# Predict Optimal Price for a Flash Sale Campaign

Ryan Mokarian, Jan. 20, 2023

## Table of Contents

## Question

A product manager plans to run a "flash sale campaign" for new users for one day. Eligible users will see a sale icon on the side of the screen with a countdown timer attached. When tapped on the icon, the offer's price and the content will be shown on an in-game pop-up window. If the user closes the popup and dismisses the warning message, both the pop-up and the offer icon vanish. That user has lost the opportunity to purchase a fantastic offer. The option is also lost if the timer expires without the user tapping on the icon.

In trying to optimise the **monetisation** performance, the product manager asked you to build a model that, using data **within 7 days**, predicts the optimal price for the flash sale for **each new user**. There are pre-designed sale offers at different prices, and the game will take the model output and show the offer closest to the predicted price.

You are given ~6 weeks of data (15th Dec - 24th Jan) immediately before the campaign, which will be run on 1 Feb. In the datasets are the following fields:

| FULLNAME | TYPE | DESCRIPTION |
|---|---|---|
| **BFGUDID** | STRING | unique BFG device ID; the user's first device they installed the game |
| **ACTIVITY_DATE** | DATE | partition field based on the date of user/device activity |
| **PLAY_SECONDS** | NUMERIC | the total valid play seconds for a user/device on the activity date |
| **PLAY_SESSION_COUNT** | NUMERIC | the number of valid play sessions for a user/device that have ended on the activity date |
| **APP_STORE** | STRING | platform where game is sold; for example 'itunes', 'google', 'amazon' |
| **COUNTRY_CODE** | STRING | ISO alpha 2 country code based on the IP address associated with the install; or, if missing, the country code provided by the SDK |
| **PRODUCT** | STRING | the sku for the IAP product |
| **CURRENCY** | STRING | the currency used for the transaction |
| **GROSS_USD_AMOUNT** | FLOAT | gross revenue in USD for the transaction |
| **PRICE_TIER** | FLOAT | the USD price for the bundle (i.e. normalized price) |
| **DAY_SINCE_INSTALL** | INTEGER | the number of days between the current and the day the user started to play |
| **MAX_PRICE_TIER** | FLOAT | the highest price tier the user purchased in the first 7 days |
| **AVG_PRICE_TIER** | FLOAT | the average price tier the user purchased in the first 7 days |

