CONTIAMO

Schneider data challenge

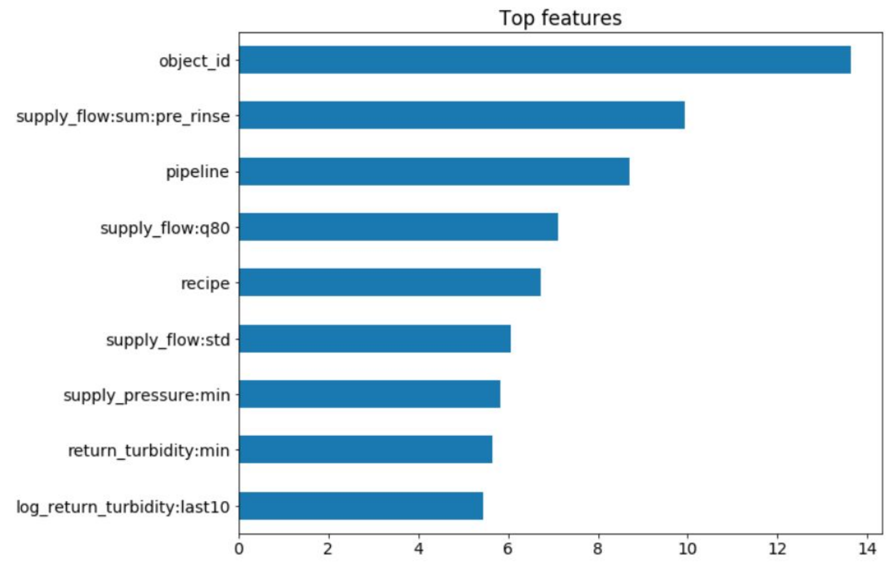
Stage 2: Modeling Report

The data science team (contact: ian@contiamo.com)

Outline

Model explanation: Feature importance

- Object_id is by far the most important feature.
- Pipeline and Recipe consistently rank in the top 5, close to the best time-series features.
- An simple model with only the 3 features above already performs rather well (~**31%** error).
- A dozen simple time-series features provide most of the lift from the a-priori model (~**27%** error).
    - Supply flow is generally the most important time-series.
- None of the thousands of time-series features we tried brought any significant improvement.
    - The fact that we ranked close to the top makes us confident that we did not miss any crucial time-series feature.

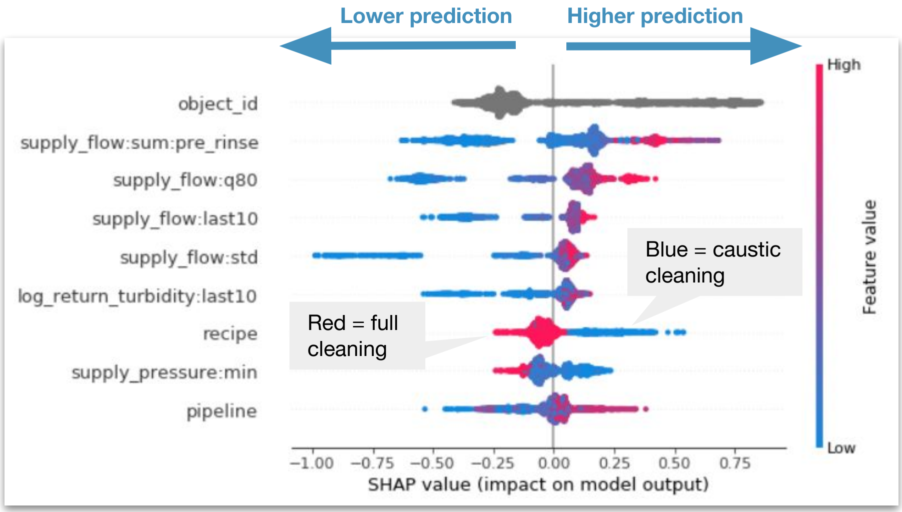


Legend:

q80	80% quantile (essentially a smoothed maximum)
last10	average of last 10 values
std	standard deviation

Note: unless a phase is specified (at the end), features are calculated on the entire length of the time-series.

