ESE 527 Project Final Report
In-App Purchase (IAP) Recommendation System

Songhua Zheng (501284), Yuedi Mi (502306)

Mentor: Dr. Patricio S. La Rosa

1. Executive Summary:

In this project, we explore the insight of In-Game consumers' behavior for a mobile game called "Ninja Must Die". The goal is to classify players for the IAP recommendation aimed with various customized products.

The processes include but are not limited to: Background study of our client, understanding fundamentals of recommendation system, data preprocessing for the classification problem, implementation of machine learning models, comparison of machine learning models cost and efficiency, evaluation of models' performance, and making data-driven decisions.

The inspiration of this project is to expose to the real-world scenario, and practice data analysis skills with raw data to build a project from the ground up.

For us being a gamer for a long time, we have experienced the attraction of IAP to us to contentiously making purchases in gaming. Also, some imperfections that could influence users' experiences. According to an article from Mihonvil Grguric "79% of gaming apps currently monetize via in-app purchases, while 50% of non-gaming apps do the same." (Grguric, M. 2022, January). And an article from Ted Verani "In-app purchase (IAP) sales drive 43% of mobile gaming revenue and 21% of non-gaming app revenue. Long-term users may be more profitable than new users. In fact, data suggests that 8% of long-term customers account for 40% of all sales." (Verani, T. 2020, August). There are three concentrations in profit from a mobile game: In-App Purchase (IAP), Advertising revenue, and the hybrid mode. **IAP is the one that maximize the user experience to create long-term values.** Analyzing users' in-game habits, usage habits and demographic features gives valuable insights for enhancing the users' experience.

A precise recommendation stimulates users to making purchase. At the same time, from the customer behavior aspect, users tend to stay online (loyalty increases) after making a purchase, either the frequency online or the time duration of playing. This potentially keeps revenue rolling since the frequency of in-game products delivery is proportional to the frequency of players online.

2. Data Description and Preprocessing

Data Asset

Two datasets (game_data_1.csv and game_data_2.csv) were provided by the company YanHun Network Technology (https://app.mokahr.com/apply/yanhun/24016#/). Each dataset contains information of users and users' behaviors data. A new dataset generated after the processing ("project_data.csv").

the generated dataset has 26 attributes in two categories: user information and in-game behaviors. They are:

"uid": user ID

"sex": Gender

"level": Level of user in-game

"rank.level": Rank level of users in-game

"pur_amount": amount of purchase made in-game

"num_of_pur": number of purchase made in-game

"date_last_pur": date of last purchase

"date_sign_up": date of user started to play

"active_days": days active in-game after sign up.

"server": which server user registered

"rank_city": rank of the city where user located

"rating": user's rating for the game

The following attributes are summarized of user in-game for specific activities:

"num_x33", "num_arena", "num_reward", "num_main", "num_secret", "num_fam_abyss", "num_fam_battle", "main_schedule", "daily_33", "daily_arena", "daily_reward", "daily_main", "avg_daily_battle"

And our target variable:

"type_player": player value classified by the purchase made

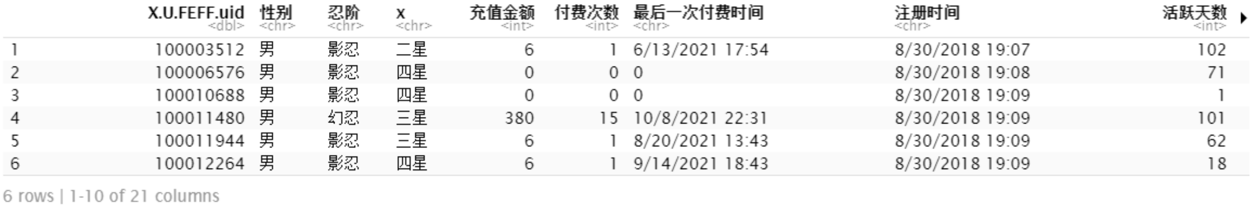
Preprocessing

- Transform Chinese characters

  R cannot process Chinese characters properly for some functions. Also, for the audiences to understand the content better. We transferred all Chinese characters into either English or numbers while processing the data. The generated dataset is now in English characters.

