報 告

# Practical Data Science - Final Report

Rakuten Card

Refaldi Putra — Graduate School of Engineering, The University of Tokyo
refaldiputra@g.ecc.u-tokyo.ac.jp

Yi Zhang — Graduate School of Frontier Science, The University of Tokyo
6463389093@edu.k.u-tokyo.ac.jp

Guanting Liu — Graduate School of Arts and Sciences, The University of Tokyo
8252598822@g.ecc.u-tokyo.ac.jp

Zihan Li — Graduate School of Information Science and Technology, The University of Tokyo
1396367002@g.ecc.u-tokyo.ac.jp

Zekun Cai — Graduate School of Frontier Science, The University of Tokyo
7292502406@edu.k.u-tokyo.ac.jp

Mergen Kungaa — Natural Environmental Studies, GSFS, The University of Tokyo
kungaa.mergen@s.nenv.k.u-tokyo.ac.jp

**keywords:** transaction demand, time series forecasting, machine learning, deep learning, tree model

**Summary**

With the development of big data technology, understanding and modelling people's behavior of withdrawing money from bank has been an important topic in recent years. The prediction of transaction can reach a relatively good accuracy in normal times. However when some emergencies happen, people may have some reactions respond to the event, leading to unusual fluctuations in the amount of transaction. Meanwhile, vast amounts of data related to money withdrawal and special events can be stored and exploited. Such informative data should be utilized to predict the upcoming transaction fluctuations, giving banks an early warning so that they can have enough time to prepare for capital in place to deal with emergencies. In this paper, we tried many existing models, including XGBoost, LightGBM, Seq2Seq, CNN-LSTM, and so on to predict the sum of transaction for future three months. The models are conducted based on the data from 2018 to 2021 given by Rakuten Card Company. EDA is employed in the feature engineering part to dig deep into the data, leading to the relatively useful features and watershed time period. We also collected some other related information, such as state of emergency policy, and consumer confidence, hoping to improve the accuracy. Cross validation has been used to compare the performance of these models. Some adaption and optimization has been made to the models with relatively better performance, and we combined those models together in order to show the best performance.

## 1. Introduction

Time series forcasting is one of the most classic theme in regression problem and a large number of successful models have been specially developed to solve this problem such as SARIMA, Prophet and so on. Nevertheless, the long-term forcasting is still a challenging task for those kind of models due to the error accumulation. In this paper, we explore some models to exmaine their performance on 3-months transaction forcasting problem and propose the comprehensive model with better performance and expandability which means it can absorb more new information.

The remaining of this paper is organized as follows: Section2 gives some details about data given by Rakuten. Section3 introduces the exploratory data analysis for transaction and some other open-source data, showing some insights of how time period correlate with the sum of transaction. Section4 firstly introduces processed data then gives detailed struture and mechanism of models. Section5 shows experiment results of proposed models then predict future 3 months both in daily and weekly step size. Section6 concludes the paper, discusses some insights based on experiment including the advantages and disadvantages of deep neural network.

## 2. About the data

The data was provided by Rakuten Card company consisted of Cashing (Personal loan) aggregated by daily with following mesh : sex, age, income, occupation, household status, family status, prefecture, and flag if home prefecture is the same as workplace prefecture. The time span of data is between 2018-01-01 until 2021-09-30. Here the daily sum of cashing (Cashing Sum) is the target feature that we would like to analyze and predict.

## 3. Exploratory Data Analysis (EDA)

### *3·1 Seasonal Decompose and Autocorrelation*

#### § 1 Seasonal decomposition

One of common preliminary analysis for time-series data is by checking its **trend**, **seasonality** and **residual** [Pourahmadi 01]. To determine trend and seasonality, the data can be fit with a function such as polynomial function $f(x) = \sum_{i=0}^{k} b_i x^{i+1}$ here k is the degree of the polynomial and b is the coefficient we want to find [Auffarth 21].To find the trend then the whole data is fit meanwhile for seasonality is over a cyclic period. As for the residual, it is simply difference between the data and trend or $y_t = x_t - s_t$ with $y_t$ as residual point, $x_t$ as data point and $s_t$ is the trend point.

To implement it, seasonal_decompose module within stats model package was used in this task which conveniently produce the decomposition. Period of seven was chosen to represent a week cycle. The result can be seen in **Figure 1** below. Based on the plot, we found that the data is relatively noisy and having a weekly cycle. The weekly cycle is one of the important insight here which means the daily cashing sum trend repeated over the week and could be related to the general behaviour of the credit users over the week.

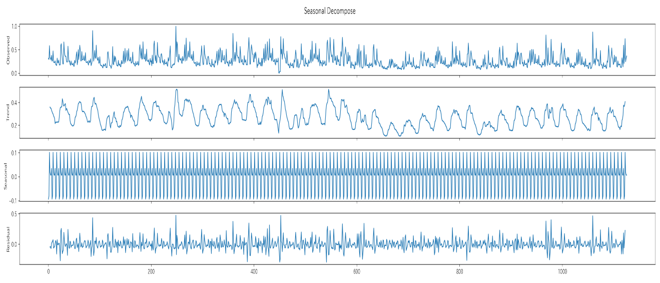


**Fig. 1** Seasonal decompose, (top) is the raw data (second) is the trend (third) is the seasonal and (bottom) is the residual.

#### § 2 Autocorrelation

Another method to see the pattern in the time-series data is by using **autocorrelation**. Autocorrelation is the correlation of the data with a lagged version of itself. Autocorrelation for given dataset $x_t$ can be denoted by $\rho_k = \frac{Cov(x_{t+k}, x_t)}{\sqrt{Var(x_{t+k})Var(x_t)}}, k = 0, \pm 1, ....$ Meanwhile, **partial correlation** is the correlation between $X_1$ and $X1 + k$ after eliminating the effect of the intervening values $X_2, ...X_k$. Thus, both methods can be used to see the patterns in the data. To implement it acf and pacf module within statsmodel package were used with lags 7 days and the plot can be seen in **Figure 2**.

From the plot we could see there are several repeated peaks over some periods. This strengthen the hypothesis of weekly periodicity in from seasonal decomposition. Moreover, the repeated peaks can also be seen for monthly period. This is also indicated that the general behaviour of the user credits over the months affect the daily cashing sum data. From this analysis one of possible inference is the feature date such as weekdays, weekends or payment day may have a strong effect to the data. Hence, feature date can be considered as one of feasible feature for prediction.

