MSBD5013 Final Project:
M5 Forecasting - Accuracy and Uncertainty

TSE, Chun Lok
20517188
cltse@connect.ust.hk

YUEN, Zhikun
20505288
zyuen@connect.ust.hk

Abstract

In this project, we have used experimented different machine learning methods to forecast the sales and the uncertainty distribution of the upcoming 28 days of different Walmart products. In particular, our LightGBM model achieved the lowest score of 0.59419 in the private leaderboard in the Accuracy track and our Seq2Seq ensemble model achieved the lowest score of 0.17660 in the private leaderboard in the Uncertainty track.
GitHub: https://github.com/SamYuen101234/MSBD5013/tree/master/project2

1 Introduction

The M5 forecasting competition aims to advance theory and practice of forecasting by identifying methods that can accurately predict the sales and estimate the uncertainty distribution. The competition is separated into two tracks and they are complementary in which the same dataset is used for both of the tracks. For the accuracy track, 28 days ahead point forecasts of the competition time series are required. For the uncertainty track, the corresponding median, 50%, 67%, 95%, and 99% prediction intervals from the result of the accuracy track are required.

2 Data

The dataset contains 3 files. The main data is sales_train_evaluation.csv which contains the sales information. In addition, calendar.csv and sell_prices.csv contains additional information about date and the price of the products. The target label is the unit sales of a product at a certain date, in a certain store, category and department.

As described in the competition guide, the structure of the dataset can be organized as a hierarchical tree as shown in Figure 1. Furthermore, different aggregation levels of the products are possible by aggregating according to state, store, category, department or a combination of them. **Our hypothesis is that having a more fine-grained aggregation will lead to better result.**

2.1 Data Preprocessing

Aggregating dataset In sales_train_evaluation.csv, the evaluation sales is not included. To join the table in calendar.csv, we have created columns for the future 28 days and set the sales as NaN. This ensures the combined dataset contains the date related information about the testing 28 days.

Cyclic Encoding Although date features, such as month, day and weekday, are cyclic already, the cyclic function of date features cannot present cyclical data well to a machine learning algorithm. The problem is similar to using one-hot encoding for categories features instead of continuous number. There is a discontinuous jump in the original date function, e.g. at the end of each day, the hour value goes from 23 to 00, albeit only 1-hour difference. We solve this problem by transforming the time features through sine and cosine function as shown in Equation 1.

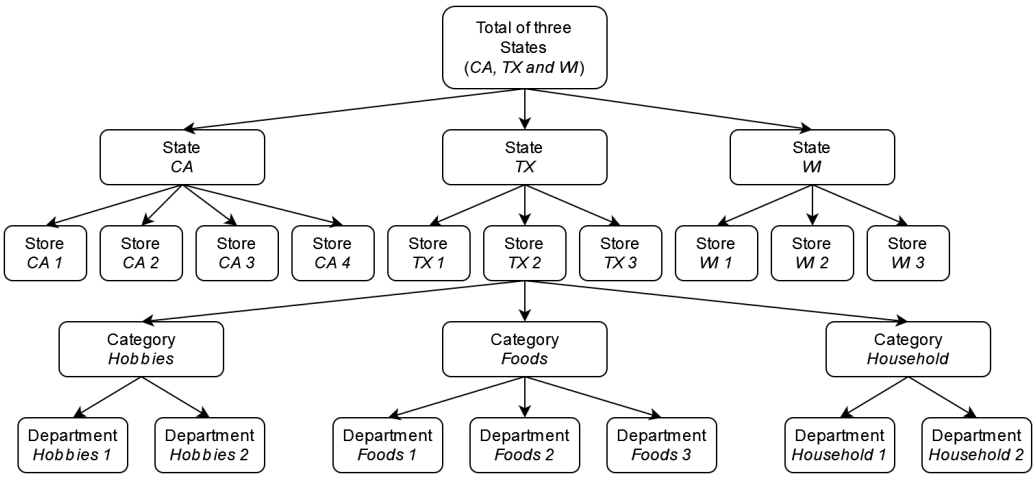


Figure 1: Overview of the hierarchical structure of the dataset

$$X_{sin} = \sin(\frac{2\pi x}{\max(x)}) \quad \text{and} \quad X_{cos} = \cos(\frac{2\pi x}{\max(x)}) \tag{1}$$