Model explanation: SHAP values



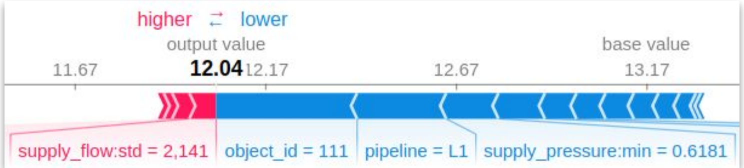
Interpretation

- SHAP values measure the contribution of each feature to the final prediction in our model.
- Each dot represents one cleaning process.
- The color of each dot is associated to the value of the feature.

Further reading: [A unified approach to interpreting model predictions]

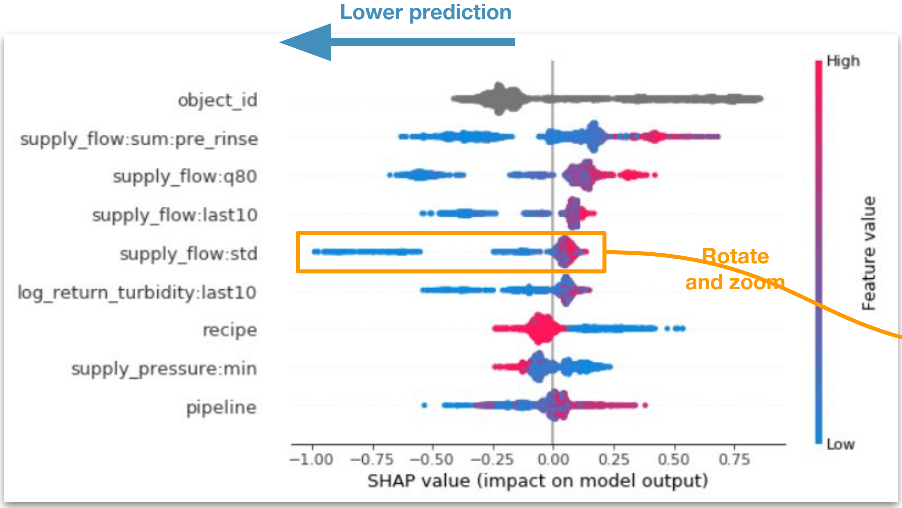
This chart shows the effect of feature values on our model's prediction for each individual observation.

Example: the (log) prediction for process id **20016** is "low" due to the contribution of object id (111), pipeline (L1) and supply pressure (min).

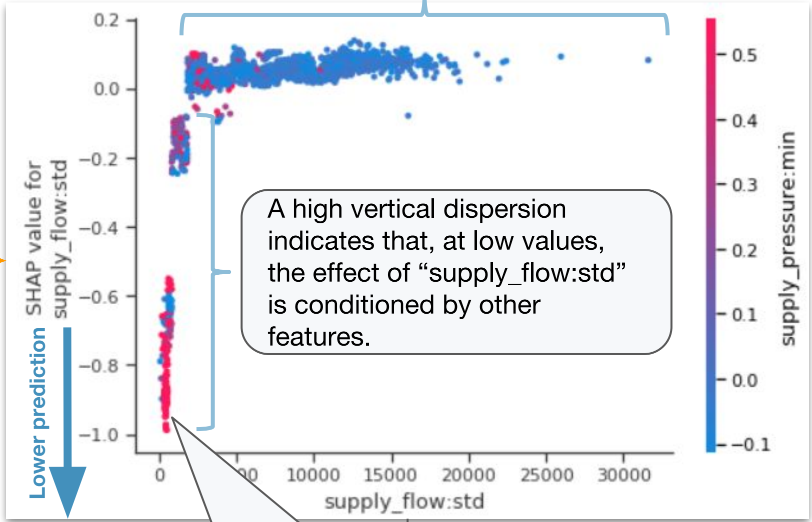


- Object id generally has the strongest impact on the prediction of final total turbidity.
- The caustic recipe (without acid phase, blue dots) contributes to a higher prediction.
- As a rule of thumb, higher pipeline numbers contribute to a higher prediction.
- Generally, higher values of the supply flow contribute to a higher prediction.