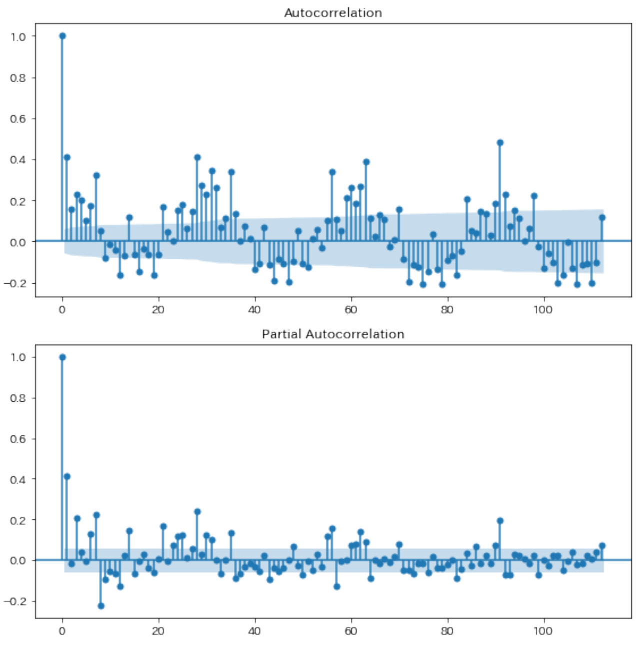


**Fig. 2** Autocorrelation and partial autocorrelation of cashing sum

### *3·2 Analysis of data over COVID-19*

#### § 1 Cashing sum versus COVID-19 infection rate

The global pandemic of COVID-19 has been greatly affected many people and goverment of many countries have been responded the situation with restriction measures. Hence, it is also affecting to the credit users' behaviour of the cashing collectively. **Figure 3** shows the plot of daily cashing sum with daily COVID-19 infection rate in Japan. Here, the cashing sum has been smoothed over 7-days rolling period and the infection rate was obtained from NHK news data.

As can be seen from the plot, red line is the smoothed cashing sum data while the green one is daily infection rate in Japan the data had been normalized to ease the comparison. Qualitatively, one can see that there was sudden decrease when the first case of COVID-19 occured in Japan around April 2020 and it continued to decrease until July 2020. However, after that the cashing sum data weren't affected greatly even though the cases were increasing throughout the country over the time. Since, it's hard to conclude what happened with the data qualitatively then quantitative approach must be used.

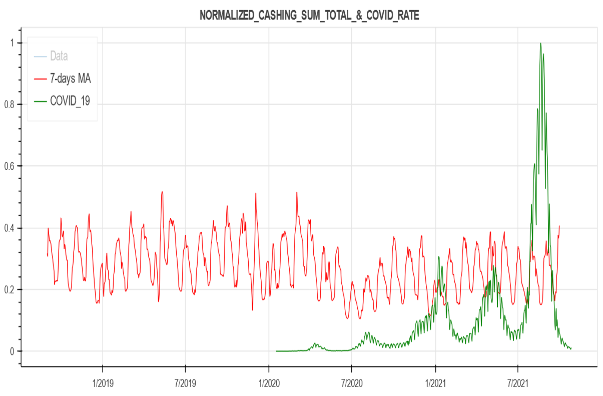


**Fig. 3** Normalized daily cashing sum versus daily COVID-19 infection rate in Japan

## § 2 Change Point Detection and Analysis

To quantitatively observe the trend over period of time, **change point detection** (CPD) was employed. CPD is one of unsupervised machine learning method for time series and related with anomaly detection. The idea is to find the points where the state changes over the time. Therefore, the time-series data could be divided into segmentation and for each set index $T = t_1 ... t_k$ for each segment, the best possible segmentation $T$ are searched according to quantitative criterion $V(T, x)$ that must be minimized [C. Truong 20].

CPD was used by using a package named *ruptures* in which KernelCPD = 'Linear' was used for 10 break points. Resulting in 11 window-time in which the state or average of each segmentation detected to be changed. From there, the change of daily cashing sum can be quantitavely compared with the situation happened related with COVID-19. Other than daily infection rate data, the state of emergency (SOE) of Tokyo was also used to analyze the effect of SOE to the cashing sum data. The plot can be seen in **Figure 4**, here the data had been normalized. Possible explanations are also given.

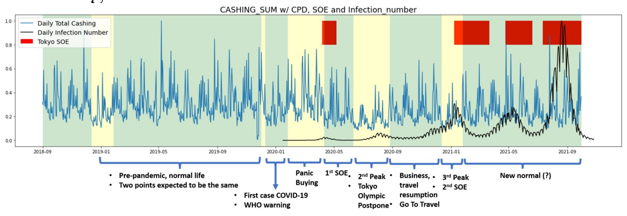


**Fig. 4** Change point detection on cashing sum data alongside daily infection rate and Tokyo SOE. The possible explanation of each window time are given in the bottom text

One particular insights of this analysis is that after the second SOE or the second peak, there were no changes to the state although we have seen more peaks and the highest peak at the end of the given dataset. Meanwhile, at the beginning of COVID-19 case and the first SOE we saw drop of the trend. It also happened during the second peak and when Japan's Olympic 2020 was postponed. Therefore, one of insights from this is that the Cashing sum may not directly related to the infection rate or the state of emergency but it was more to the psychological state that happened during the time because of sudden event or disruption from COVID-19 impact. This is also may related with the economic situation in Japan which later will be explained in the next subsection. In addition, this insight also can be used to build a feature based on psychological state from the data.

### 3·3 *Building feature from CPD analysis*

As explained in the last subsection, CPD segmented the dataset into 11 window according to each state. In addition, it's more likely the change of cashing sum happened due to the psychological state happened during the disruption events. However, this pyschological state is a qualitatively feature therefore need to be change to quantitative feature by assigning some values according to its average in the window. Hence, psychological state are proposed here as one of features that affect the cashing sum and can be used for prediction. It's similar to how consumer index are being quantitative using satisfaction index (e.g. 5: very satisfied, 4: somewhat satisfied,...). The average of each window (avg), the difference between average (diff) and the psychological states are compiled in **Table 1**.

**Table 1** Features from CPD analysis

| Windows | Avg | Diff. | State |
|---|---|---|---|
| 1 | 0.31 | 0 | 0 |
| 2 | 0.22 | -0.09 | 0 |
| 3 | 0.32 | 0.09 | 0 |
| 4 | 0.39 | 0.08 | 0 |
| 5 | 0.24 | -0.15 | -3 |
| 6 | 0.34 | 0.10 | 2 |
| 7 | 0.23 | -0.10 | -2 |
| 8 | 0.17 | -0.07 | -1 |
| 9 | 0.26 | 0.08 | 1 |
| 10 | 0.18 | -0.07 | -1 |
| 11 | 0.26 | 0.08 | 1 |

The states are assigned according to the difference of average between each state. 0 were assigned to windows 1-4 as it represents the old normal (we assume there was no disruption at all). State 1 for diff. score diff $< 0.1$, state 2 for $0.1 <$ diff $< 0.15$ and $0.15 >$ diff. The sign of positive and negative will follow the diff. value. The advantage of using this feature is the user can predict the cashing sum based on the possible scenarios that will likely happens in the future based on COVID-19 related-events. For example, the resumption of business and travel relaxation from government will likely increase the cashing sum as shown in windows 8 in the past therefore we could assign the ’state : 1’ as the feature, or else the new variants of COVID-19 are emerging and disrupt the situation again then we could assign the ’state :-1’. Therefore the prediction will be closer to the situation as shown in the **Figure 12** where the lines of ’good scenario’ and ’bad scenario’ are separated based on the situation.