2.2 Feature Engineering

Extra features are also crafted to provide additional insight. We added the day and week in month as well as whether the date is a weekend. Additionally, from this Kaggle notebook [1], we found out that lag features could be useful. Therefore, we constructed lag features by shifting the sales by 28 to 42 days as there may be monthly relationship. Moreover, rolling averages can be identified as trend baselines. Therefore, We also used rolling window with window size of 7, 30, 60, 90, 180 to slide over the previous sales. We compute the mean and standard deviation from the data that is first shifted by 28 days.

3 Methods and Experiments

In this project, we have experimented LightGBM and LSTM Sequence to Seqence model to predict the sales.

We can either formulate the tasks as a regression problem or a time series prediction problem. By treating this problem as regression problem, we can use LightGBM to predict the sales of each day independently based on the features. On the other hand, we can use sequence to sequence approach to model a sequential relationship of the data to predict the future 28 days sales based on the previous 28 days. We believe both methods can achieve good results.

3.1 Accuracy Track

3.1.1 LightGBM

LightGBM is a gradient boosting framework that uses tree based learning algorithms. Boosting allows many weak learners to combine together to form a strong learner. The weak learners are created sequentially with the goal of correctly classifying what the previous learner misclassifies.

Aggregating Different Levels To verify our hypothesis, we experimented on different aggregation levels of the data to see whether there is a correlation between the aggregation level and the validation accuracy. We approach this by considering 3 levels of aggregations, including (1) aggregating unit sales of all products for each store, (2) aggregating unit sales of all products for each store and category, (3) aggregating unit sales of all products for each store and department. The total number of trained models are 10, 30 and 70 respectively.

Early Stopping Early stopping of 50 rounds is used such that the LightGBM training procedure stops when there are no improvements on validation Root Mean Squared Error (RMSE) within the 50 rounds so as to avoid overfitting to the training data significantly.

3.1.2 Sequence to Sequence Model

Long Short-Term Memory (LSTM) (Figure 2) is quite often used in sequential prediction. If we want to use the information from t days before, we need to input the features to a LSTM cell in each time step. However, a single LSTM cell cannot learn all of the complex long-term and short-term dependency. Therefore, we have experimented with Seq2Seq models (Figure 2). The advantages are (1) the number of steps to be forecasted can differ from the number of hidden units in encoders, (2) the past time series can be well encoded by the LSTM encoder instead of passing as the inputs of LSTM only, (3) we can use attention mechanism in Seq2Seq model to use all useful outputs in each encoder's LSTM cell efficiently rather than only using the last hidden unit of the encoder.

Apart from using attention mechanism, we have also experimented on a dilated LSTM [2] (Figure 2) for the encoder part. Dilated LSTM can solve the problem of vanishing gradient in RNN because it reduces the average path length between nodes at different timestamps through different lengths of skip connection, called dilation. Also, dilated LSTM can reduce the computational complexity in long dependency because skip connection reduces the number of recurrent connection. we tried both dilated Seq2Seq and attention-based Seq2Seq.

To optimize the Seq2Seq models, we implemented the official metric, *Weighted Root Mean Squared Scaled Error* (WRMSSE), as the objective function. This means that minimizing the loss can directly achieve the better results in the competition. We set the *learning rate = 0.0003*, *dropout = 0.2*.

High variance is a very common problem in time series because of the unpredictable white noise in the data. Also, LSTM can overfit the time series very easily. To solve this problem, we use 6 different approaches to avoid the problem of high variance.

1. Use the last 3 contiguous 28-day periods to do a 3-fold validation
2. Use dropout in both encoder and decoder
3. Early Stopping
4. Ensemble the forecast results from the 3 models in 3-fold validation
5. Ensemble the forecast of Dilated Seq2Seq model and Attention-based Seq2Seq model
6. Randomly add Gaussian noise to some of the rolling features and previous day sales

All of the techniques above can help reduce the variance and achieve better result.

