A1. Background on you/your team

- Competition Name: M5 Forecasting – Uncertainty
- Team Name: Ouranos
- Private Leaderboard Score: 0.15892
- Private Leaderboard Place: 3

- Name: Ioannis Nasios
- Location: Athens, Greece
- Email: ioannis.nasios@nodalpoint.com

A2. Background on you/your team

- What is your academic/professional background?

  Bachelor Degree in Geology, MSc in Oceanography. Turned to data science through online courses (Coursera Data Science Specialization certificate and many more), constant participation to Kaggle competitions over the last 6 years (59 competitions with 4 gold medals so far, including 2 solo ones), and working experience as a professional data scientist at Nodalpoint Systems over the last 3 years.

- Did you have any prior experience that helped you succeed in this competition?

  Although, as outlined above, I was already a seasoned and experienced Kaggle competitor, truth is that my focus, especially for the last years, have been mainly on computer vision applications and deep learning models (in both Kaggle and my day job). Practically speaking, I had almost zero prior exposure to forecasting problems.

- What made you decide to enter this competition?

  My participation in this competition (as well as in its twin Accuracy track) was explicitly tasked and sponsored by my employer, Nodalpoint Systems.

- How much time did you spend on the competition?

  Because of our company-sponsored engagement with the SpaceNet 6 data challenge (where our team, SatShipAI, also won the 3rd prize), I was not able to start my participation to both M5 tracks before the beginning of May, having thus worked on both of them only over the last 2 months.

- If part of a team, how did you decide to team up?

  I competed as part of a team on the Accuracy track, transferring my own contributed models to the Uncertainty competition. The solutions of the Uncertainty track are 100% obtained with my effort alone.

A3. Summary

- The training methods used

  LightGBM and Neural Networks for Accuracy and coefficient optimization for quantile calculations for uncertainty.

- The most important features

  `'item_id'`, `'tm_d'` (day of month), `'tm_w'` (week of year), `‘tm_dw'` (day of week), `'tm_w_end'` (is weekend), `'rolling_mean_60'`, `'rolling_std_60'`, `'rolling_mean_180'`, `'rolling_std_180'`, `'rolling_mean_tmp_1_7'`

- The tool(s) you used

  Python, Tensorflow Keras, LightGBM, pandas.

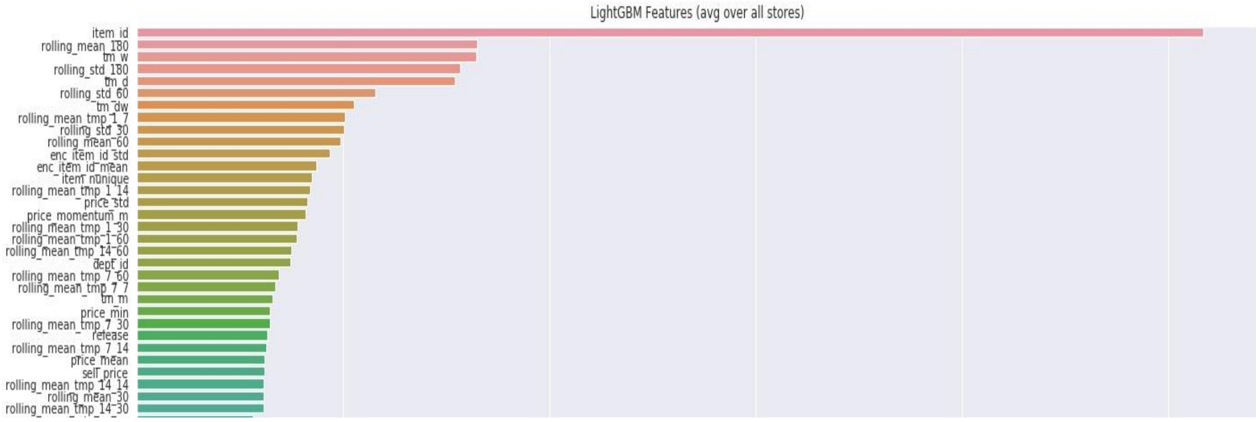
- How long it takes to train your model

  About 2 hours to build data for use with LightGBM. Training itself took 4 hours for the LightGBM models and 2 hours for the Keras ones.

A4. Features Selection / Engineering

- What were the most important features?

  For the LightGBM model, which is the better performing one, the plot below shows the feature importance:



- How did you select features?

  For LightGBM, I just added all that seemed relevant and then removed 2 which improved the validation score. For Keras models, I had to remove several features to improve stability as its predictions were found to be highly unstable.

- Did you make any important feature transformations?

  For Keras models, where I only use data from last 1 year and also removed time features year and month, I added a `log(d)` feature to capture trend.

- Did you find any interesting interactions between features?

  No

- Did you use external data?

  No

A5. Training Methods

- What training methods did you use?