- Handling categorical data

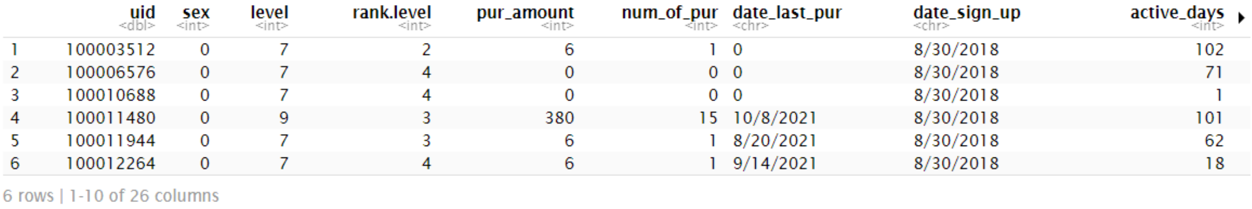
  Since we are constructing a classification problem, it is better to avoid categorical data, we use functions like "factor" and "level" and package "tidyverse" to modify most categorical data. Some of them are redefined, for example our target value "type_player" was divided into 6 classes, we transferred them into "low", "medium", and "high" to match the respective meaning of original 6 classes.



	X.U.FEFF.uid <dbl>	性别 <chr>	忍阶 <chr>	x <chr>	充值金额 <int>	付费次数 <int>	最后一次付费时间 <chr>	注册时间 <chr>	活跃天数 <int>
1	100003512	男	影忍	二星	6	1	6/13/2021 17:54	8/30/2018 19:07	102
2	100006576	男	影忍	四星	0	0	0	8/30/2018 19:08	71
3	100010688	男	影忍	四星	0	0	0	8/30/2018 19:09	1
4	100011480	男	幻忍	三星	380	15	10/8/2021 22:31	8/30/2018 19:09	101
5	100011944	男	影忍	三星	6	1	8/20/2021 13:43	8/30/2018 19:09	62
6	100012264	男	影忍	四星	6	1	9/14/2021 18:43	8/30/2018 19:09	18

6 rows | 1-10 of 21 columns

Before



	uid <dbl>	sex <int>	level <int>	rank.level <int>	pur_amount <int>	num_of_pur <int>	date_last_pur <chr>	date_sign_up <chr>	active_days <int>
1	100003512	0	7	2	6	1	0	8/30/2018	102
2	100006576	0	7	4	0	0	0	8/30/2018	71
3	100010688	0	7	4	0	0	0	8/30/2018	1
4	100011480	0	9	3	380	15	10/8/2021	8/30/2018	101
5	100011944	0	7	3	6	1	8/20/2021	8/30/2018	62
6	100012264	0	7	4	6	1	9/14/2021	8/30/2018	18

6 rows | 1-10 of 26 columns

After

- Outlier Detection and Removal

  We used Mahalanobis Distance to determine the outliers because it worked well with multivariable and imbalanced data. It measures the distance between sample point and distribution. The problem we encountered is even we set up the cut-off extremely high, we still have around 9% of the samples are marked as outlier. We decided to compare the model performance with and without outlier removal to determine if it is worth to remove them instead of losing some significant information in the dataset. i.e., if the accuracy of the validation sets doesn’t change much before and after removal, we will not consider outliers as a problem.