3.2 Uncertainty Track

The required output of the uncertainty track is the predicted value of the median, 50%, 67%, 95% and 99% prediction intervals of the uncertainty distribution. Specifically, we have to provide the 0.005, 0.025, 0.165, 0.25, 0.5, 0.75, 0.835, 0.975, 0.995 quantile of the estimated distribution.

The predicted results from the Accuracy Track can be converted to uncertainty distribution for the submission of the Uncertainty Track due to the nature of the competition. We identified two methods, one naive method using residuals of the fitted model and another method using cumulative distribution function of the normal distribution as introduced in this Kaggle notebook [3].

Apart from conversion methods, we also have experimented on directly optimizing the *Weighted Scaled Pinball Loss* (WSPL), which is the official scoring metric for the uncertainty track. In this approach, we use the WSPL function as the loss function for our Seq2Seq models in section 3.1.2 and minimize the loss directly instead of transforming the accuracy results to uncertainty result. **We believe directly optimizing WSPL can achieve the best result in the uncertainty task.**

3.2.1 Naive Method (Baseline)

In time series forecasting problem, residuals are what is left over after fitting a model [4]. We can use the prediction $\hat{y}$ and residual $(y - \hat{y})$ to compute the forecasting interval through Equation 2 with the assumption of $\mathcal{N}(0, 1)$. Since some prediction interval may be lower than 0 after transformation, we round these negative prediction intervals to 0.

$$\textbf{Prediction Interval} = \hat{y}_{t+h} \pm c\hat{\sigma}_h \tag{2}$$

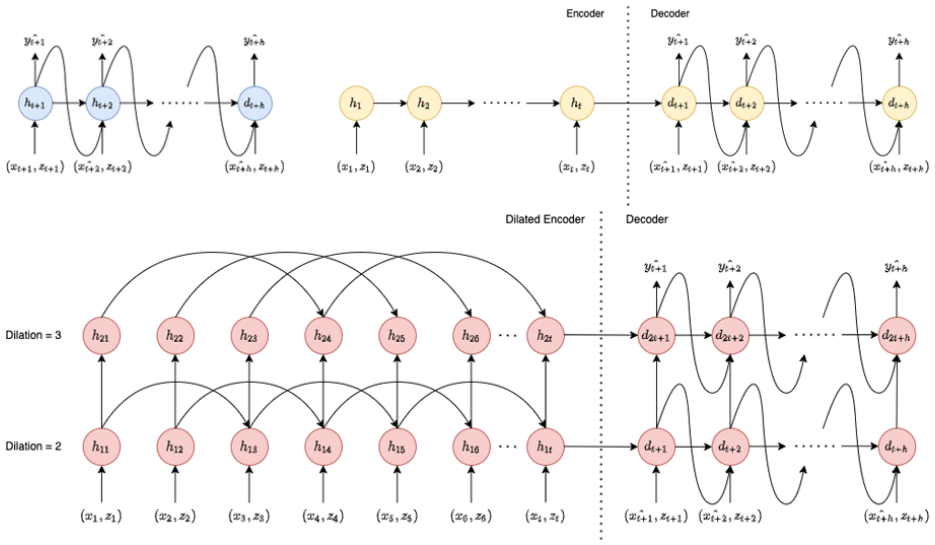


Figure 2: (Top left) A single layer LSTM model for h-step forecast. (Top right) A Seq2Seq model with LSTM encoder and decoder for h-step forecast. (Bottom) A 2-layer Seq2Seq model with a dilated LSTM encoder and LSTM decoder for h-step forecast. x_i denotes the time series (e.g. rolling and lag features). z_i denotes the exogenous features (e.g. festival, date and item ID).

Table 1: Multipliers to be used for uncertainty intervals

Interval	Multiplier c
0.005	-2.5758
0.025	-1.9600
0.165	-0.9741
0.25	-0.6745
0.5	0
0.75	0.6745
0.835	0.9741
0.975	1.9600
0.995	2.5758

, where $\hat{y}_{t+h}$ is the prediction of h-step forecast and c is the multiplier depending on the intervals (Table 1) and $\hat{\sigma}$ is an estimate of the standard deviation of the h-step forecast distribution.