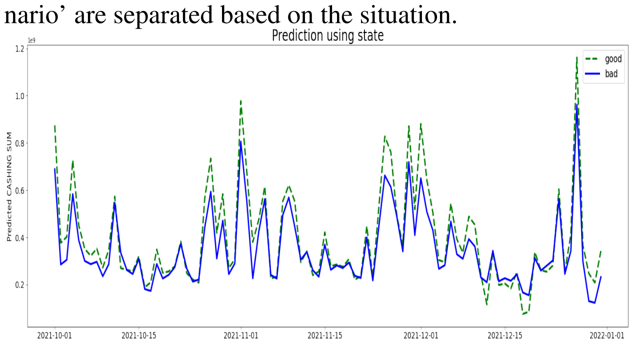


**Fig. 5** Future prediction based on psychological state according to CPD analysis, model : XGBoost along with feature date

However, the usage of this feature need much cautious because (1) it is only based observation of a little repeated events hence this feature itself is not robust, (2) needs experts’ opinion on deciding the state and other data, (3) only rely on COVID-19 situation but not with other aspect. Therefore, as in the current stage it’s still not recommended using this as a reliable feature and need to be explored more in the future.

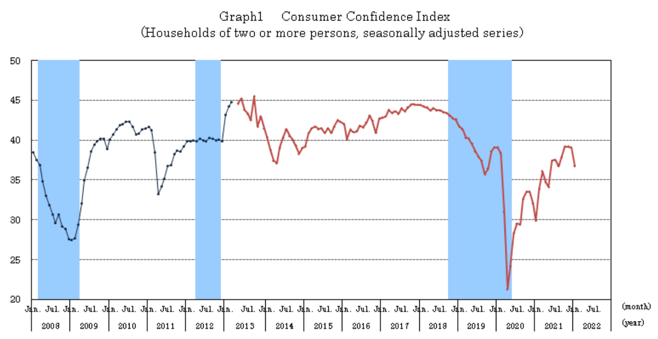
## 3·4 *Exploring economic feature*

COVID-19 pandemic led to temporary disruptions and restrictions in economic activities. In order to understand how different governmental measures have disrupted the way people of Japan work, learn, socialize and consume we aim to explore data on economic drivers that helps us understand changing consumer behavior within Japan.

### §1 Consumer Confidence Index(CCI)

Consumer confidence index illustrates how people perceive their financial prospects and the overall economy state. A higher confidence level indicates that consumers are willing to spend more, while a low confidence level echoes uncertainty about the future and unwillingness to spend. Consequently, consumer confidence will be reflected in saving and spending activities.

Seasonally adjusted consumer confidence surveys conducted by Japanese statistics agency every month and results published by the end of the month or in the beginning of the next month:



Notes:
1) The recession periods are shadowed.
2) Until March 2013 the survey was conducted by the"direct-visit and self-completion questionnaires method".
3) From April 2013 the survey method was changed to the "mail survey method" and the sample households number has increased from 6,720 to 8,400. Accordingly there is a discontinuity between March and April in 2013.
4) From October 2018 the survey method was changed to the "method using mail and online together". However, the first month of the survey is carried out by the"direct-visit and self-completion questionnaires method" (the survey method in the first month is the same as before).

**Fig. 6** Consumer Confidence Index from June 2009 to January 2022. Source: https://www.esri.cao.go.jp/en/stat/shouhi/shouhi-e.html

A strong dip in consumer confidence on the economic situation during COVID-19 pandemic is visible and have not reached pre-pandemic levels yet. In the beginning of pandemic CCI of Japan was even lower than it was at its worst point after the 2008 financial crisis. The sharp deterioration in 2020 highlights the insecurity and uncertainty that Japanese consumers had regarding situation in the economy and clearly left an imprint on consumption. If you impose CCI graph on credit card monthly caching data provided by Rakuten Bank you can clearly see correlation.

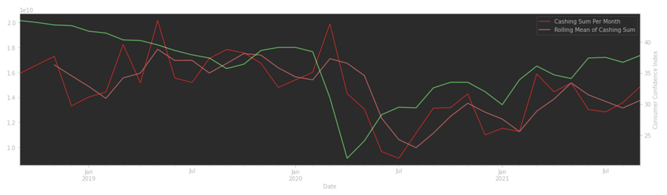


**Fig. 7** CCI(green line, households with 2 and more people), Monthly Cashing Sum(red line, all), Rolling Mean of Monthly Caching Sum(pink line, all)

Since easiest obtainable CCI data is limited to households with 2 or more people, we also dropped single-person household data from Monthly Cashing Sum:

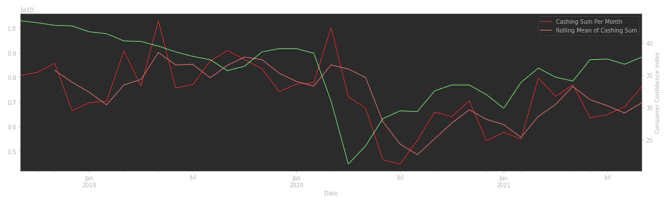


**Fig. 8** CCI(green line, households with 2 and more people), Monthly Cashing Sum (red line, households with 2 and more people), Rolling Mean of Monthly Caching Sum(pink line, households with 2 and more people)

Since we can see that CCI and Cashing Sum correlates with some sort of time-lag, we tried to calculate cross-correlation with different time shift: from 1 to 5 months just in case. We found out that **cross-correlation** between CCI and Cashing Sum **is 0.67 with 2 months lag** and cross correlation between CCI and Rolling Mean of Caching Sum **is 0.78 with 3 months lag**. This level of correlation with time lag makes CCI perfect candidate as a feature.

