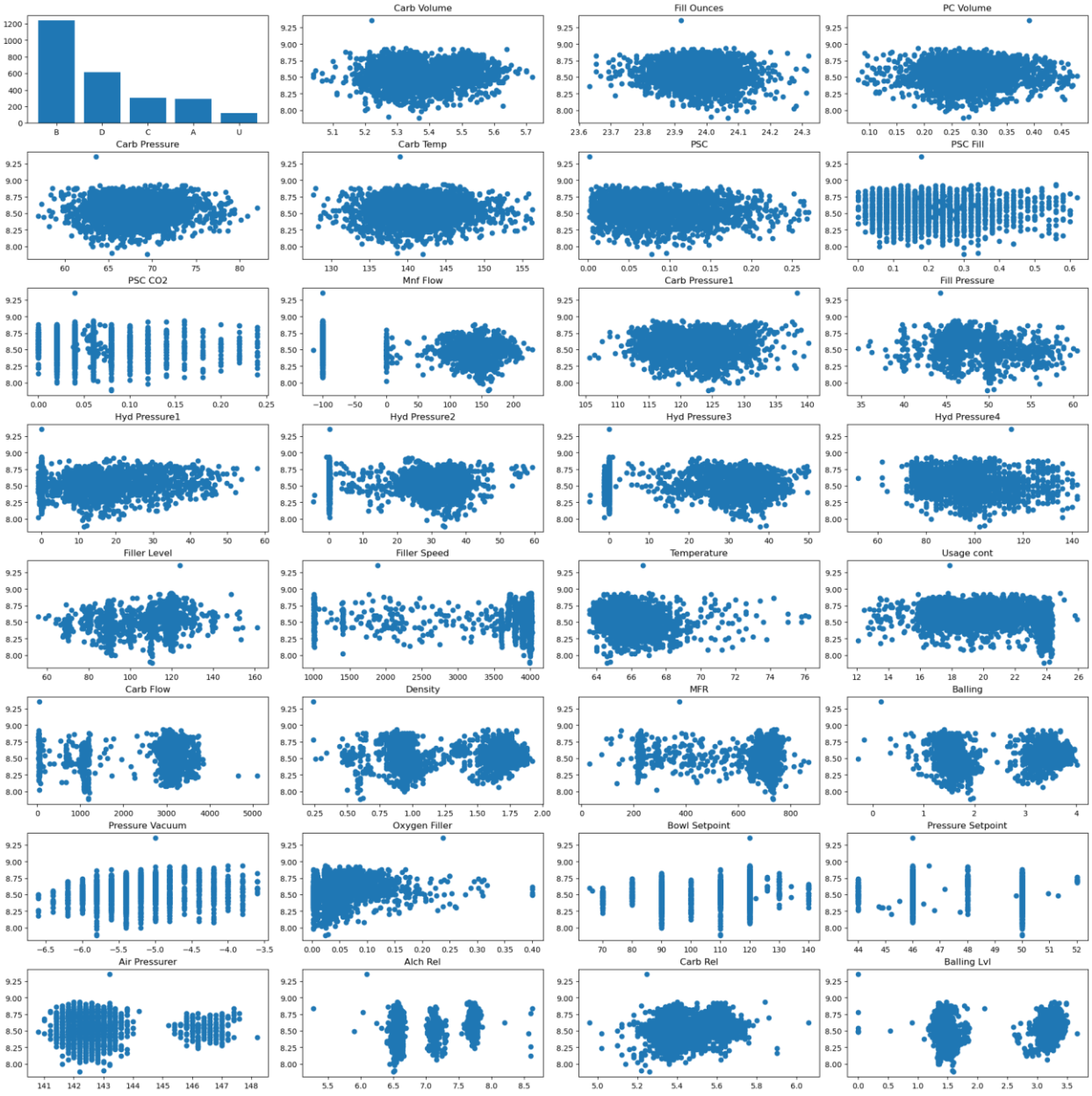




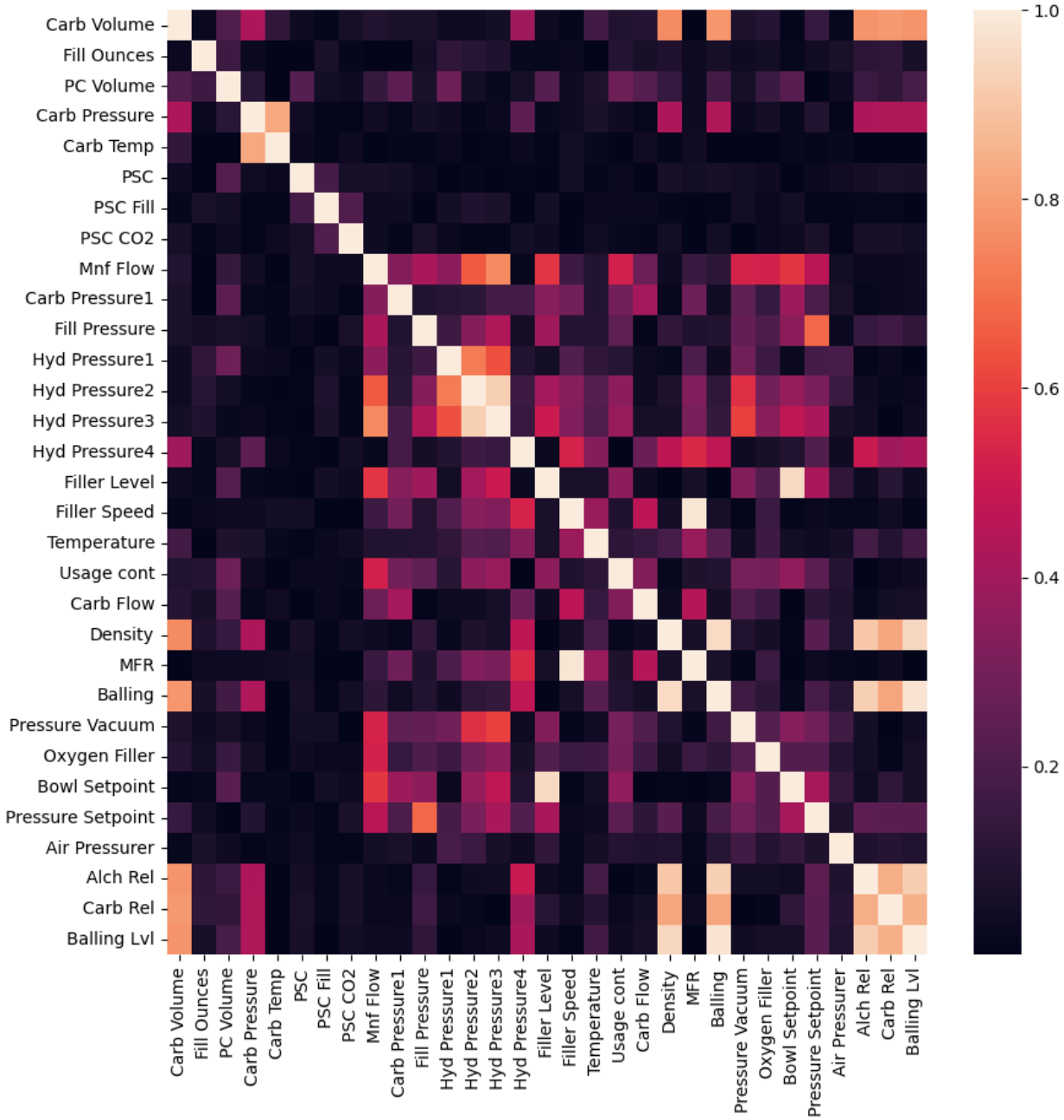
3. Explored the distribution of predictors. This included the only categorical predictor variable, Brand Code. Many variables had skewed distributions that would later be transformed.



4. Analyzed the relationships between the target variable (pH) and other numeric variables using correlation matrices. We looked for strong correlations or patterns that might indicate predictive power.



5. Examine correlation between predictor variables. Where high correlation exists, we consider removal of variables that may have collinearity.



Primary Observations

1. Target variable: the summary function for pH suggests a distribution of pH values centered around 8.5 (given mean and median are nearly equal) with some variation on both ends, but no extreme outliers. The histogram of the variable suggests the distribution of pH is nearly normal, with a slight left skew, most values towards the right, the more alkaline end.

