Predictive Analytics of Soft Drinks Production

Anjum Banu Ismail
MSc. Data Analytics, Queen's University Belfast
aismail03@qub.ac.uk

Abstract - This report aims to represents my analysis for the soft drink production to predict the quality if product in several stages of the production process.

I. DATA INSIGHTS AND VISUALIZATION

Given the historical data from Jan 2021 to June 2021, we see the predict variable "g4_var2" is dependent on the values generated from its preceding variables. This is a complete time series forecasting and the variable "datetime" should be considered". The optimal value is -0.8574 and hence a threshold of +0.5 can be considered to predict the variable being optimal. A quick overview of the data spread is given in fig1.

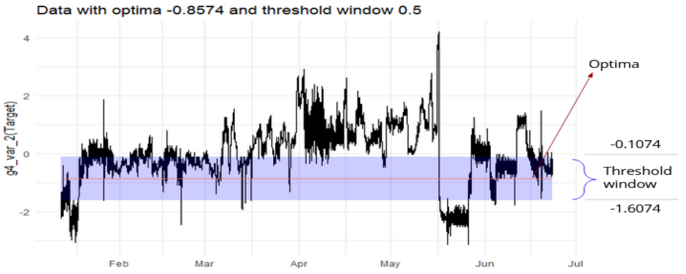


Figure 1 Time Series forecast

Despite the data being normalized to mean=0 and var=1, target fails to show to any relationship with other variable except g2_var_4 (Positively correlated).

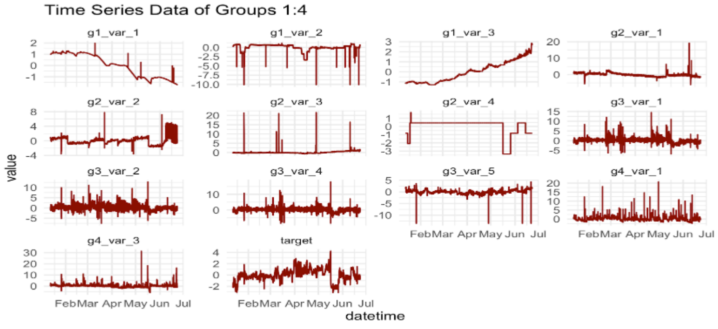
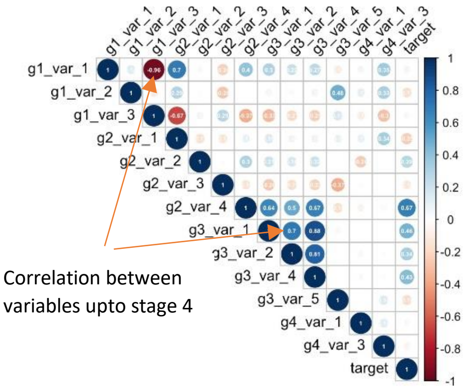


Figure 2 Variable visualization

The outliers seen are very informative about the subject-area and data collection process. It's essential to understand how these outliers occur and it's best to keep them, they can capture valuable information that is part of our study area. Though the data looks noisy and It is obvious, that we have peaks in every month. Smoothing is forbidden because it neglects the ups and downs associated with random variation and underlying phenomenon.

Across variables the levels of collinearity are show in Figure 3.



Correlation between variables upto stage 4

Figure 3 Explained Correlation

II. FEATURE ENGINEERING

Around 0.1% of predict variables are missing, this may be due to failure to load information. The right decision is deleting missing data rows and there is no need of imputation keeping in mind the missing percentage is small. However Variable "g3_var_3" and "g6_var_9" has 0 variance and this will not improve the performance of the model. In that case, it is removed.

Using just the variables alone to predict optimal value is not sophisticated and will likely result in a poor model. Nevertheless, this information coupled with additional engineered features may ultimately result in a better model. This is achieved by including the values at previous time steps by lagging and rolling mean.

g4_var_2	Lag_1 (15min)	Lag_2 (30 min)	Lag_3 (45 min)	Lag_4 (60 min)
-1.537	NA	NA	NA	NA
-1.613	-1.537	NA	NA	NA
-1.689	-1.613	-1.537	NA	NA
-1.689	-1.689	-1.613	-1.537	NA

Table 1 Lagging performed on target variable

g4_var_2	Roll Mean3	Roll Mean4	Roll Mean5
-1.537	NA	NA	NA
-1.613	NA	NA	NA
-1.689	NA	NA	NA
-1.689	-1.613	NA	NA
- 1.764	-1.663	-1.632	NA
-1.840	-1.714	-1.689	-1.658

Table 2 Rolling mean performed on target variable

III. MODEL BUILDING

The entire data is divided into three separate data sets, i.e. train, validation and test, each of which will be used for only one phase of the project. When creating each data set, we have ensured to keep mixture of data points at the high/low extremes, this ensures that the model can and must be accurate at all ranges of the spectrum.