### § 2 Labor force statistics

Statistics agency of Japan provides monthly data about overall labor force status, employment status by industry, unemployment data, etc. The scope of labor data provided by statistics agency was quite huge and impossible to explore within this course. And unfortunately labor status stats are provided with overall 2 months delay, which makes it impossible to use for predictions. Nonetheless, we explored some statistical data regarding labor participation:

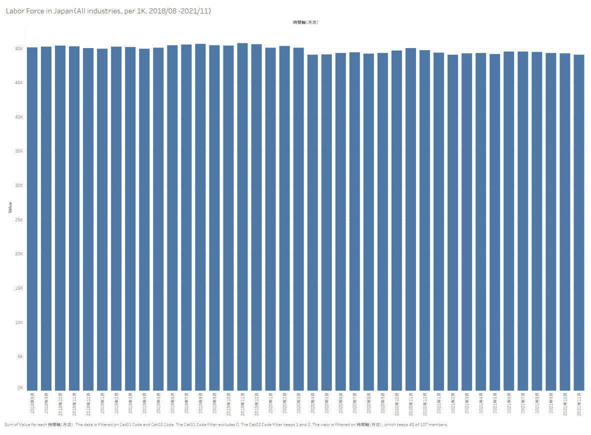


**Fig. 9** Labor Force in Japan in All Industries from August 2018 to November 2021

As we see, due to the conservative labor market in Japan, there are very small fluctuations in the actual labor force data. This trend is even more typical for labor force participation among men:

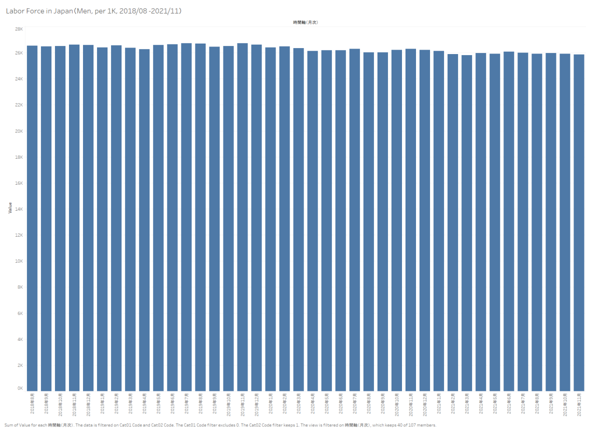


**Fig. 10** Labor Force in Japan in All Industries Among Men from August 2018 to November 2021

But if we narrow down the labor force data to the most vulnerable economic categories, such as women:

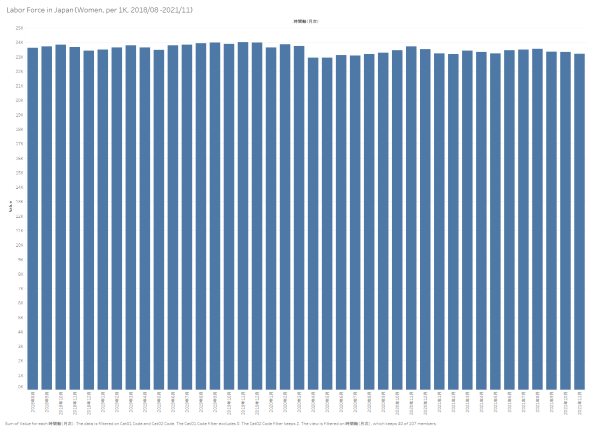


**Fig. 11** Labor Force in Japan Among Women from August 2018 to November 2021

- or part-time workers:

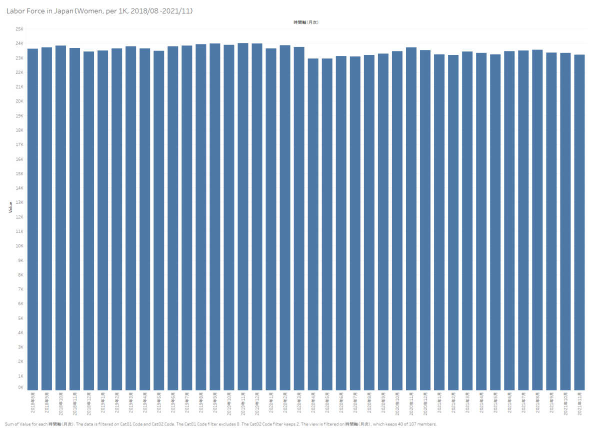


**Fig. 12** Part-time Labor Force in Japan in All Industries from August 2018 to November 2021

- then we clearly start to see fluctuations related to the economic situation during COVID-19 pandemic in Japan. While the actual employment data do not have high correlation with consumer behavior, some of it(such as employment among women or part-time laborers) can be used as and indicator of economic situation in the country. But unfortunately the 2 months delay with which the data provided is impossible to overcome.

## 4. Model

### *4·1 Prepared Data*

There are many kinds of data have been used and each model may have different choice among these data. For a consistent representation, we first process data and name different processed data with a responding alias. For the data given by Rakuten Card company, we split it into two parts: 1. Users' information including age, gender, family condition and so on. 2. our forecasting target which is daily cashing sum. 3. transaction information such as the prefecture where this behavior happened. Since this data is given by daily, we then develop date information based on time series, consisting of date, the day of the week, quarter, month, year, day of year, day of month, week of year, and week of month. the holiday information has also been considered, including both Japanese holiday and Western Holiday. Otherwise, we utilize the open-source data including the state of emergency policy information (the detailed information will be based on both the date and the prefecture where the transaction behavior happened) and consumer confidence information. All the categorical data will be preprocessed as one-hot vector.

The data from Rakuten can be preprocessed to generate 88 features for one day record, including one feature for daily cashing sum plus 87 features which are the counting of frequency for each item appeared in one day, e.g., for column "SALARY_CD", there is one item(choice) called " 400 万以上 600 万未満 ", then we count the number of transactions satisfying this choice in one day and make it one feature for that daily data. We extract all these features to fully utilize all the information contained in the provided transaction data.

The other 6 features are from Japan consumer confidence. From:

https://www.esri.cao.go.jp/en/stat/shouhi/shouhi-e.html

We used " seasonally adjusted series " version of consumer confidence. From:

https://www.esri.cao.go.jp/en/stat/shouhi/shouhi2.xlsx

The 6 features are:

(1). Consumer Confidence Index,

(2). Component indicators of Consumer Confidence Index_Overall livelihood,

(3). Component indicators of Consumer Confidence Index_Income growth,

(4). Component indicators of Consumer Confidence Index_Employment,

(5). Component indicators of Consumer Confidence Index_Willingness to buy durable goods,

(6). Other consumer confidence indicator_Asset value.

Since these data is given monthly, to get daily features, We use the same values for these consumer confidence features in the same month. Then to get weekly features, We calculate the average value for 7 days (one week may span two months with different monthly consumer confidence for 7 days). Using these features makes the model converge faster and leads to better results.

To summarize, there are 6 kinds of data including: Users' information, cashing sum, transaction information, date information, the state of emergency policy information, consumer confidence information.