2. Categorical variable: It was clear from the analysis of the categorical variable, Brand Code, that most products fall in Brand Code, B (+1200), followed by D (+600). This could suggest there is a potential correlation between Brand B and PH. Moreover, given the predominance of one category over the others, we'd need to ensure categories are represented proportionally during cross-validation.

3. Numeric variable: the analysis of the numerical variables shows that there is a correlation between the variables and pH. We set a threshold of 0.9 and remove data with pairwise correlations above that threshold to ensure a more robust model. Our remaining variables are 23 as opposed to 32. Plots of variables with high predictive potential are below:

Overall, the EDA performed on the dataset provided valuable insights into the dataset. Issues such as missing data and multicollinearity are addressed. Further preprocessing will be done in the modeling to align with the technique being used.

MODELING

We assessed a broad range of models to assess various approaches for predicting pH. The table below details all models considered, along with some key details. Namely, the table details two key error metrics, and indicates whether the model requires transformation of input predictor variables and whether the model requires tuning of hyperparameters (all discussed below in more detail). **A RandomForest model of 500 trees testing 28 splits was chosen as the final model; it had the highest R-Squared value and tied for the lowest MAPE.**

	Model	Transformations	Tuning Required	Test MAPE	Test Rsq
Linear	Ordinary Least Squares	1,2,3	N	0.012	0.405
	Partial Least Squares	1	N	0.012	0.390
	Ridge Regression	1,2	Y	0.009	0.383
	LASSO	1,2	Y	0.009	0.325
Non-Linear	MARS	0	N	0.011	0.523
	Cubist	0	N	0.008	0.699
Tree-based	Single Tree	0	Y	0.009	0.589
	Bagged Trees	0	Y	0.009	0.591
	Random Forest	0	Y	0.008	0.708
	Boosted Trees	0	Y	0.009	0.623
ML	Neural Networks	1,2,3	Y	0.011	0.509
	Support Vector Machines	1	Y	0.009	0.572
	K-Nearest Neighbors	1,2	Y	0.011	0.508

0 No transformation
1 Scale and Center
2 Remove Near-Zero Variance Predictors
3 Remove Highly Correlated Predictors

A broad summary of the model categories from the approaches highlighted below.

1. Linear: Models our target variable as a linear combination of predictor variables. Many "traditional" statistical models fall into this category. Linear models provide more easily interpretable results, but it may not forecast as accurately as more complex models.
2. Non-Linear: Models our target variable based on non-linear (i.e. curved) relationships between variables. These models may provide more accurate predictions, but they are typically more difficult to interpret than linear models.
3. Tree-based: Models our target variable using one or more decision trees. These trees use a series of if-then rules to generate predictions using the values of predictor variables. While single tree models are relatively easy to interpret, more complex tree models (such as Random Forest) use an ensemble approach that aggregates results across many trees. Such approaches may enhance predictive accuracy at the cost of interpretability.
4. Other Machine Learning (ML): We employ two other ML model types: Support Vector Machines (SVM) and Neural Networks (NN). SVMs are technically linear models,

whereas NNs constitute their one class of modeling. Both can produce highly accurate predictions, but they can be difficult to interpret.

With all models, we followed a similar protocol. First, we wished to create models robust to changing trends in the data, so we used 70% of the data to train models and the remaining 30% to evaluate their predictive capabilities.

Next, we implement key transformations for predictor variables, as detailed in the summary table above. These transformations include centering data on zero and scaling so that they are all represented in comparable units. Centering and scaling aim to prevent variables with very large values from "overshadowing" variables on smaller scale during model training (for example, consider Carb Flow, which ranges from ~0 to ~4000, and PSC, which ranges only from ~0 to ~0.3). For some models, we may also remove variables that have near-zero variance (i.e. they primarily take on only a single value) and variables that are highly correlated with each other. Both of these conditions may negatively impact model interpretability and performance.