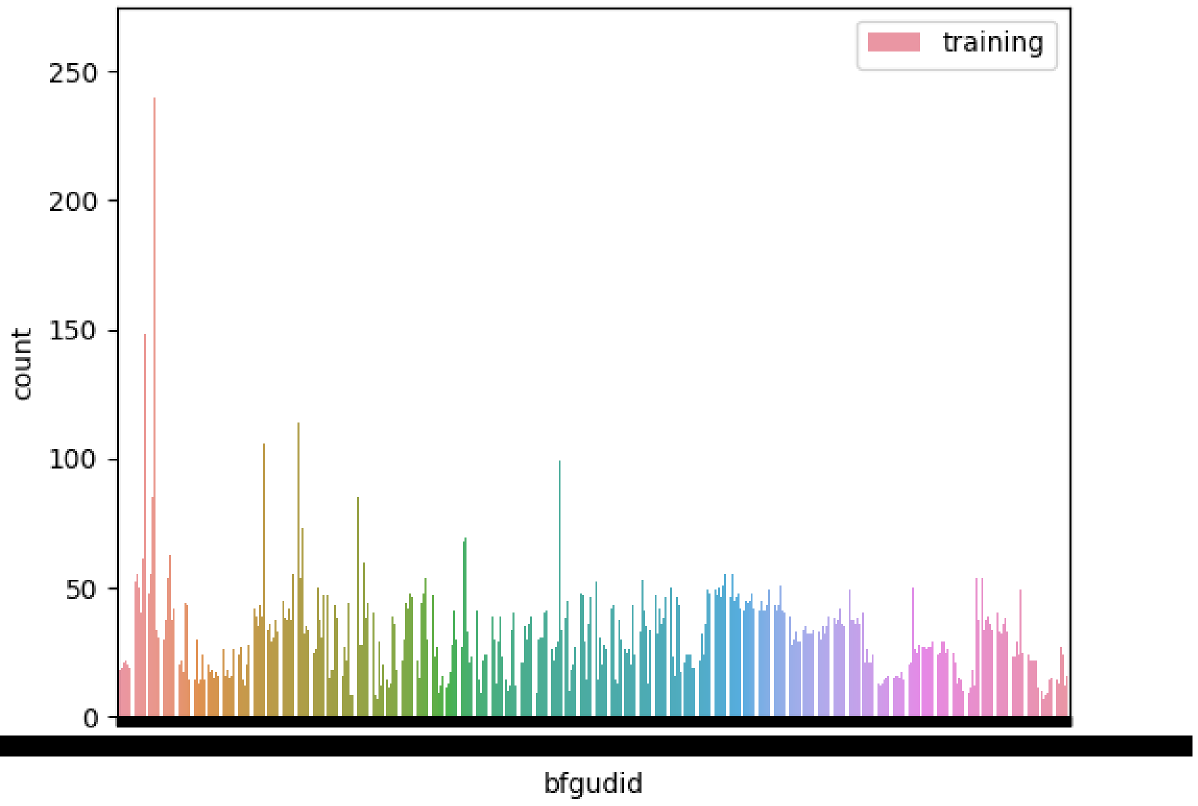
***Note**: each row represents a unique user day, unless there is more than one purchase on that day, in which case the purchase data ('product', 'currency', 'gross_usd_amount', 'price_tier') would be different but the rest of the columns are duplicates*

You will build a model to predict the optimal offer price tier for each user (denoted by a unique "bfgudid") in the test set. There are a few valid approaches, such as:

1. Use all but the last purchase history as input to predict the price of the last purchase
2. Use the single previous purchase history as input to predict the price of the next purchase
3. Use all, up to nth purchase history to predict the price of the n+1 purchase
4. Any other that achieves the same objective

## Data Exploration and Pre-processing

Training dataset includes game information of 8000 gamers with 255,969 entries. Gamers' number of entries are variable, as shown below.



Investigation on NaN values showed that "product", "currency", "gross_usd_amount", and "price_tier" features have 176,329 not defined values. Since "price_tier" is considered as dependent variable, it was decided to remove the users that do not have a complete data for their "price_tier". Size of the dataset was reduced to 79,367 entries. For the simplicity purpose and overcome shortage of time, the categorical features were removed. The gamers minimum, median, and maximum number of entries were identified.

Minimum number of plays: 1

Median number of plays: 5

Maximum number of plays: 259

Then, users with less than 5 entries (median in counts of the number of plays) were removed as for the modeling we need to have previous behavior of gamers to be able to predict their future behavior. Number of unique gamers is 4296.

## Training and validation data

Since all the users have at least five entries, again to do a simplification only the first five gamers' entries, ascendingly sorted by time, were stored in two training and testing data frames, shown below. Testing dataset is 556/(556+2966) or 16% of total data.

| Dataframe Name | Date Criteria | Number of entries | Number of gamers |
|---|---|---|---|
| df_train | Starting date: 2020-12-15<br>Last entry <= **2021-01-20** | 14,830 | 2966 |
| df_test | First entry > **2021-01-20**<br>Ending date: 2021-01-31 | 2780 | 556 |
| df_notTrain_notTest | Data with entries before and after **2021-01-20** | 3045 | 609 |