Model explanation: SHAP values and feature interaction



Above a certain threshold, "supply_flow:std" makes a consistently small, positive contribution to the prediction.



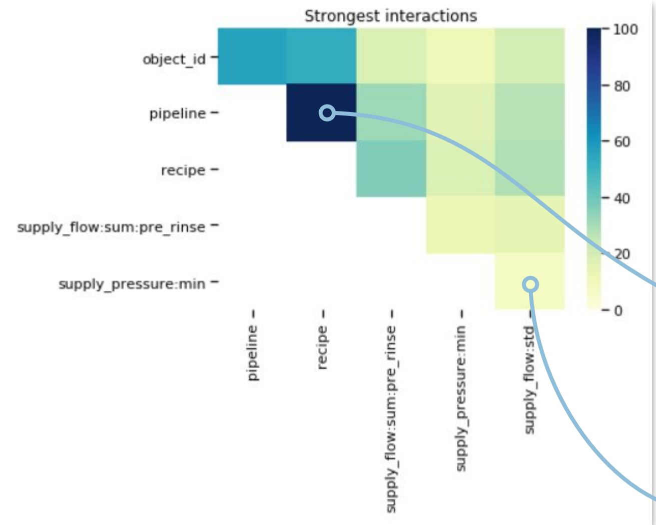
A high vertical dispersion indicates that, at low values, the effect of "supply_flow:std" is conditioned by other features.

Specifically, low "supply_flow:std" values contribute strongly to a lower prediction **when** "supply_pressure:min" is high (red dots).

Note: This is just one example of the type of explanatory analysis that can be conducted in order to answer questions from business actors and increase the confidence in our model.

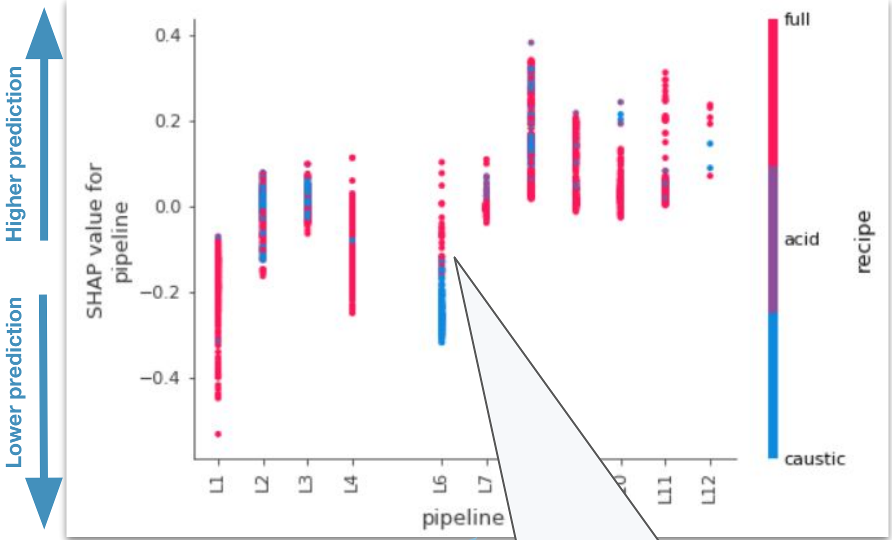
Model explanation: Feature interaction

Feature interaction denotes the extent to which the contribution of one feature to our prediction is influenced by the value of another feature.