After implementing any required transformations, we aim to tune hyperparameters. Many of the models we consider (especially more complicated parameters) require us to manually set certain parameters, and model results can be sensitive to these values. For example, tree-based models often require a "max depth" parameter, which dictates the maximum number of levels considered for a tree. To identify optimal parameters, we use cross-validation, which takes various sub-samples of the data, fits models with various hyperparameters, then compares the accuracy of predictions resulting from each fit. The set of hyperparameters that produces the most accurate predictions across sub-samples then provides our optimal set for usage in final model assessment.

With our final set of our tuned models calibrated on the 70% training data, we then generate predictions using the remaining 30% test data to assess performance. We focus on two key error metrics: (i) R-squared, which indicates how well the model explains the response variable, and (ii) Mean Absolute Percentage Error (MAPE), which measures the accuracy of the model's predictions out-of-sample.

VARIABLE IMPORTANCE

Most models performed well, particularly Random Forest and Cubist. Most models placed Mnf Flow as the single most important variable. There also tended to be a choice between Mnf Flow or Oxy Filler receiving significant weight. If Mnf Flow took heavy weight, Oxy Filler was reduced. There may be some strong, yet not crippling, correlation between the two that could be investigated.

This section is meant to serve as a basic guide to the components that played roles in controlling PH. A discussion with engineers for a more thorough optimization of controls for PH

is recommended before testing or changes to production is committed. Please contact one of the analysts listed at the top of the report.

Strong and Common Variables:

1. Mnf Flow + Oxy Filler
2. Hydr Pressure 1 - 3
3. Bowl Setpoint
4. Filler Level
5. Pressure Setpoint
6. Usage Cont

Strong and Uncommon Variables:

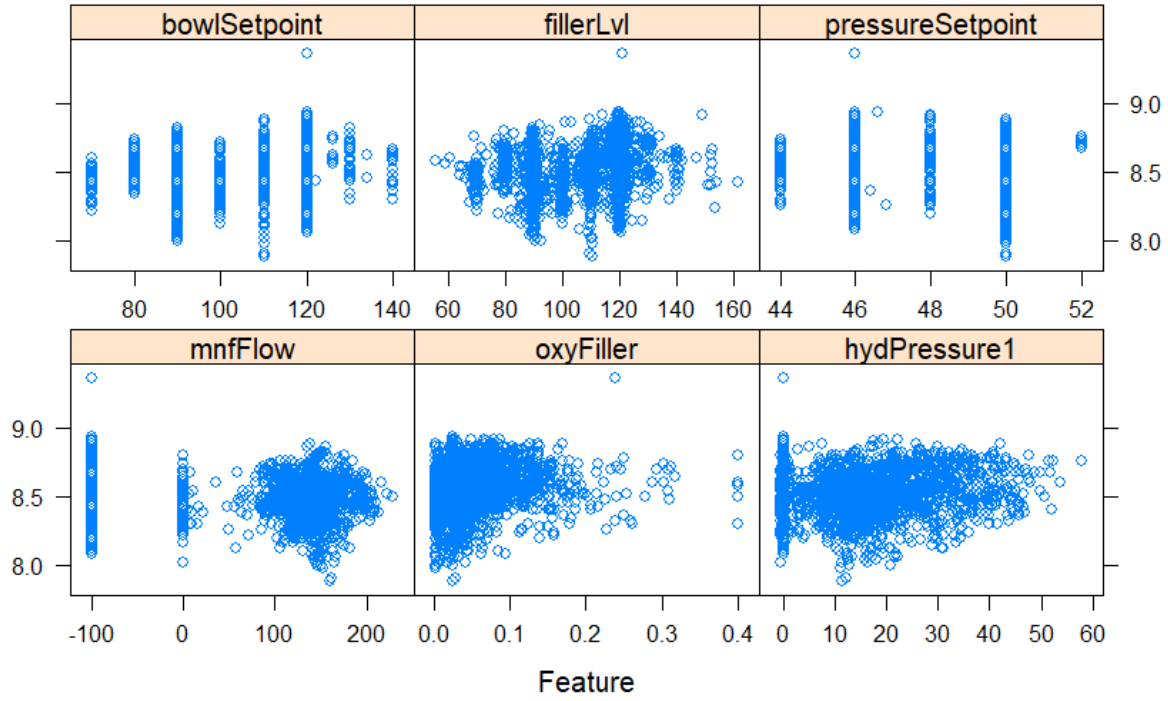
1. Brand Codes C
2. Filler Speed
3. Pressure Vacuum

Weak and Common Variables:

1. Alch Rel
2. Carb Rel
3. Temperature
4. Hydr Pressure 4
5. Brand Code D, A

For an exhaustive list, please reference the Variable Importance Section in Code to run. Weak and Uncommon Variables were excluded due to the unlikelihood of meaningful results.