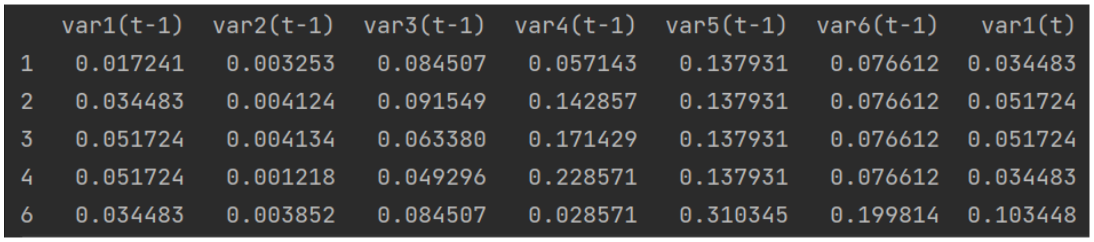
## Data Normalization

To equalize impact of features with different range, they normalized using maxmin scaler method from sklearn library.

## Data Restructuring for LSTM using 2nd approach (single history and future purchase)

The ideal solution is to test all the three proposed approach and to consider number of purchasing history (sliding window in LSTM terminology) as a hyper-parameter. However, to keep the time spent on the question at the scope of few hours, the second approach (usage of single history to predict one timestamp in future) was selected, as it was deemed that the best predictor of a user's behavior depends on the most recent behavior in the past. Of course, this assumption could be tested by going through other approaches and comparing the accuracy of respective results.

In the LSTM model, we need to include the dependent variable from a previous time along with the features to predict the dependent variable of the next time step. For that reason, a restructuring of the data frame as shown below is performed.


```
   var1(t-1)  var2(t-1)  var3(t-1)  var4(t-1)  var5(t-1)  var6(t-1)   var1(t)
1   0.017241   0.003253   0.084507   0.057143   0.137931   0.076612  0.034483
2   0.034483   0.004124   0.091549   0.142857   0.137931   0.076612  0.051724
3   0.051724   0.004134   0.063380   0.171429   0.137931   0.076612  0.051724
4   0.051724   0.001218   0.049296   0.228571   0.137931   0.076612  0.034483
6   0.034483   0.003852   0.084507   0.028571   0.310345   0.199814  0.103448
```

Where

| Var1 | Var2 | Var3 | Var4 | Var5 | Var6 |
|---|---|---|---|---|---|
| price_tier | play_seconds | play_session_count | day_since_install | max_price_tier | avg_price_tier |

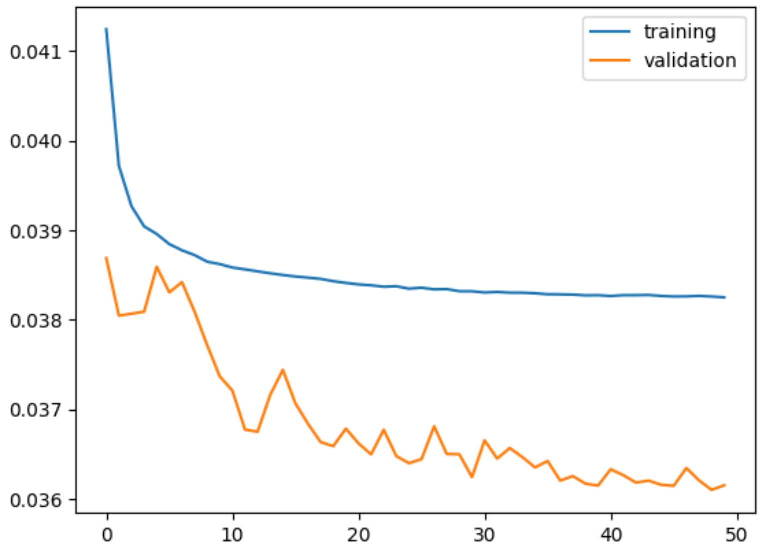
Note in above example, the fifth entry had been removed because that would include dependent variable data from the first gamer at (t-1) timestamp along with dependent variable data from the second gamer at (t) timestamp. To keep data pure and without mixing between the gamers with five samples, all the gamers' entries that were multiplier of five were removed. The same was done on the testing data.