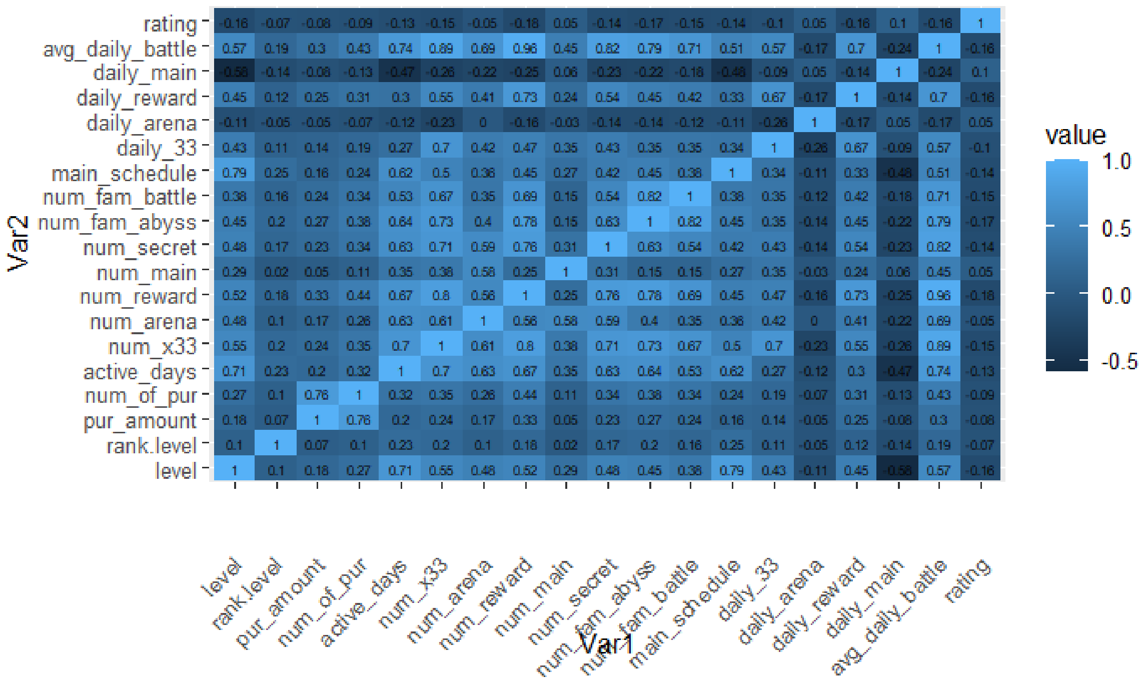
- Feature Scaling

  It is noticeable we have different ranges in features across the dataset. Since most of our models are distance-based approach (ANN, SVM, KNN), we decided to rescale the features with normalization.

  By examining feature distributions, we discovered some of the feature have highly right skewness caused by zero value samples. We performed the natural log transformation, with adding a constant to each sample to avoid resulting in “-inf” for those zero value samples.

- Dimension Reduction

  We used three approaches for dimension reduction. Examine the correlation between features to remove overlay features to reduce the redundancy, ranking features importance and Greedy-Backwards feature elimination for the last if needed.



Modeling Approach

- Experience during modeling process

  In the early stages of modeling, we have encountered serval difficulties. We have reached extremely high performances in the training set for every model. First, we thought the models were overfitting the data, we increased the number of folds for the cross-validation and adjusted some of the details in data cleaning. And the models still reached an extreme score. We then investigated our features. By adding our target variable ("type_player" represents the value of the player) into the correlation matrix we discovered one feature ("pur_amount" represents the purchase amount player made) we selected has a correlation of 1 with the target variable, which means our target variable was conceived based on the purchase amount, in another words, the value of the player is generated by the purchase amount. The problem was solved after removing this feature. However, our models lost their performance after removing that feature, the scores are dropped to an average of 65, which is not much higher than tossing a coin. The problem was solved after investigating feature distributions and the median of the residuals of models. We chose to apply log transformation by taking the natural log with adding a constant to each sample. This solved the performance issue and concentrated our feature distribution which led to a better performance in the models.

  Another difficulty encountered is we lack computing power. Models like Random Forest, SVM, and ANN took significant time to fit the data. We decided to divide our features into two groups. The first group is the features that represent the daily in-game behaviors of users (column name begins with the keyword "daily"); the second group is the features that represent the total average of specific in-game behaviors of users (column name begins with the keyword "num"). By dividing the features into groups, we were able to conclude faster by comparing the results of models for the two groups.

- Model Selection and hyperparameter tunning

  We chose 5 models for the classification problem and tunning the parameters for each model for the best performance. Consider the size of our dataset, we chose 10-fold cross-validation with 10 iterations to make sure the predictions are unbiased. Below is the table of models' Pros and Cons we concluded in this project and parameter we tuned.

	Pros	Cons	Parameters Tuned
K Nearest Neighbors	Easy to implement and apply adjustments	Preprocess required Cost too high for large dataset like we have	K values
Decision Tree	Fast and Efficient	Overfitting	Mtry
Random Forest	High accuracy and prevent overfitting	Extremely cost in training time	Tree size (cp)
Artificial Neural Network	Well fitted with dataset and result in good scores.	Hard to implement after modeling	
Support Vector Machine	Works well in high dimensional space	Doesn’t fit large dataset and timely cost.	Margins of boundaries

- **Evaluation Metric**

We used Mathew Correlation Coefficient (MCC) learned from lecture as the evaluation metric. The reason we use MCC is we have an imbalanced target variable (“low” for 46,691, “medium” for 20,554, “high” for 13,641), and some of the data in class “high” were removed as outliers due to unnormal user behaviors. Using MCC is a more reliable statistical rate because of MCC evaluating all the four confusion matrixes (TP, FN, TN, FP). Since the MCC normally works for binary-class classification, we had to make sure the function we used works for multi-classes as well. According to the article by Jurman.G, Riccadonna.S, and Furlanello.C, the MCC for multi-classes could worked if the condition were met. The basic concept is