## 4·2 *Proposed Models*

### § 1 XGBoost

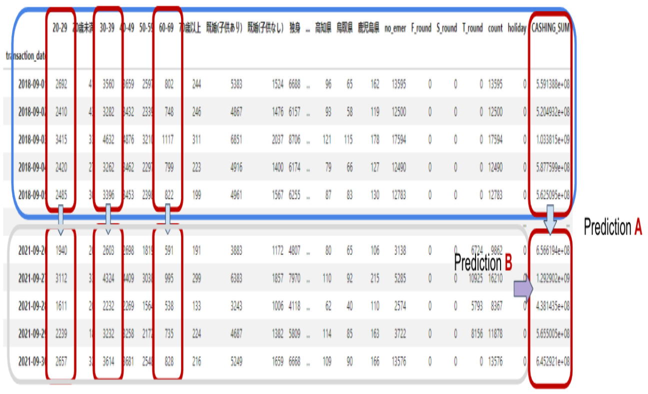
XGBoost stands for eXtreme Gradient Boosting, which is an implementation of gradient boosting decision tree algorithm. This algorithm was designed to compute fast and save memory resources, which supports oth regression and classification predictive modelling problems. Therefore, this method can well fit the goal of predicting transaction amount in the future, given the date and some other feature information.

The data I'm going to use in this method will include the date information, the user's information, the transaction information, and the state of emergency policy information (the detailed information will be based on both the date and the prefecture where the transaction behavior happened).All of them including the cashing sum has been accumulated daily.

As in this paper, since the goal is prediction, all the information we can get for sure in the future is the date information. So, we should make full use of the date. The first prediction will be conducted based on the date information. The input will be the date information and the training output is daily cashing sum.

Such prediction is a basic one, which works under normal conditions since the transaction is a relatively regularized behavior. But when some emergency happens, such as COVID-19, people may feel uneasy and tend to transact more money from bank. Such behavior will make the transaction amount fluctuate, and may cause the prediction under normal condition not reliable. Therefore, it is rather necessary to improve this model to make it more robust under emergency conditions.

Here, in order to make full use of the user features and the transaction behavior features, it comes to me to get a prediction feature matrix.



| transaction_date | 20-29 | 20歳未満 | 30-39 | 40-49 | 50-59 | 60-69 | 70歳以上 | 既婚(子供あり) | 既婚(子供なし) | 独身 | ... | 高知県 | 鳥取県 | 鹿児島県 | no_emer | F_round | S_round | T_round | count | holiday | CASHING_SUM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2018-09-0 | 2692 | 4 | 3560 | 3659 | 2597 | 802 | 244 | 5383 | 1524 | 6688 | ... | 96 | 65 | 162 | 13595 | 0 | 0 | 0 | 13595 | 0 | 5.591388e+08 |
| 2018-09-0 | 2410 | 4 | 3282 | 3432 | 2339 | 748 | 246 | 4867 | 1476 | 6157 | ... | 93 | 58 | 119 | 12500 | 0 | 0 | 0 | 12500 | 0 | 5.204932e+08 |
| 2018-09-0 | 3415 | 3 | 4632 | 4876 | 3210 | 1117 | 311 | 6851 | 2037 | 8706 | ... | 121 | 115 | 178 | 17594 | 0 | 0 | 0 | 17594 | 0 | 1.033815e+09 |
| 2018-09-0 | 2420 | 2 | 3262 | 3462 | 2297 | 799 | 223 | 4916 | 1400 | 6174 | ... | 79 | 66 | 127 | 12490 | 0 | 0 | 0 | 12490 | 0 | 5.877599e+08 |
| 2018-09-0 | 2485 | 3 | 3396 | 3453 | 2390 | 822 | 199 | 4961 | 1567 | 6255 | ... | 87 | 83 | 130 | 12783 | 0 | 0 | 0 | 12783 | 0 | 5.625095e+08 |
| 2021-09-2 | 1940 | 2 | 2603 | 2698 | 1819 | 591 | 191 | 3883 | 1172 | 4807 | ... | 80 | 65 | 106 | 3138 | 0 | 0 | 6724 | 9862 | 0 | 6.566194e+08 |
| 2021-09-2 | 3112 | 3 | 4324 | 4409 | 3038 | 995 | 299 | 6383 | 1857 | 7970 | ... | 110 | 92 | 215 | 5285 | 0 | 0 | 10925 | 16210 | 0 | 1.292902e+09 |
| 2021-09-2 | 1611 | 2 | 2232 | 2269 | 1564 | 538 | 133 | 3243 | 1006 | 4118 | ... | 62 | 40 | 110 | 2574 | 0 | 0 | 5793 | 8367 | 0 | 4.381435e+08 |
| 2021-09-2 | 2239 | 1 | 3232 | 3258 | 2172 | 735 | 224 | 4687 | 1382 | 5809 | ... | 114 | 85 | 163 | 3722 | 0 | 0 | 8156 | 11878 | 0 | 5.655005e+08 |
| 2021-09-3 | 2657 | 3 | 3614 | 3681 | 2548 | 828 | 216 | 5249 | 1659 | 6668 | ... | 109 | 90 | 166 | 13576 | 0 | 0 | 0 | 13576 | 0 | 6.452921e+08 |

**Fig. 13** Prediction structure

Based on the first kind of prediction, I predict every features in the future three months. After predicting every features, a predicted feature matrix can be made which can be used as the test input. So in the training part, all the features can be used as input, and the cashing sum will be the training output. In this way, the second prediction can be made, which will be the improved one.

## § 2 Seq2Seq

Sequence to Sequence model (we call it Seq2Seq in following pages) is widely used in NLP field such as machine translating task, training models to convert sequences from one domain (e.g. sentences in English) to sequences in another domain (e.g. the same sentences translated to French). it is good at processing the Unequal input and output which means it fit our forecasting problem well since we can use long-term historical data to predict future 3 months cashing sum without caring the length of input sequence. Seq2Seq mainly consists of two parts: 1. "encoder", it processes input sequence and return its own hidden state. Note that we discard the outputs of the encoder, only recovering the state. This state will serve as the "context", or "conditioning", of the decoder in the next step. 2. "decoder", it is trained to predict the next characters of the target sequence, given previous characters of the target sequence. Importantly, the decoder uses the last hidden state which is also output from the encoder as initial state, which is how the decoder obtains information about what it is supposed to generate. Effectively, the decoder learns to generate targets[t+1...] given targets[...t], conditioned on the input sequence. Above these, we try to use attention mechanism during training decoder aims to give a better understading on long-term sequences. Attention mechanism will focus on all encoder's output and gives a responding weight instead of using the last output from encoder.

The data I'm going to use in this method will include the date information, the consumer confidence information and cashing sum. The detailed struture of Seq2Seq is 2-layer GRU both for encoder and decoder with the size of hidden state is 200.