## Model, Analysis and Results

History of the users' purchase behavior and the system response are captured for the prediction of the users’ next purchase. In order to learn patterns from training data, a time-series Long short-term memory (LSTM) algorithm on a Keras machine learning framework was selected. LSTM is one of the Recurrent Neural Network (RNN) algorithms well-suited to time sequence data. Model summary is shown below.

```
Model: "sequential"

_________________________________________________________________

 Layer (type)             Output Shape          Param #

=================================================================

 lstm (LSTM)              (None, 50)          11400

dense (Dense)              (None, 1)          51

=================================================================

Total params: 11,451

Trainable params: 11,451

Non-trainable params: 0
```

Loss values for training and validation data is shown in the following plot.



LSTM result showed Test RMSE = 3.473 and R2 = 0.2. Obtaining this result, is to some extend due to many simplifications that were applied to the question in order to be able to complete the whole cycle of solving the test in the limited few hours, as it was requested.

## Conclusion and Disclaimer

The whole cycle of data cleaning, feature engineering, and modelling was performed on the technical exam. It included data exploration, data pre-processing, splitting the data to the training and testing sets, normalizing the data, restructuring the data for LSTM, building a LSTM model, running it, and obtaining the result. A basic and simplified approach was considered and performed on the mentioned steps, to follow the test's instruction in terms of the spent time.

In the following, the questions that were proposed at the last part of the test are answered.

- **You can discuss any potential improvements or other ideas in your write-up.**

While I still think LSTM is one of the best models to use for this task, more AI models could be tried and compared. In terms of the current used data and model, I believe there are possibilities of improvement in terms of dealing with both data and model. Rather than using a portion of data, it is the best to maximize usage of data, if computing resources allowed. I should share that my personal PC is not powerful enough to deal with big data. In terms of the model, hyper-parameter optimization could be performed on the LSTM model through a grid search.

- **The reasoning behind your decision, along with your other ideas.**

The second approach (usage of single history to predict one timestamp in future) was selected for two reasons. (1) To keep the time spent on the question at the scope of few hours. (2) It was deemed that the best predictor of a user's behavior depends on the most recent behavior in the past. Of course, this assumption could be tested by going through other approaches and comparing the accuracy of respective results.

- **What is your strategy for approaching problems like this?**

I follow a Minimum Viable Architecture (MVA) and incremental improvement strategy for approaching problems like this. If a simple model can quickly go from A to Z, it can be improved incrementally and thoroughly.

- **Why did you choose your approach over the others?**

To follow my strategy, i.e. a MVA and incremental improvement's approach. Meanwhile, I think the best predictor of a user's behavior depends on the most recent behavior as the user's memory is fresher, either if we look at recent past (one timestamp before), or near future (one timestamp after).

- **What other features could improve model accuracy?**

Gamers' involvement in playing can be affected by the time they play. For example, in the holidays, users spend more time to play, or which day of the week can have an impact. Therefore, from the "activity_date" feature I would create other new features such as "isHoliday" vs "isNotHoliday", or day difference from boxing day, or the day of the week and so forth.

The gamer per se could be considered as a separate feature so that model can learn something about the behavior of each gamer as there will be new gamers with similar pattern of behavior, for example gamer X is relatively effective, but not on weekends, whereas gamer Y is great every day.

The categorical features with few numbers of classes such as “app_store” need to be encoded by one-hot encoding method to be converted to three numerical features. However, to save computing resources for categorical features that have many numbers of classes, we can use a clustering method and group similar gamers in smaller number of classes. We can also use latent representation to embed the classes with high dimensional vectors to a smaller vector whose length approximately matches the hidden size of the LSTM.

- **Are there additional data you would request?**

Yes, any users’ demographic data such as gender, age, income, social class, even personality traits (like big five) and so forth could increase power of the prediction.

- **If you had four weeks on the project rather than a few hours, how would you use that time to improve your work?**

I would start with I did now. Then, based on the results and what were proposed and or disclaimed above, I would write a prioritized action plan, and perform a more thorough analysis of what was performed in this study.