We looked at this interaction in the **previous slide**.

Effect of **recipe** on the contribution of **pipeline** to the prediction



Pipeline vs recipe interaction

→ When the **full** recipe is used (red dots), being in pipeline L6 contributes little to the prediction (the dots cluster around 0).
→ When cleaning uses only **caustic** (blue dots), being in pipeline L6 significantly contributes to a lower prediction.

Confidence measures

- We train an auxiliary model to estimate, for each observation, a confidence interval (CI) that contains the true target value in **80%** of cases.
    - The CI can be used to inform decisions on the length of the final rinse (see examples).
    - However, it is difficult to interpret the CI across the very wide range of target values.

Confidence interval is narrow; the model has good confidence.

Predicted value is “close” to true value.

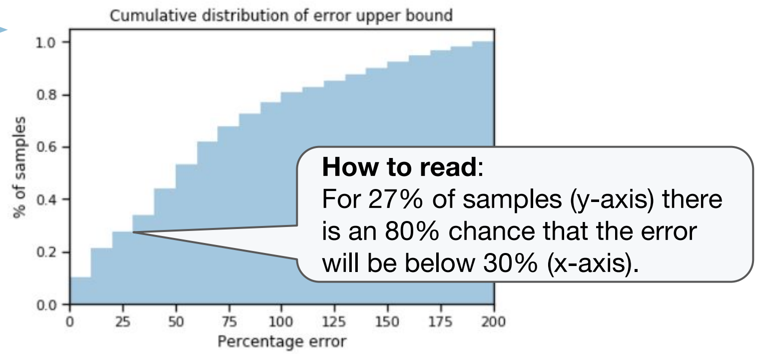
Examples

process_id	object_id	pipeline	recipe	y_low	y_high	conf_interval	y_pred	y_true
20001	405	L4	full	3,543,307	5,129,214	1,585,907	3,936,473	4,318,275
20166	930	L8	caustic	885,249	5,512,665	4,627,416	3,694,336	25,080,104

Confidence interval is large; the model has low confidence.

Predicted value is “far” from true value.

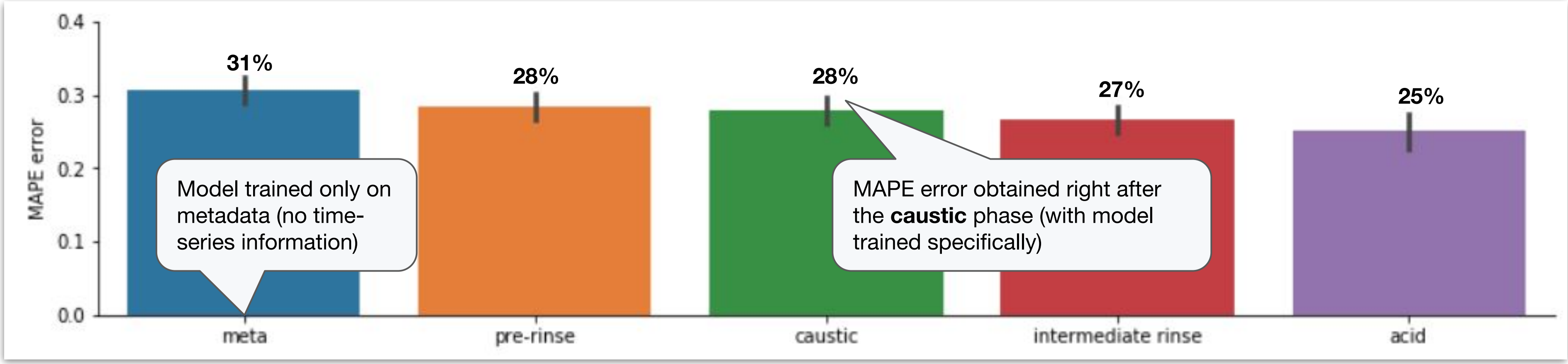
- Therefore, we also used the CI to calculate, for each observation, an 80% upper bound for the error.
    - This is another measure of the confidence of our model; it is less specific but easier to understand for all values.
    - The upper bound is conservative: in many cases the actual error will be much lower!