$$\mathrm{MCC} \quad = \frac{T^2 + (N-2)TF - (N-1)F^2}{[T + (N-1)F]^2},$$

Result and Insights

- Results

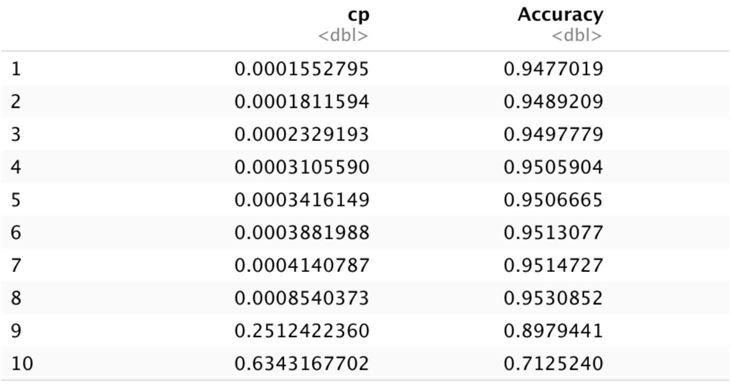
The result we evaluated is the models' performance measured by the MCC, the calculating time and efficiency of the models, and the comparison of two groups of features. Below is the table of results mentioned.

	MCC for group "num" [-1,1]	MCC for group "daily" [-1,1]	Run Time "daily" in seconds	Run Time "daily" in seconds
KNN	0.736	0.754	162.2	132.9
Decision Tree	0.913	0.947	16.0	14.9
Random Forest	0.909	0.882	396.6	347.9
SVM	0.896	0.900	1212.8	1731.0
ANN	0.913	0.874	1333.8	1294.1

- Insights

By looking at the table above, we noticed that KNN has the worst performance due to the large size of sample and rounding mechanism inside "caret" function misleading the distance calculation for KNN. Other models performed relatively close. The cost and efficient of SVM and ANN are obviously higher than the others. Which we have mentioned in the modeling section. These two models were designed to solve complex problem with high dimensional data. Which consider unworthy for this problem. However, Decision Tree is significantly cheaper than the others, which also has high scores. We then checked the validation accuracy to see if the decision tree model is overfitting the data:

The accuracies of performance on most validation sets slightly increased and steady around 0.95. So, it doesn't overfitting the data. Similarly, Random Forest also has a reasonable run time and accuracy. We chose Decision Tree and Random Forest as our final candidates. To select the best model, we random draw samples from validation set and simulate them as real time data feeds to the two models.



```
                  cp      Accuracy
               <dbl>         <dbl>
1     0.0001552795       0.9477019
2     0.0001811594       0.9489209
3     0.0002329193       0.9497779
4     0.0003105590       0.9505904
5     0.0003416149       0.9506665
6     0.0003881988       0.9513077
7     0.0004140787       0.9514727
8     0.0008540373       0.9530852
9     0.2512422360       0.8979441
10    0.6343167702       0.7125240
```

Conclusion

Throughout this project, the most impressive part of the process is the importance of the quality of data. We mentioned above that one of the inspirations for choosing this dataset is to solve a problem from the ground up. The dataset we usually handled and provided by other courses is cleaned and scaled. This dataset provided by the company is not originally for classification purposes. We had to fix, clean, and transfer data types, scales, distribution, etc. to process with classification problem. Back to the question from the beginning, we have successfully developed a model to predict the value of a user based on the in-game behaviors. The result still has imbalanced classes, the number of class “low” covers the total number of class “medium” and class “high”. This indicates using the IAP strategy to increase profit might not be an ideal decision. We suggest the company develop new content built around IAP products, however, it is an expensive process. Or shift the concentration to hybrid mode, which is a combination of advertising revenue and IAP.

To improve our model, we are looking to develop a combined model with two groups of features we divided. The purpose is to automatically find the best performance groups of features. And a new model for the survey data in the dataset which we removed in this project. Those survey data were collected as binary (0 and 1), which are potentially easier to interpret and modelling. Also, the survey data might give the potential of uses’ future value at early stages before going into the behaviors data.

Reference:

Verani, T. (2020, August 3). *The most important IAP statistics for mobile game publishers in 2020*. wappier. Retrieved February 6, 2022, from https://wappier.com/blog/iapstatistics#:~:text=IAPs%20drive%2043%25%20of%20gaming,by%20rewarded%20video%20ad%20revenue.

Jurman, G., Riccadonna, S. and Furlanello, C., 2022. A Comparison of MCC and CEN Error Measures in Multi-Class Prediction.