3.2.2 Cumulative Distribution Function (CDF)

This method is provided by this Kaggle notebook [3]. The uncertainty distribution is calculated for each of the prediction level and combinations of levels. The prediction intervals are first mapped from [0,1] to the real number line while multiplying a coefficient that is specific for each prediction level which is found based on trial and error according to the author. Then the mapped values are normalized and then multiplied to the result from the accuracy track to generate the uncertainty distribution. As mentioned by the author, there are a lot of trial and errors to get the coefficients right. We consider this as non-scientific which cannot be fully explained and should not be used in academia. Nonetheless, we have tried to apply this method to generate the uncertainty distribution.

3.2.3 Optimizing the Weighted Scaled Pinball Loss (WSPL)

The WSPL loss is used as the official scoring metric for the leaderboard score calculation. It is a weighted version of the SPL loss. The SPL loss is calculated as follows:

$$\textbf{SPL(u)} = \frac{1}{h} \frac{\sum_{t=n+1}^{n+h} (Y_t - Q_t(u)) u \mathbf{1}\{Q_t(u) \leq Y_t\} + (Q_t(u) - Y_t)(1-u)\mathbf{1}\{Q_t(u) > Y_t\}}{\frac{1}{n-1}\sum_{t=2}^{n} |Y_t - Y_{t-1}|} \quad (3)$$

Table 2: WRMSSE of different approach we tried in the Accuracy Track

Method	WRMSSE	
	Private	Public
LGBM Store Dept	**0.59419**	**0.69987**
LGBM Store Cat	0.61885	0.71367
LGBM Store	0.63840	0.73367
Dilated Seq2Seq	0.65082	0.73131
Attention Seq2Seq	0.66373	0.76811
Ensemble Seq2Seq	0.63256	0.73027

, where Y_t is the actual future value of the examined time series at point t, $Q_t(u)$ the generated forecast for quantile u, h the forecasting horizon, n the number of historical observations, and $\mathbf{1}$ the indicator function. The weighted SPL (WSPL) is then the average performance of the 9 quantiles across all $42,840$ series. A lower WSPL score is better.

Although there are no ground truth labels of the uncertainty forecast for 9 quantiles, the WSPL function (Equation 3) can still be used to optimize our models since the function only requires the actual future sales as the ground truth label. Therefore, we can easily minimize the loss from the 9-quantile predictions in Seq2Seq models through this function.

In this approach, we use the WSPL function as the objective function to optimize the Seq2Seq models in section 3.1.2. All the models, inputs to the models and hyper-parameters are identical as the approach in accuracy task. The only difference is that we change the number of output of the last fully-connected layer from 1 to 9 so that the 9 outputs represent the 9 quantiles.

3.2.4 Prophet

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Also, Prophet is robust to missing data and shifts in the trend, and typically handles outliers well. Because of the non-linearity and robustness to outliers, we try Prophet instead of seasonal-ARMA or ARIMA model.

We use the open-source version of Prophet provided by Meta to examine whether the sales of items have strong seasonal effects. In this approach, we use the sales series of all 12 aggregation levels to do probability forecasting. The package computes the uncertainty interval directly without the need of converting forecasts to prediction intervals (section 3.2.1). If the result of this approach is pretty well, this implies that the data have strong seasonality.

4 Result Analysis

4.1 Accuracy Track

4.1.1 LightGBM

We have done 3 experiments using LightGBM to evaluate the effect in different aggregation levels. As Table 2 shows, **we indeed observed that a fine-grained approach will lead to significantly better results compared to a more generalized approach.** This may be due to the vast variety of the products in the dataset. Generalizing everything into a single model may not be able to predict different cases accurately due to diverse purchasing behavior of different products in different states, categories and departments. For example, everyday food products such as eggs and milk will always have a high demand since these products are consumed quickly. Whereas household products such as detergent may have much lower demands as these products are comparatively long-lasting.