In this model, Our input will split into encoder'input and decoder'output. As mentioned before, since the goal is future forecasting, we want encoder'input is the historical data and decoder's input is the information we can get in the future. For encoder, we use last 180 days cashing sum, date information and consumer confidence as input. For decoder, we use date information about future 90 days as input and the output is 90 days cashing sum in the future.

## § 3 CNN-LSTM + MLP Decoder

The input data for my model is with shape (samples, time steps, features). One sample is one time sequence, and one time step represents one week. There are 94 features in my input data for the model. 88 features from preprocessed Rakuten data and 6 features from Japan consumer confidence.

My model consists of two predictions, prediction for weekly cashing sum (CNN + LSTM) and prediction for daily cashing sum (MLP Decoder). I use CNN plus LSTM layers to build a model for weekly cashing sum prediction, the TimeGAN[Yoon 19] is used to generate more weekly training data to improve the accuracy of the model since the amount of training data is very limited. Then based on weekly prediction, I build an MLP decoder to transfer weekly cashing sum to daily cashing sum as the daily prediction.

The reason for doing so is that using LSTM based model directly to make daily prediction is very hard since the LSTM cannot well remember a sequence longer than 150 200 time steps. To make a prediction for future 3 months (90 days) and using the training data contains past 9 months (270 days), the number of time steps for each sample (sequence) is 270 which is too long to be remembered by common LSTM layer. As a result, the daily prediction behaves poorly by applying LSTM directly in my model. So, I tried make weekly prediction first, in this case, 9 months roughly equals to 39 weeks, which makes it appropriate to apply LSTM model for a prediction based on training data with only 39 time steps for a sample (sequence). As a result, the weekly prediction can be much more accurate than daily prediction. Then based on this accurate weekly cashing sum prediction, I use an MLP decoder to get daily prediction. The following is the details for my model.

To make weekly prediction, I use CNN layers with their dilations set to 1, 2, 4, 6, 8, 12, 16 to figure out the time series patterns from short term (every two weeks) to long term (nearly every 4 months). Then after going through all CNN layers, the input data turns to the data with the same number of samples and time steps but different feature numbers (60 features), then each sample in training set with shape (39, 60), (time steps, features) is fed to LSTM layers with depth 3. The last TimeDistributed layer output the results with shape (samples, time steps, predictionForNext13Weeks). So, for each time step, the model outputs the next 13 weeks (time steps) cashing sum prediction and the prediction for last time step for one sample in the training set is the prediction for future 13 weeks. The reason to use " return_sequences=True " in each LSTM layer and TimeDistributed layer here is to backpropagate errors at each time step instead of just backpropagate the errors after one sample containing all time steps. In this way, more errors will be backpropagated to achieve better training results.

Directly applying such CNN-LSTM model for weekly prediction can generate much more accurate prediction compared with applying CNN-LSTM model for daily prediction. To further improve the weekly prediction, TimeGAN may be used to generate more training data (originally, there are only 77 samples in training set). The original TimeGAN implantation by the authors of that paper is based on Tensorflow 1.x which is outdated. So, I found another TimeGAN implementation in this repository:

https://github.com/ydataai/ydata-synthetic
and use it to generate more training data.

To use TimeGAN, I use only original training set and validation set as TimeGAN input, the original testing set is not visible for TimeGAN. The training set and validation set is regrouped so that each sample for TimeGAN input data contains 39 weeks + 13 weeks, so, for input data (samples, time steps, features), the time steps is 52 for one TimeGAN input data. After the new samples with the same shape (time steps, features) generated by TimeGAN, each new sample is transferred to one sample for input X with shape (39, features) and another sample for output y with shape (39, 13), 13 cashing sum prediction for the following 13 weeks for each time step. So that such transferred sample can be fed to CNN-LSTM model.

However, there are some problems when applying TimeGAN and the result is not stable. First, the best number of epochs to train TimeGAN is unknown here and the way to evaluate the quality of generated training data is also unknown for me. As a result, if the generated training data is good enough to capture the features in training set than the results tend to be better than prediction without TimeGAN, otherwise the results will get worse.

To make daily prediction based on previous good weekly prediction results, I built an MLP decoder to transfer one week cashing sum along with time features like the which week in current month, which month in one year and whether this week contains any holidays for each day. I use one-hot encoding to encode all these information. So for which week in current month I got 6 features (one month can at most span 6 weeks), for which month in one year I got 12 features (one year contains 12 months), and for holiday information for one day, I use 4 types: not holiday, a day belong to a long holiday for more than 7 days (From Dec. 24 to Jan. 3 or from April 29 to May 5), a day belong to a short holiday (a holiday for 3 consecutive days including weekends) and an independent one-day holiday in weekdays. Each day in one week is encoded by 4 features as one-hot encoding so I got 28 features for holiday. As a result, I use 6+12+28+1(cashing sum) = 47 features for each week and the MLP decoder outputs 7 cashing sum distribution in 7 days in one week.

### § 4 Stacking

Since the tree model shows excellent performance in this task, it is natural to use a stacking strategy to further boost the model. In model stacking, we do not use a single model to make predictions – instead, we use several different models to make predictions, and then use those predictions as features of higher-level metamodels. It works especially well for different types of low-level learners, all of whom can contribute different strengths to the meta-model. In our stacking model, we will use a series of intermediate models to make non-leakage predictions on our training data, and then their results are fed as features into the meta-model to obtain the final predictions.

Specifically, we selected XGBoost, LightGBM and CatBoost as the low-level learners of the stacking model, and linear regression model as the meta-model. Since using feature unknown in the future is complicated and time consuming on the tree-based model, we only selected features that are known in the future. That is features such as dates, holidays, and the state of emergency declarations are used as inputs to the model, and the cashing sum is used as the output of the model. The entire data was used to train the model, which was then used to predict the cashing sum for the next 90 days.

## 5. Experiments

### *5·1 Setup:*

We apply nested cross validation to adjust hyperparameters, i.e., 6 training and test sets were split to validate the model (test: 2021-02-03∼2021-05-03; 2021-03-05∼2021-06-02; 2021-04-04∼2021-07-02; 2021-05-04∼2021-08-01; 2021-06-03∼2021-08-31; 2021-07-03∼2021-09-30; train: the data before each test set is the corresponding train set). The effectiveness of the model will be measured by the average score of the 6 test sets. Experiments are performed on a CPU cluster server or a GPU cluster server in Graduate School of Information Science and Technology.

### *5·2 Metrics:*

We evaluate the overall performance based on MAPE (Mean Absolute Percentage Error).

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{\hat{Y}_i - Y_i}{(\hat{Y}_i + Y_i)/2} \right| \qquad (1)$$

where $n$ is the number of samples, $Y$ and $\hat{Y}$ are the ground-truth data and predicted data.