Prediction improvement with time

Legend for time-series features
timeseries_name:feature[:phase] (phase optional)
e.g. return_temperature:max:acid

Until now we have been explaining our best overall model. Here we focus on cleaning processes that use the full recipe (5 phases). We train a specific model for each point in time: before cleaning starts, after pre-rinse, and after each subsequent phase. The goal is to see how the flow of information in each phase improves our prediction.



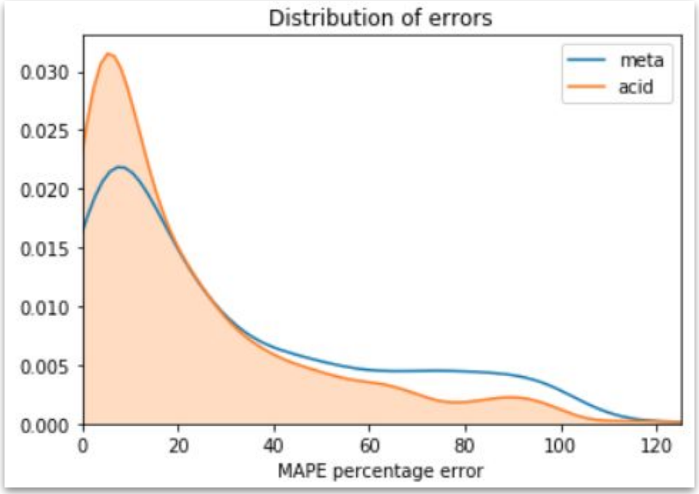
Top 3 features

object_id pipeline recipe	**supply_flow**:sum:**pre_rinse** object_id supply_flow:std	supply_flow:sum:pre_rinse **supply_flow**:first10:**caustic** **supply_pressure**:min	supply_flow:sum:pre_rinse **return_temperature:last10** return_flow:sum	supply_flow:sum:pre_rinse **return_temperature**:max:**acid** supply_flow:sum:interm_rinse

The top 3 features of each model (trained at a specific phase) show **which time-series matter** as each phase in the cleaning process completes.

Prediction improvement with time (cont.)

- The improvement in prediction is somewhat modest:
  - From **31%** average error for a model trained without any time-series information to **25%** for a model trained on all available data up to the acid phase.
- We show how the distribution of errors (over our validation set) improves between the first and last models (metadata vs acid phase):
  - The tail is slimmed down;
  - There are more samples with an (individual) error below 20%;
  - Median error improves from 18% to 13%.



Next steps

- On the machine learning side:
  - Prune the number of time-series features necessary to obtain “good” performance. (This will increase the robustness of the model and make it easier to understand.)
  - Evaluate the usefulness of training a different model for each recipe and phase (rather than one general model).
- On the business side, discuss how exactly the decision on the length of the final rinse is to be made:
  - Should we err on the side of caution when making a decision? In particular, is there a final check to catch dirty objects if the model makes a mistake?
  - Is the (adjusted) absolute percent error a good number to guide the decision?
- The answers might inform our modeling. Indeed, if applicable, we might suggest slightly different approaches.
  - Considering the operational complexity of feeding real-time data from sensors, is it worth adding time-series features to the model? Or is the performance of the metadata model good enough?
  - As another alternative, a classification model could be easier to interpret for the user (e.g. clean, dirty, very dirty).
  - A classification has the advantage of naturally providing the probability (according to the model) of belonging to a class. This is easier to interpret than the confidence indicators we presented earlier.
  - A classification also avoids dealing with target values (final total turbidity) that have a very wide distribution.