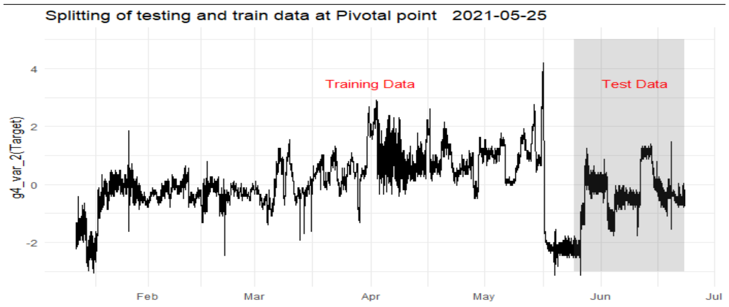


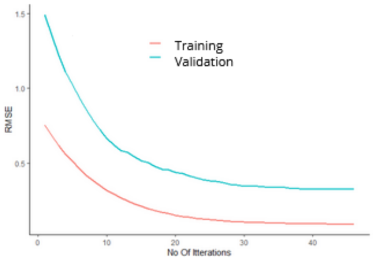
Figure 4 Data Split

Because all of the input variables are numeric and the problem is a simple supervised binary classification, Ensemble method seeks to create a strong classifier (model) based on "weak" classifiers, hence have used "**XGBoost**" to train the data. We are going to use Root mean square error (RMSE), Accuracy and Sensitivity to evaluate the quality of our predictions.

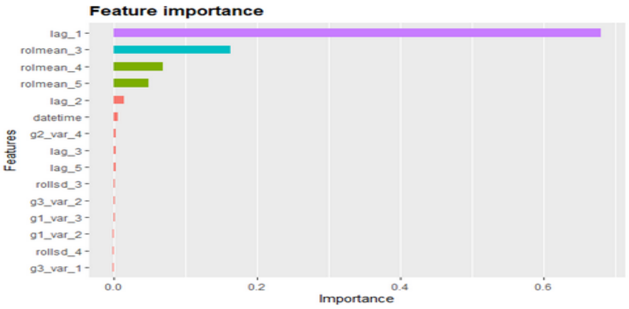
Grid search 5-fold validation model along with all the hyper tuning parameters are used to find an optimal solution. Here we will tune 5 of the hyperparameters that are usually having a big impact on performance like eta, max_depth, subsample, colsample_bytree, gamma. The best module generated is used to train the model and achieve RMSE as below.

RMSE:

Train	**0.091**
Val	**0.323**



Feature importance is used to estimate the relative importance of input features. The plot shows the high relative importance of lag and rolling mean and a lesser degree of importance to the actual provided variables.



For the data trained in the model we have obtained the classification accuracy of about **98%** on the validation set and with RMSE of **0.1**. This occurs to be a good model.

The trained model was used to predict the test data and a confusion matrix provides a more detailed breakdown of correct and incorrect classifications. In our case, the classifier predicted all the 929 non optimal and 1565 optimal. However, it incorrectly classified 95 instances of non-optimal data as optimal and another 96 instances of optimal data as non- optimal. Looking at our problem statement non optimal category getting predicted to be optimal is huge risk for our client and hence sensitivity plays an important role.

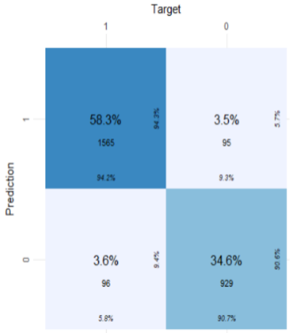
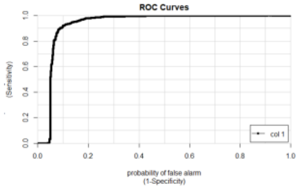


Figure 5 Confusion Matrix



Accuracy	92.8%
Sensitivity	90.7%
Specificity	94.2%
Kappa	84.9 %
RMSE	0.03
Sum of Sqr	1105.79

IV. MODEL DEPLOYMENT

Incoming training Data can be stored in on-premise, in cloud storage, or in a hybrid of the two. It makes sense to store your data where the model training will occur and the results will be served. Build a web app using Flask framework. It will use the trained ML pipeline to generate predictions on new data points in real-time. Create a docker image and container. Publish the container onto Cloud Container Registry. Deploy the web app in the container by publishing onto Registry. Once deployed, it will become publicly available and can be accessed via a Web URL.

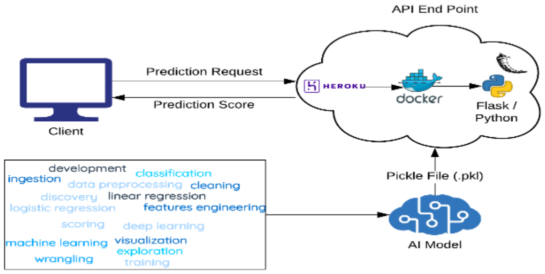


Figure 6 Model deployment

V. REFERENCES

- https://arxiv.org/pdf/1603.02754.pdf
- https://kth.diva-portal.org/smash/get/diva2:1089425/FULLTEXT01.pdf
- https://christophm.github.io/interpretable-ml-book/rules.html
- https://cran.r-project.org/web/packages/datarobot/vignettes/TimeSeries.html
- https://r4ds.had.co.nz/