Strong Variables plotted Against PH Values:



Strong relationships aren't clear, partially due to the amount of variables at play, but one can see that lower values of PH tend to stop occurring as these Strong Variables hit certain set points. An analysis of interaction effects and ANOVA would prove useful in teasing out certain combination effects between variables.

- **bowlSetpoint:** Bowl Setpoints at 110 and 120 tend to hold the most high PH values
- **fillerLvl:** Higher PH values appear at fillerLvl = 120 with fewer low PH scores after that point
- **pressureSetpoint:** The most variable PH values come from setpoint 50, suggesting lower
- values of pressure may keep PH more controllable
- **oxyFiller:** As oxyFiller rises, low PH values seem to reduce in frequency
- **mnfFlow:** As mnfFlow increases, PH becomes less predictable in outcome
- **hydPressure1:** As hydPressure1 tends to increase, low PH values seem to reduce in frequency

CONCLUSION

Although many hope to find a silver bullet in a single model to guide actions and release constituents from using their own intuition, our analysis reveals highlights and next steps. EDA revealed some correlated variables and skewness, which were removed and transformed. Brand Code B seemed relatively unimportant compared to the other Brand Codes, despite being used heavily. Several models performed strongly, but the Random Forest model was chosen as the model to predict with. Checking commonalities between models found that variables Mnf Flow, Oxy Filler, Hydr Pressure 1 - 3, Bowl Setpoint, and Filler Level were acknowledged as some of the strongest variables to predict PH. Brand Code C, Filler Speed, and Pressure Vacuum are strong predictors of PH in some models, but not others. Alc Rel, Carb Rel, and Brand Codes D, A were weak but common among all models. Python and R corroborated models with small degrees of difference in accuracy. Actionable steps for change, if desired, towards impacting PH for product quality or cost savings are below.

Grouped by their relevance from the Variable Importance Section, they are the following:

1. Identify which variables that have been highlighted in Variable Importance and are easiest to change or implement in production first. Confirm their impact with statistical testing for a difference in means and commit a process change to production.

2. **Strong and Common Variables:** Confirm set points across have been thoroughly tested for ideal levels. Results from changes in these variables will be clear and easy to determine which levels are ideal through testing and could result in several tenths of changes in PH (i.e. changing these could easily move PH from 8.2 to 8.6). In essence, these are an individual's most powerful, heavy-handed tools to affect PH. Statistical testing for difference in means between batches would be necessary.

3. **Strong and Uncommon Variables:** Exploration through ANOVA and more data to confirm their roles in controlling PH as alternative methods, particularly if these have an impact on cost. These variables could be tested to confirm relationships in case they are significantly cheaper to use as means of controlling PH.

