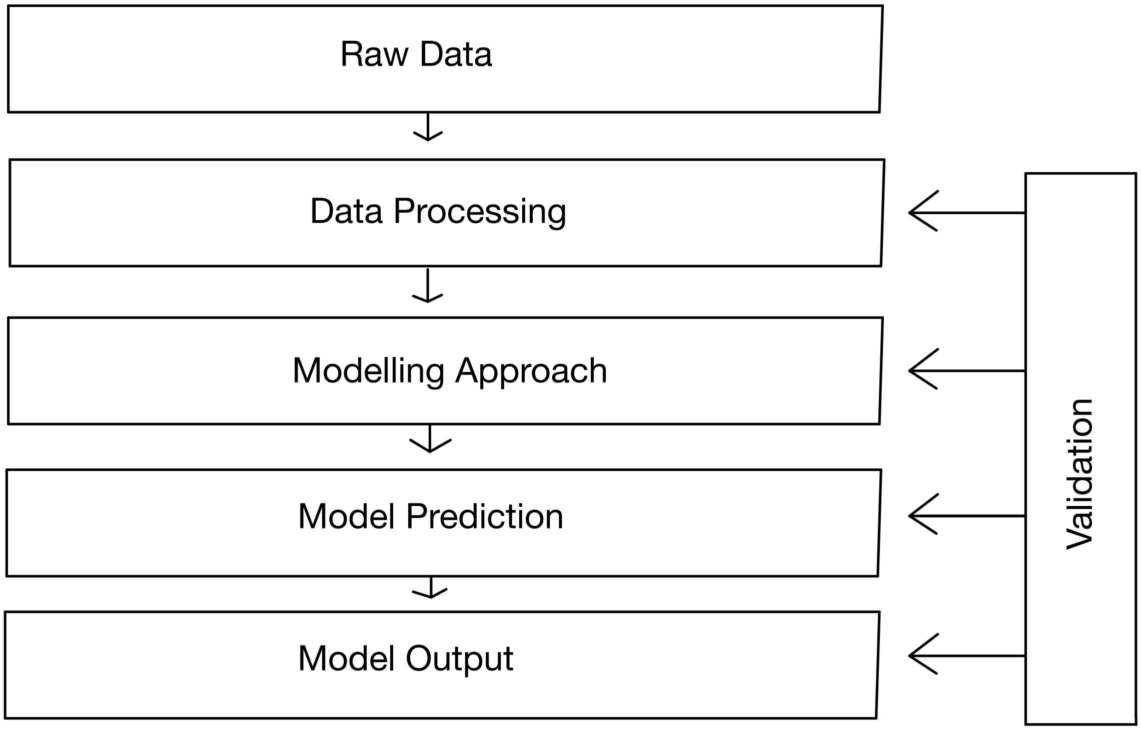
Workflow

A workflow is crucial in transforming the raw data through a series of steps to valuable data or insights. A workflow also ensures that the detailed steps are consistent and repeatable, where other users can recreate the workflow and arrive at a similar result. Implementing a workflow is also beneficial for the team to incorporate improvements or make changes without disrupting the code downstream.



Data Processing

In the real-world environment, data rarely come processed and ready to be churned into predictive models. The primary step of completing the challenge is through the acquisition of valuable data through the mountain of data available. Even though the data provided in the M5 competition went through a series of processing, further improvements can be made to the raw data before we train our predictive models with it. Some examples of other improvements include, but are not limited to, addressing the unequal periods between each time-series data, accounting for special holidays and seasons, and abnormal sale prices.

Feature Engineering

Frequently, feature engineering provides the most improvements to the predictive performance of our models. Using the data sources available and the teams' domain knowledge, teams can create relevant metadata that is fed into the predictive models. The decision to include a feature is dependent on the type of application. Having too many unrelated features may deteriorate the models' performance, referred to as 'noise.' Hence, the teams should always verify if the set of features created are relevant to the cases.

Modelling Approach

With the data ready to be predicted, the teams can plan how to carry out the prediction. An example approach is first to perform the train-test split, then train a model on all the available training data. Next, the teams would perform hyperparameter tuning on the model to improve the prediction capabilities. Lastly, the teams would assess the performance of the model based on the test data. The teams are free to use any approaches which they deem fit. The guidelines given below are just as the phrase suggested, as guidelines. Teams are not expected to strictly follow the steps below.

Model Selection

During this stage, the teams select the model(s) that execute the teams’ modelling approach. To compare the performance of each model, the teams can use RMSE as the comparison metrics.

Hyperparameters

After selecting the best model to carry out the prediction task, teams can optimize the model by going through hyperparameter tuning. Sometimes, hyperparameters may provide better predictive performance. However, teams should be aware of the risk of overfitting while employing hyperparameter tuning.

Model Performance on Test Dataset

Teams are advised to compare the performance of their models on the test dataset, which is the dataset that the model has not encountered. If the model's performance deteriorates significantly, then it suggests that the model is overfitted on the training dataset.

Model Prediction

With the predictive model that the teams have prepared, teams are expected to predict the values of each time series for the next 28 days. The model's output will be compared against the actual values of the next 28 days for each time series. Teams are expected to compile the output where each row is a time series, uniquely identified by a concatenation of its <item_id> and <store_id>. Each column of the output is the predicted value for the day, ranging from F1 to F28. An example of the sample output is as follow:

```
id, F1,..., F28
HOBBIES_1_001_CA_01,0,...,20
HOBBIES_1_002_CA_01,2,...,40
```

Validation

Every step in the workflow detailed above should be subjected to validation. In time series, a common problem is data leak, where the model has access to features that are not made available at the time of prediction. Data leaks are common if teams are doing feature engineering without an adequate understanding of the created features. Another common issue is infeasible predictions. Hence, teams should perform validation and sanity checks at every step of the workflow.

Evaluation

Teams are assessed based on a mixture of interpretability, accuracy, and creativity. On interpretability, I will need to be able to understand your code and be able to run it on my local machine. Do specify any non-native packages used so I can install the dependencies on my local machine.

On accuracy, I will be using RMSE to evaluate the accuracy of the prediction output.

On creativity, teams are encouraged to venture with materials outside of the curriculum.

There are three types of files to be submitted:

1. Your code that includes the workflow (*Your code will need to be able to produce the model(s) and the output CSV*)
2. The model(s) saved in RDS format
3. Output CSV (*refer to the sample output provided in model prediction*)

Before submission, compress all the required files into a single zip file, and label the file as M5Forecasting_TEAMNAME.