Moreover, the shopping behavior in different states and stores can vary significantly. Customers in different states may purchase differently as there could be different preferences in food and hobbies. Furthermore, the sales of different stores in the same state also significantly differs. Therefore, a fine-grained approach is able to separately predict for each of these differences and thus resulting in a significant improvement over the generalized approach.

4.1.2 Sequence To Sequence Model

The dilated Seq2Seq model achieved lower WRMSSE than attention-based Seq2Seq model (Table 2). We believe the dilated approach helps preventing vanishing gradient problem and further increasing the gradient signal. We ensemble the results together and achieved a even lower WRMSSE in 0.63256. Even though we have applied lots of methods to prevent overfitting, we can still observe high variance issue from the large gap between the results of private and public leaderboard.

4.1.3 Feature Importance

We display the feature importance of the LightGBM models of different aggregation levels in Figures 3, 4 and 5. From the visualization, we observe that the engineered features are important such as week in month and shifted rolling means. Overall, the item_id is overwhelmingly important in predicting the sales across the experiments. Other features become more important when the data is being aggregated to a higher level such that there are fewer models to predict different products.

To explain the input features of the Seq2Seq model, we used a technique in Explainable AI called SHapley Additive exPlanations (SHAP). It is a game theoretic approach to explain the output of any machine learning model. We used DeepExplainer provided by SHAP. We plot both the Shapley values of the features of both encoder and decoder in the attention-based Seq2Seq model in accuracy task (Figure 6 and 7). The feature attribution are visualized as "forces". Each feature value is a force that either increases (red) or decreases (blue) the prediction. 0.96 in Figure 6 and 1.27 in Figure 7 are the baseline of Shapley values. We can view the range of each feature as feature impact [5]. This means that wider range of the feature, the larger the impact is. From the SHAP values plot in Figure 6 and 7, we can obverse that the recent the sales information, including previous day sales, rolling and lag features, have higher impact than the older sales data.

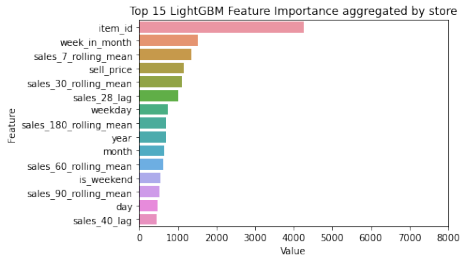


Figure 3: Top 15 LightGBM Feature importance aggregated by store

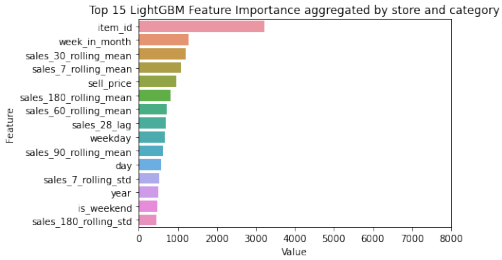


Figure 4: Top 15 LightGBM Feature importance aggregated by store and category

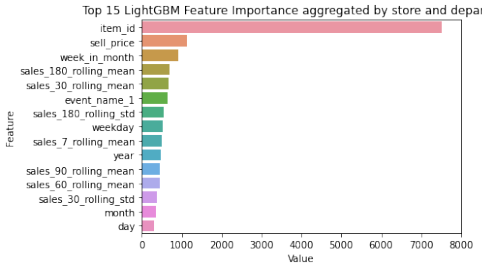


Figure 5: Top 15 LightGBM Feature importance aggregated by store and department

4.2 Uncertainty Track

In this task, naive method gets the worst results for all machine learning models we used in the accuracy track. Using CDF significantly improves the performances. However, compare these two approaches to transform sales forecast to quantile intervals, Seq2Seq models achieves much lower WSPL than LightGBM. This implies that the sales prediction of the accuracy task predicted by Seq2Seq models are closer to the median in probability forecast with standard normal distribution than the predictions of LightGBM, even though LightGBM is better in the accuracy track. Among all the approaches we experimented, using WSPL as the objective function to optimize the Seq2Seq