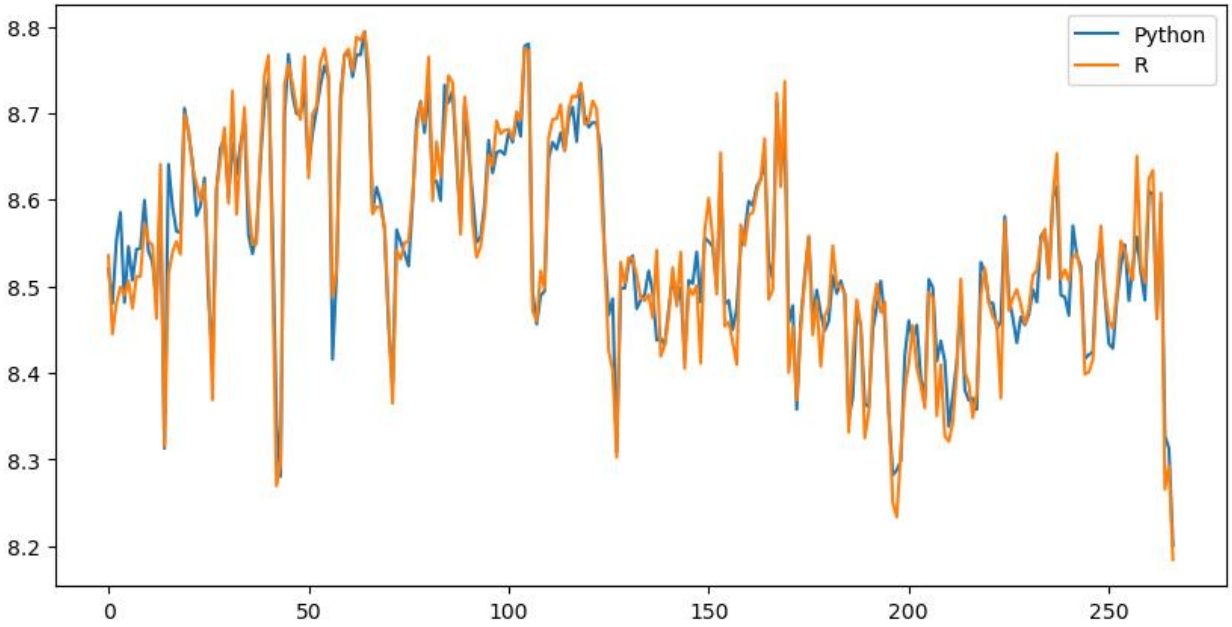
4. **Weak and Common Variables:** Exploration to support 'fine tuning' changes in PH, or using them as a whole to notably change PH in the order of tenths. If a cost-benefit analysis showed that a substantial increase or decrease across all of these variables showed improvement in equivalence to a more expensive variable, these could be an alternative. A PLS/PCA approach performed quite well, although not the best, which validates this as a reasonable option depending on cost.

APPENDIX A: Python Comparison

While the primary implementation of our modeling leveraged R, we also implemented the modeling protocol in Python, primarily leveraging the Sci-Kit Learn package. See Appendix B for a link to the full implementation.

The implementations produced comparable results but with significant differences in final fit and resulting error metrics. We attribute these differences to the different implementations of imputation, transformation and hyperparameter tuning across packages in both languages (namely, mice/caret and sklearn for R and Python, respectively). While we attempted to reconcile these differences by aligning logic (for example, using the same number of folds and parameter grids for cross-validation in hyperparameter tuning), some gaps remained between the two implementations.

More importantly, however, the Python implementation provided support for the top chosen models. Specifically, ensemble tree approaches performed best. Moreover, the predictions produced out-of-sample largely aligned, as demonstrated by the plot below (based on the final Random Forest models).



Full results from the Python implementation are summarized below.

