Model documentation and write-up

1. Mini Bio

   A Computer Science and Engineering Graduate from the University of Moratuwa, Sri Lanka graduated in 2018. Currently Working as a Research Engineer at Fujitsu-SMU Urban Computing and Engineering (UNiCEN) Corp. Lab.

2. High level summary of your approach: what did you do and why?

   I employed a gradient boosting tree model (LightGBM) with a 10 fold stacking strategy. The model was trained on aggregated statistics of the last n records of a process (n = 10,50,200,500). Also, phase-based statistics and time-based statistics were used.
3. Copy and paste the 3 most impactful parts of your code and explain what each does and how it helped your model.
   a. ```
      oof_preds, test_preds_0, importances, last_gbm = kfold_train(args)
      ```
      10 fold stack training helped get rid of the overfitting of the model and more generalization of the model
   b. ```
      train_labels_out = train_labels[(train_labels['log'] > train_labels['log'].mean(
      ) - 2.5 * train_labels['log'].std()) & (train_labels['log'] < train_labels['log'].mean()
      + 2.5 * train_labels['log'].std())]
      ```
      Outlier removal of the training dataset helped the negative impact on the model by outliers since most of the features are generated by aggregated statistics
   c. ```
      ts_df['return_flow__return_turbidity'] = df['return_flow'] * \
          df['return_turbidity']
        ts_df['return_flow__return_temperature'] = df['return_flow'] * \
          df['return_temperature']
        ts_df['return_flow__return_conductivity'] = df['return_flow'] * \
          df['return_conductivity']
        ts_df['object_residue'] = ts_df['supply_flow'] - ts_df['return_flow']
      ```

      Feature engineering led the model to learn from the interactions between features.

4. What are some other things you tried that didn’t necessarily make it into the final workflow (quick overview)?

   - Aggregation of the statistics from all available records from a process led to model overfitting and low accuracy
   - Statistics that depends on all available processes let to overfitting.
     ex:
     - Avg time taken for a process which uses pipeline 3
   - Could not make sense of the Boolean columns of the dataset
      ```

5. Did you use any tools for data preparation or exploratory data analysis that aren’t listed in your code submission?
   No

6. How did you evaluate the performance of the model other than the provided metric, if at all?
   Competition used a modified MAPE Metric. I evaluated using both modified and not modified MAPE metric

7. Anything we should watch out for or be aware of in using your model (e.g. code quirks, memory requirements, numerical stability issues, etc.)?
   The code takes 3-4 hrs to execute on a laptop with 8 core i7-6700HQ CPU @ 2.60GHz and 16 GB ram on a Ubuntu 16.04 OS

8. Do you have any useful charts, graphs, or visualizations from the process?
   All the useful visualizations and charts are included in the submitted report

9. If you were to continue working on this problem for the next year, what methods or techniques might you try in order to build on your work so far? Are there other fields or features you felt would have been very helpful to have?

   I would try Deep learning to couple with the model that I built so far