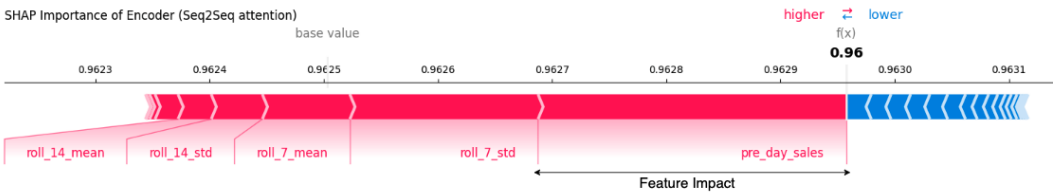


Figure 6: SHAP values of the input features of the Seq2Seq encoder

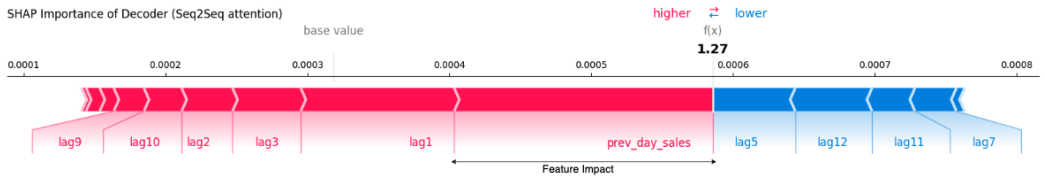


Figure 7: SHAP values of the input features of the Seq2Seq decoder, lag1 feature means lag 1 to lag 28 days and lag2 means lag 1 to lag 29 days so on and so forth

models achieve the best results with $WSPL = 0.17660$ and 0.20204 in the private and public leaderboard. One interesting point in the Seq2Seq models is that attention-based Seq2Seq is slightly better than dilated Seq2Seq which is different from the observation in the accuracy task.

In Figure 8, we also plot one example of the 95% prediction interval of forecast from the ensemble Seq2Seq model with different approaches in the validation days from Day 1913 to 1941. We observe that the lower the WSPL is, the narrower the prediction interval is. In this case, using WSPL to optimize the Seq2Seq model has the narrowest 95 % prediction interval.

Meta Prophet From the result, Prophet is even worse than our naive method while taking massive amount of time to run. Since it supports parallel computing, we used all 24 CPU cores available. Each core computes and forecasts one time series each time. With more than 40,000 time series, it still takes nearly 80 hours to complete. Also, the statistical and mathematical concepts behind Prophet are much more difficult than seasonal-ARMA or ARIMA. We should only use it after we

Table 3: WSPL of different approach we tried in the uncertainty Track

Method	WSPL	
	Private	Public
LGBM Store Dept + Baseline	0.36654	0.32224
LGBM Store Cat + Baseline	0.35785	0.32357
LGBM Store + Baseline	0.35743	0.32722
LGBM Store Dept + CDF	0.24809	0.20326
LGBM Store Cat + CDF	0.23555	0.20392
LGBM Store + CDF	0.23761	0.20882
Dilated Seq2Seq + Baseline	0.27400	0.27841
Attention Seq2Seq + Baseline	0.28307	0.29136
Ensemble Seq2Seq + Baseline	0.20617	0.21716
Dilated Seq2Seq + CDF	0.19956	0.20403
Attention Seq2Seq + CDF	0.20617	0.21716
Ensemble Seq2Seq + CDF	0.19703	0.20515
Dilated Seq2Seq + WSPL	0.18343	0.20485
Attention Seq2Seq + WSPL	**0.17641**	0.20824
Ensemble Seq2Seq + WSPL	**0.17660**	**0.20204**
Meta Prophet	0.45768	0.45178