## *5·3 Performance*

**Table 2** Overall Performance Evaluation

| Method | Metrics |
|---|---|
| | MAPE |
| SARIMA | 45.92% |
| STL | 40.40% |
| Prophet | 36.82% |
| Seq2Seq | **22.25%** |
| CNNLSTM(weekly) | **7.25%** |
| CNNLSTM+MLP(daily) | 17.90% |
| XGBoost | 14.05% |
| LightGBM | 16.02% |
| CatBoost | 17.06% |
| Stacking | **13.43%** |

The performance of all models are shown in Table 2. We added some statistical models (i.e., SARIMA, Seasonal and Trend decomposition using Loess (STL) and Prophet) as baseline models. However, these statistical models show very poor results on our prediction task due to the inability of these models to capture the complex periodic and trending patterns within the data. For the models presented in Section 3, we can find that:

### § 1 XGBoost

There will be 2 kinds of prediction, including the classic one, and the improved one.

Take 2 periods as an example. When we tried to predict from 2020/04 to 2020/07, which covers the first round of declaration of emergency in Tokyo. The MAPE will be 49.38, which has little credibility. We can see that at the beginning part of COVID-19, people did change their transaction behavior greatly. If we we predict from 2021/07 to 2020/09, which covers the third round of declaration of emergency, the MAPE will be down to 33.53. Despite MAPE is still relatively high, people may gradually get used to Corona, or our model may have learnt something from the data before, which includes the first year of COVID-19.

In order to compare the result between the classic XGBoost and the improved one, we predict the same periods based on the improved method. In the first round of declaration, the MAPE was down to 13.96. With features added, the improvement is very obvious compared to 49.38. And these features can make the prediction more robust to emergency condictions. As for the third round of emergency, MAPE was down to 12.71. As Corona's influence on people has been decresed, the prediction performance will be closer to the prediction under normal conditions. Besides, under normal conditions, the performance with or without features added are relatively the same, which reveal that under normal conditions, transaction behavior will be more related to date information.

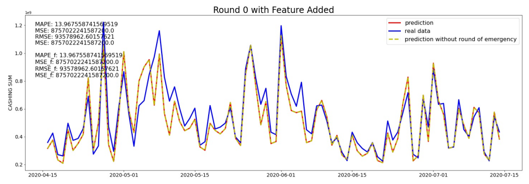


**Fig. 14** Round 0 with Feature Added (2020)

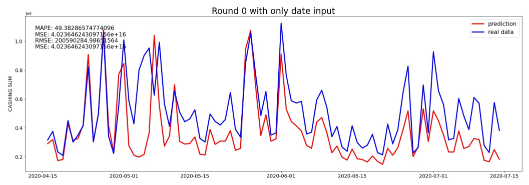


**Fig. 15** Round 0 with only date input (2020)

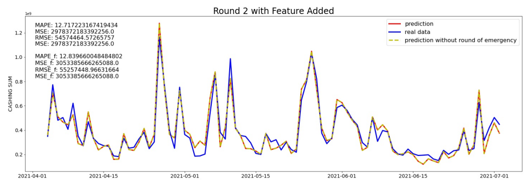


**Fig. 16** Round 2 with Feature Added (2021)

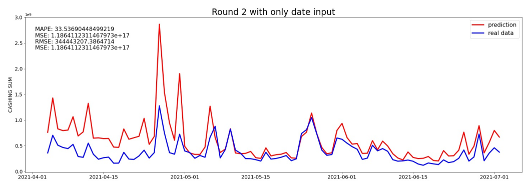


**Fig. 17** Round 2 with only date input(2021)

### § 2 Seq2Seq

There are 2 prediction results, including the basic Seq2Seq and anther Seq2Seq with attention. Note that both of them have the same structure and their only difference is whether it will use attention mechanism.

First we test different features to decide models' input, and results show that except date information and historical cashing sum data, consumer confidence is a useful feature that strongly correlated with cashing sum. Basically it can decrease the MAPE around 2.

Then we show 2 prediction results. In order to compare the result between the basic Seq2Seq and the improved one, we predict the same periods to show their performance and we found there is a large gap between their performance. In the 0-th round of testing, the MAPE of basic Seq2Seq is 36.36. With attention mechanism added, the MAPE downs to 20.97 which is a very obvious improvement compared with 36.36. Above that, the curve catch the peak and valley well which means it can accurately find the future tendancy and periodicity of cashing sum. The overall MAPE of Seq2Seq with attention is around 22.

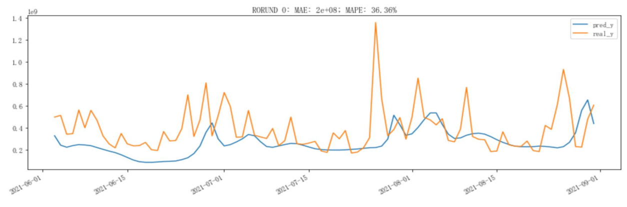


**Fig. 18** Round 0 results(basic Seq2Seq)

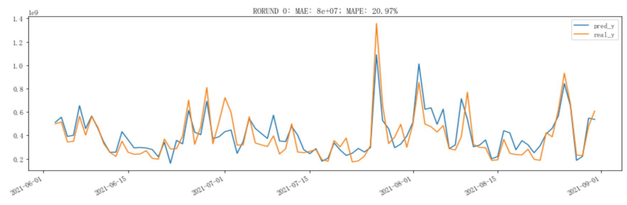


**Fig. 19** Round 0 results(Seq2Seq with attention)

## § 3 CNNLSTM + MLP Decoder

The evaluation is done by using the last 13 weeks(2021-07-02, 2021-09-30) cashing sum as testing data.

The prediction for weekly cashing sum is accurate, the MAPE can reach to 7.25% as shown in Figure 20, while the prediction for daily cashing sum is not good due to the model, MLP Decoder, to transfer weekly cashing sum to daily cashing sum needs more improvement, the MARE is 17.90%, as shown in Figure 21.

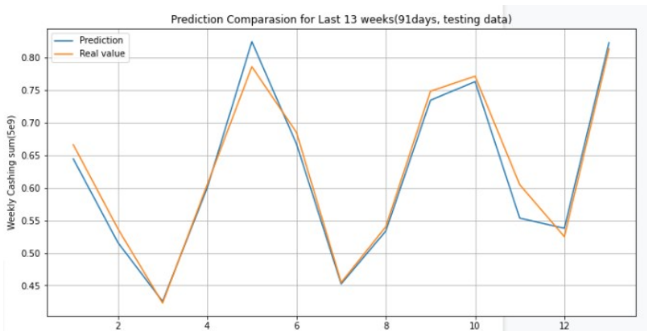


**Fig. 20** Weekly model(CNN-LSTM)

The prediction for future 13 weeks (2021-10-01, 2021-12-30) is shown in Figure 24.