  Uncertainty predictions are based on Accuracy predictions. For accuracy predictions: Trained 1 **LightGBM** model per store for different number of iterations for each store with **all available training data**. Number of iterations were obtained through cross validation

over 3 folds. **Trained 3 Keras Neural Networks** with **data of last 1 year** and simple averaged their predictions.
Accuracy Predictions equal Uncertainty's median Level 12 predictions. From this, with appropriate groupings (averaging), we get median predictions for all 12 Levels. To get predictions for all 9 quantiles (median + 8 other) we can simply multiply every median to a coefficient; this coefficient was calculated per level, by minimizing loss over the last 28 known days (public LB).

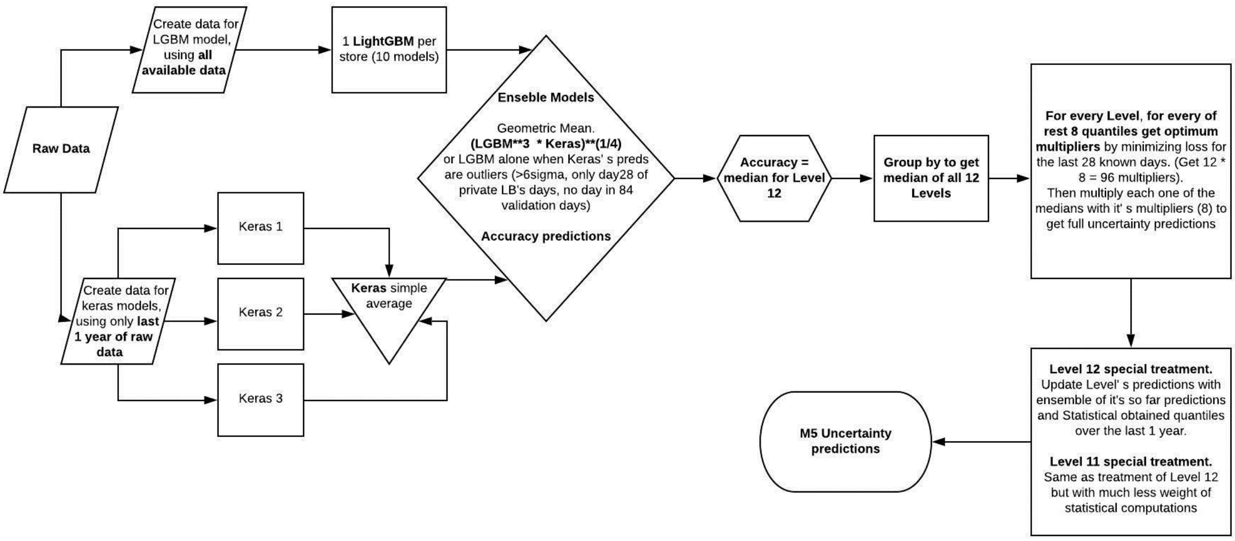
- Did you ensemble the models?

  Yes

- If you did ensemble, how did you weight the different models?

  Based on **3 folds validation** scores, the weighted geometric mean was the best method for ensembling my models.
  (LightGBM_preds3 * Keras_preds)**(1/4)**

The following flow chart tries to capture the whole process schematically:



A6. Interesting findings

- What was the most important trick you used?

  Ensembling LightGBM with an average of Neural Nets to maximize accuracy predictions and **include statistics for item quantile calculations (Level 12)**

- What do you think set you apart from others in the competition?

Trusting local validation and using **statistics for item quantile calculations for Level 12**.

- Did you find any interesting relationships in the data that don't fit in the sections above?

  The higher the aggregation level, the more confident we are in the point prediction and thus we use lower (closer to 1) coefficient multipliers. For multipliers estimation for each level the **normal distribution was used in levels 1 – 9** and a **skew-normal distribution for levels 10-12**. Also, **due to right-skewness** in our sales data on every aggregation level, the last of 9 quantiles (=99.5%) was furthermore multiplied with a factor (1.02 or 1.03).

The evolution of the private LB score and rank by successively incorporating the techniques briefly mentioned above is schematically shown below:

A7. Simple Features and Methods

- Is there a subset of features that would get 90-95% of your final performance?

  Using the 10 most important features (as ranked by LightGBM) for the LightGBM model, and keeping all 19 features for the Keras models, resulted in more than 95% of my final performance.

- What model that was most important? *

  LightGBM

- What would the simplified model score?

  The simplified model with the restricted feature set described above scored 0.1625 on the private LB.

- * Try and restrict your simple model to fewer than 10 features and one training method.

  Training a LightGBM model with 10 most important features (described above) and not using Keras ensembling scores 0.18132 on private LB.

A8. Model Execution Time

- How long does it take to train your model?

  (Already explained in section A3 above).

- How long does it take to generate predictions using your model?

  About 1 hour (assumes that data have already been processed before training)

- How long does it take to train the simplified model (referenced in section A7)?

  Using 10 features with LightGBM model will complete training in about 2 hours (we'll also need to prepare data for the LightGBM model, adding less than 1 more hour to total time).

- How long does it take to generate predictions from the simplified model?

  About 30 minutes.