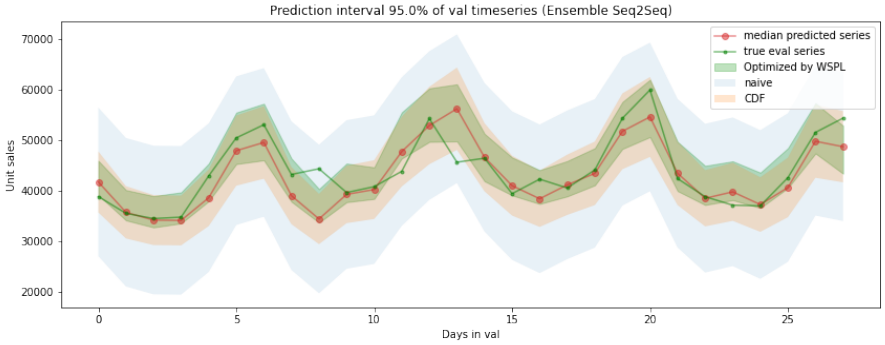
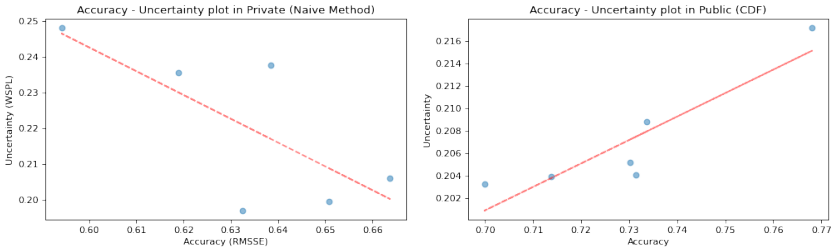


Figure 8: An example of 95 % prediction interval of different approaches to compute the uncertainty with ensemble Seq2Seq model



(a) Private leaderboard (b) Public leaderboard

Figure 9: Scatter plot of accuracy-uncertainty scores of CDF approach

understand all the concepts behind it very well first. Otherwise, we should not use it. Furthermore, it takes only the sales as input while ignoring other useful data, e.g. calendar. This may be one of the reasons why it performs so poor.

4.3 Relation Between Accuracy and Uncertainty

Since the result of quantile forecast can be directly transform from accuracy task, we want to know the relationship between them. In Figure 9, we plot the accuracy-uncertainty scores of all approaches we tried with the CDF approach (section 3.2.2) in both private and public leaderboard separately and compute the best fit line. We can see that there is a inversely proportional relationship in the private leaderboard result and a proportional relationship in the public leaderboard result. Thus, we are hard to declare that there is a correlation between the accuracy and uncertainty scores through this approach.

5 Conclusion

After performing the experiments, we have chosen the best models for the tracks. LightGBM is used for the accuracy Track. We obtained the best result with private score = 0.59419 (Table 2). Ensemble Seq2Seq (Dilated and attention) is used for the task of modelling the uncertainty distribution. We obtained the best result with private score = 0.17660. (Table 3).

6 Contribution

TSE, Chun Lok: LightGBM training and testing, report writing.
YUEN, Zhikun: LSTM Seq2Seq, Prophet training and testing, report writing.

References

[1] Yakovlev K. M5 - Lags features. Retrieved April 21, 2022, from https://www.kaggle.com/code/kyakovlev/m5-lags-features/notebook. 2

[2] Chang, Shiyu and Zhang, Yang and Han, Wei and Yu, Mo and Guo, Xiaoxiao and Tan, Wei and Cui, Xiaodong and Witbrock, Michael and Hasegawa-Johnson, Mark and Huang, Thomas S.; Dilated Recurrent Neural Networks; 10.48550/ARXIV.1710.02224 3

[3] Kkiller. From point to uncertainty prediction. Retrieved April 21, 2022, from https://www.kaggle.com/code/kneroma/from-point-to-uncertainty-prediction/. 3, 4

[4] Hyndman R J and Athanasopoulos G. Forecasting: Principles and Practice (2nd ed). Retrieved April 21, 2022, from https://otexts.com/fpp2/. 3

[5] Lundberg SM, Nair B, Vavilala MS, Horibe M, Eisses MJ, Adams T, Liston DE, Low DK, Newman SF, Kim J, Lee SI. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. Nat Biomed Eng. 2018 Oct;2(10):749-760. doi: 10.1038/s41551-018-0304-0. Epub 2018 Oct 10. PMID: 31001455; PMCID: PMC6467492. 6