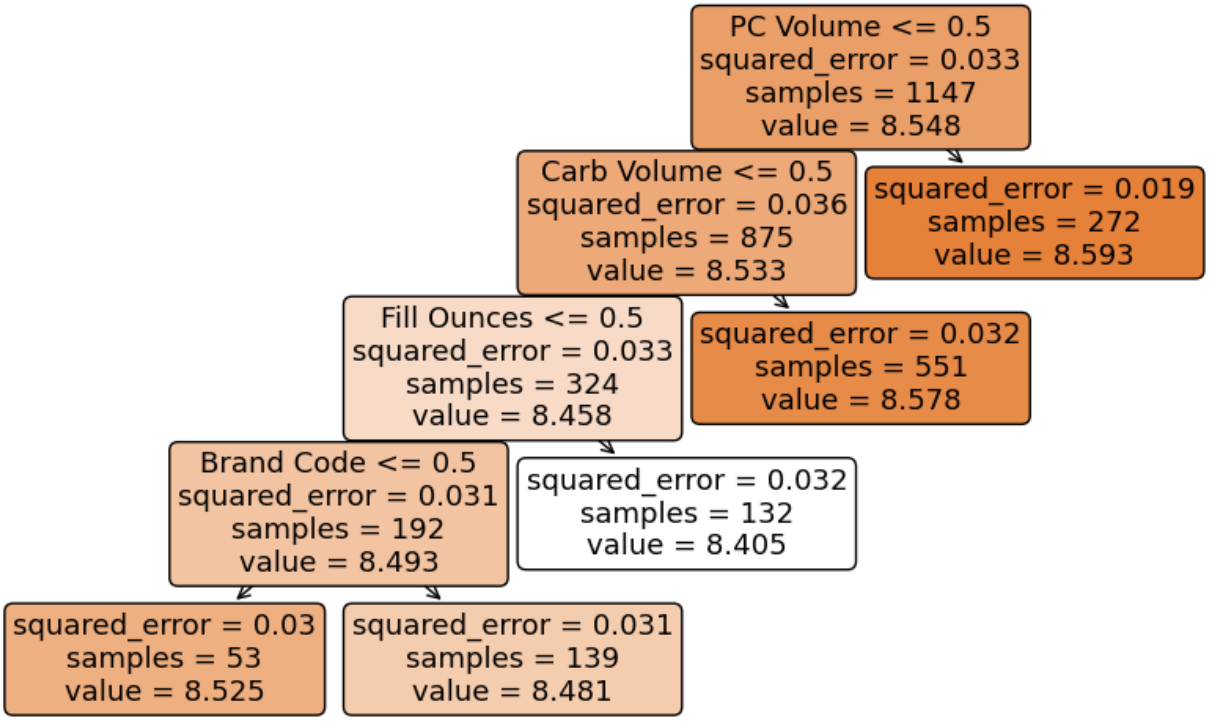
model	RMSE	Rsquared	MAE	MAPE	Transformed	Tuned
GradientBoostingRegressor	0.093906	0.691364	0.065705	0.007694	True	True
RandomForestRegressor	0.098638	0.659480	0.070381	0.008248	True	True
KNeighborsRegressor	0.120107	0.495110	0.088074	0.010344	True	True
SVR	0.115693	0.531543	0.088318	0.010346	True	True
DecisionTreeRegressor	0.125036	0.452820	0.090536	0.010612	True	True
MLPRegressor	0.131836	0.391688	0.096170	0.011274	True	True
LinearRegression	0.130391	0.404952	0.100152	0.011745	True	False
Ridge	0.130415	0.404730	0.100162	0.011747	True	False
PLSRegression	0.132002	0.390160	0.101584	0.011910	True	False
Lasso	0.139364	0.320232	0.109224	0.012801	True	True