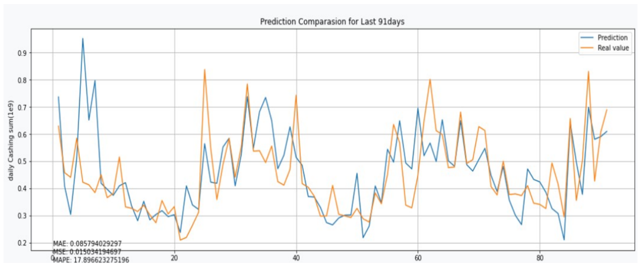


**Fig. 21** Daily model(CNN-LSTM + MLP)

## § 4 Stacking

The stacking model obtained the best results due to that the tree model is successful in capturing the complex periodic features in the data. Figure 22 show the result of one test set obtained by stacking model. We can observe that the model is very accurate in predicting the weekly trends and extreme value points.

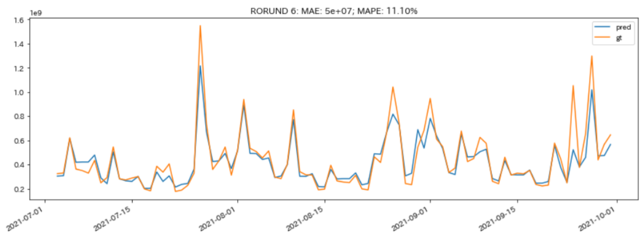


**Fig. 22** Round 5 for stacking model

## 5·4 *Prediction For Future 3 month*

Here is the daily and weekly prediction for future 3 month(2021.10∼ 2021.12).

## § 1 Daily Prediction

Daily predicton:

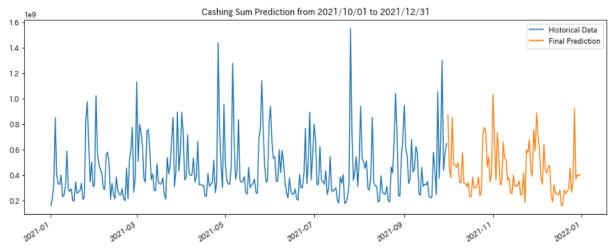


**Fig. 23** Daily prediction for the next 3 month (2021.10.1∼2021.12.31)

## § 2 Weekly Prediction

Weekly prediction:

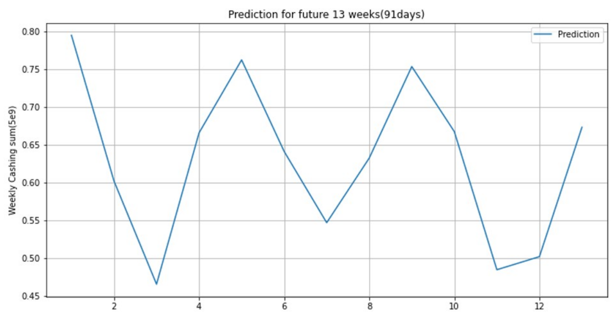


**Fig. 24** Weekly prediction for next 3 month (2021.10.1∼2021.12.30)

## 6. Conclusion

In this task, we used daily cashing sum as our target feature for analysis and prediction. According to the seasonal decompose and autocorrelation analysis, there are weekly and monthly periodicity in the data. CPD was employed to do the segmentation of the data combined with COVID-19 infection rate and SOE for analysis. It was found that the daily cashing sum was more likely disrupted due to the psychological effect from COVID-19 related events. Therefore, a feature from the analysis was built with the advantage of possible scenarios in the future but the usage needs to be cautious and need to be explored in the future.

As for the XGBoost part, the prediction has been improved a lot with features added, indicating this improvement can make the prediction more robust facing with emergencies. But since there's not enough time for parameter tuning, if given enough time for future parameter tuning, the prediction may reach a better accuracy.

For CNN-LSTM + MLP Decoder part, the weekly prediction is accurate since LSTM can well remember patterns in time series if the time step is not too long. But the MLP Decoder needs more improvement like adding more features to generate better daily prediction from weekly prediction and time features.

***Acknowledgments***

Special thanks to Mr. Kyosuke Tomoda and Mr. Takumi Yamaoka from Rakuten Card Co., Ltd for provided data and time spent participating in the discussion and to our TA Yuki Takazawa for his assistance and valuable suggestions.

## ♢ References ♢

[Auffarth 21] Auffarth, B.: *Machine Learning for Time-Series with Python*, Packt Publishing (2021)

[C. Truong 20] C. Truong, e. a.: Selective review of offline change point detection methods, *Signal Processing*, Vol. 167, No. 107299 (2020)

[Pourahmadi 01] Pourahmadi, M.: *Foundations of Time Series Analysis and Prediction Theory*, John Wiley Sons, Inc (2001)

[Yoon 19] Yoon, J., Jarrett, D., and Schaar, van der M.: Time-series Generative Adversarial Networks, *NeurIPS 2019* (2019)

## ♢ Appendix ♢

## A. Appendix

### Contributors' list

- Refaldi Putra
  - ○ Exploratory Data Analysis (EDA)
    - ∗Seasonal decompose and autocorrelation
    - ∗Cashing sum vs COVID-19 infection rate
    - ∗Change point detection analysis
  - ○ Feature Engineering
    - ∗Feature from CPD analysis
- Mergen Kungaa
  - ○ Exploratory Data Analysis (EDA)
    - ∗Economic feature exploration
    - ∗Cross-correlation between Consumer Confidence Index and Monthly Cashing Sum
    - ∗Labor force statistics
- Zihan Li
  - ○ Data preprocessing, model building (Model, Experiment)
    - ∗Preprocess data from Rakuten and consumer confidence to get input data and features for my model.
    - ∗Build CNNLSTM + MLP Decoder model to make weekly and daily prediction.
- Guanting Liu
  - ○ Data preprocessing, model building (Model, Experiment)
    - ∗Preprocess data from Rakuten to get input data and features for my model.
    - ∗Build Seq2Seq with attention model to make daily prediction.
- Yi Zhang
  - ○ Data preprocessing, model building (Model, Experiment)
    - ∗Preprocess data from Rakuten and other open-data to get input data and features for my model.
    - ∗Build XGBoost model to make daily prediction.
- Zekun Cai
  - ○ Designing experiments and Planning schedules
  - ○ Data preprocessing, model building (Model, Experiment)
    - ∗Designing the cross-validation framework.
    - ∗Building SARIMA, STL and Prophet model to make daily prediction.
    - ∗Building the simple version of Seq2Seq model to make daily prediction.
    - ∗Building Stacking model to make daily prediction.