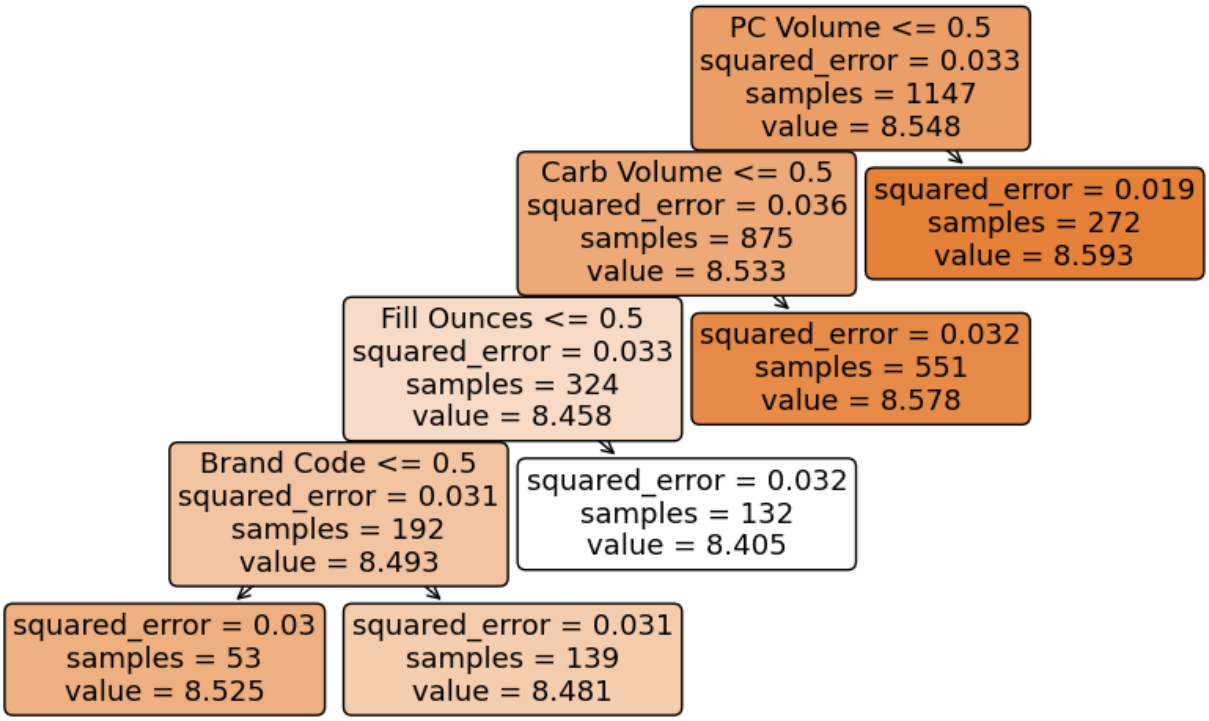
APPENDIX B: Links to Implementation

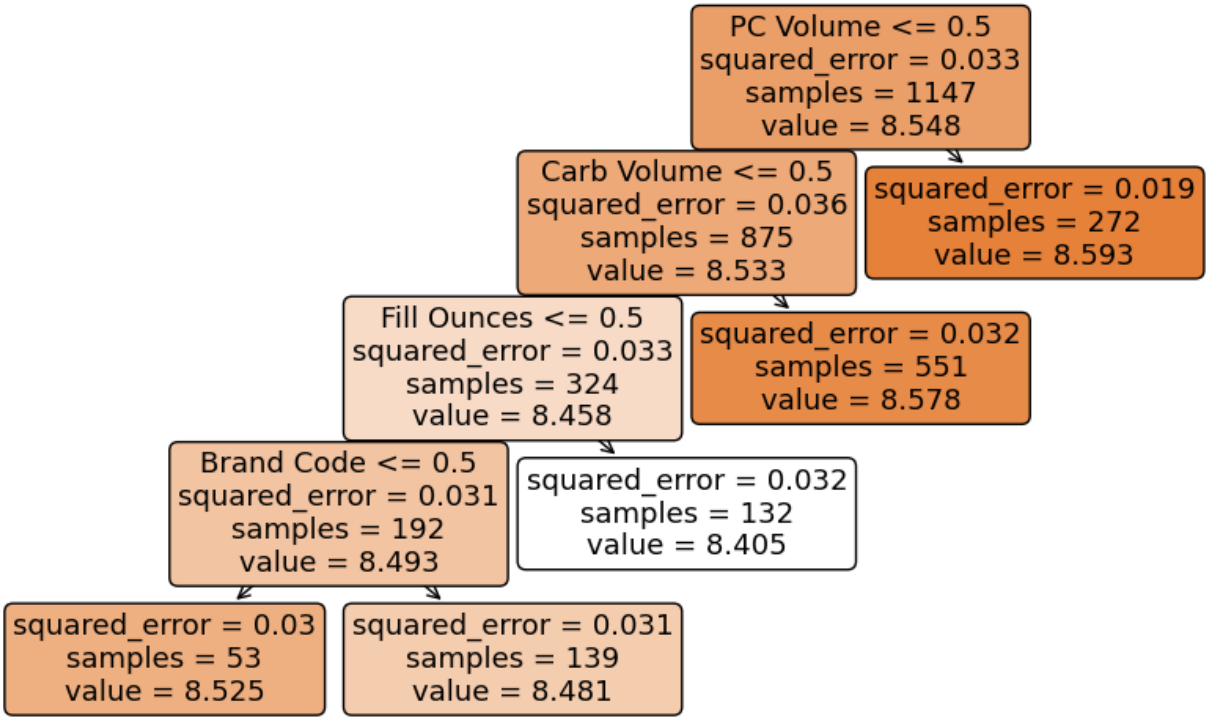
R Modeling Implementation: https://github.com/d-ev-craig/DATA624/tree/main/Project%202
Python Implementation: https://github.com/kac624/cuny/blob/main/D624/project2.ipynb

ADDITIONAL PLOTS

Sample trees from RF







Decision Tree tuned over Complexity