EBA5002: Graduate Certificate in Business Analytics Practice

Operation Optimization with a Dive into Genshin Impact's Comments

16 Nov, 2022

Team Members

Wu Yangyi A0261688U

Li Kongwen A0261826B

Shen Yi A0262013X

Chi Yijin A0261891X

Lin Fangzhou A0261850H

Contents

1 Introduction & Background

Genshin Impact, a prevailing open-word action role-playing game launched in Sep 2020, has recently reached its second anniversary. It was awarded Best Mobile Game in 2021 and remains on the list of top-grossing mobile games worldwide since its launch (2020-2022). Its player pool keeps growing larger and results in an increasing number of comments in the application stores, including Google Play Store. And meanwhile, Mihoyo, Genshin Impact's company, is also developing two additional new game products targeting similar players. At the second anniversary, the operation team wants to get some insights from the players 'comments to prevent a declining trend in reputation.

2 Problem Statement

As the player pool grows larger, the number and the variety of comments in the application stores also grow larger. As popular as it is today, a potential reputation drop would be costly. And meanwhile, for the two developing new games with similar target players, what might be the alerts, and what can be referenced?

3 Business & Technical Objectives

3.1 Business Objectives

- To deal with the growing variety in comments, the operation team wants to update the auto-reply corpus. To design new automatic replies, the distribution of sentiment and topics are requested.
- What is complained by the players, and what is appreciated? In knowing these, improvements can be made for Genshin Impact and the appreciated features can be referenced by the two developing games.
- In preventing a reputation crisis, the operation team wants to detect what is the timing to take some maintenance actions in advance.

3.2 Technical Objectives

- Conduct sentiment analysis and LDA over the comments to know the sentiment and topic distribution.
- Develop a classification model that served as a pre-warning model to decide whether the comments are showing an alert so that actions need to be taken to appease the players.

- Through word cloud analysis and LDA on focused groups of comments, to find out what is doing bad for refinement, and what is doing good for the new games' reference.

4 Stakeholder Strategy

Stakeholder	Key Concerns	Response Action
Operation Team	1. Find out major topics involved in players' comments 2. Score isn't an indicator as effective as expected, sentiment of comments maybe a good surrogate 3. Penalty should be higher on misclassification of a timing when intervention are necessary	1. Apply LDA model to find out the drawbacks and strengths of the product 2. Weighted average of score and sentiment to create a new indicator of reputation 3. Use cost function to determine the model

5 Project Design

5.1 Data Management

In terms of data management for this project, we initially extracted Genshin's comments, game performance data and industry data from various external sources. Before loading the data-to-data warehouse, we conducted data validation, cleaning, transformation and integration, including dealing with data inconsistency, removing duplication and outliers, imputing missing values as needed, and deriving new variables. In the process of analysis, we selected valuable features from the warehouse and took advantage of BI tools to derive insights.

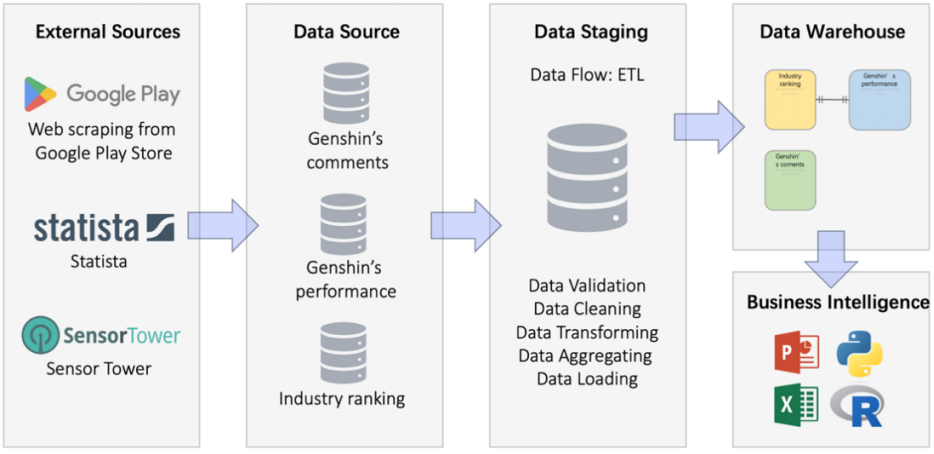

Figure 1. Process of data management

5.2 Analytical Pathway

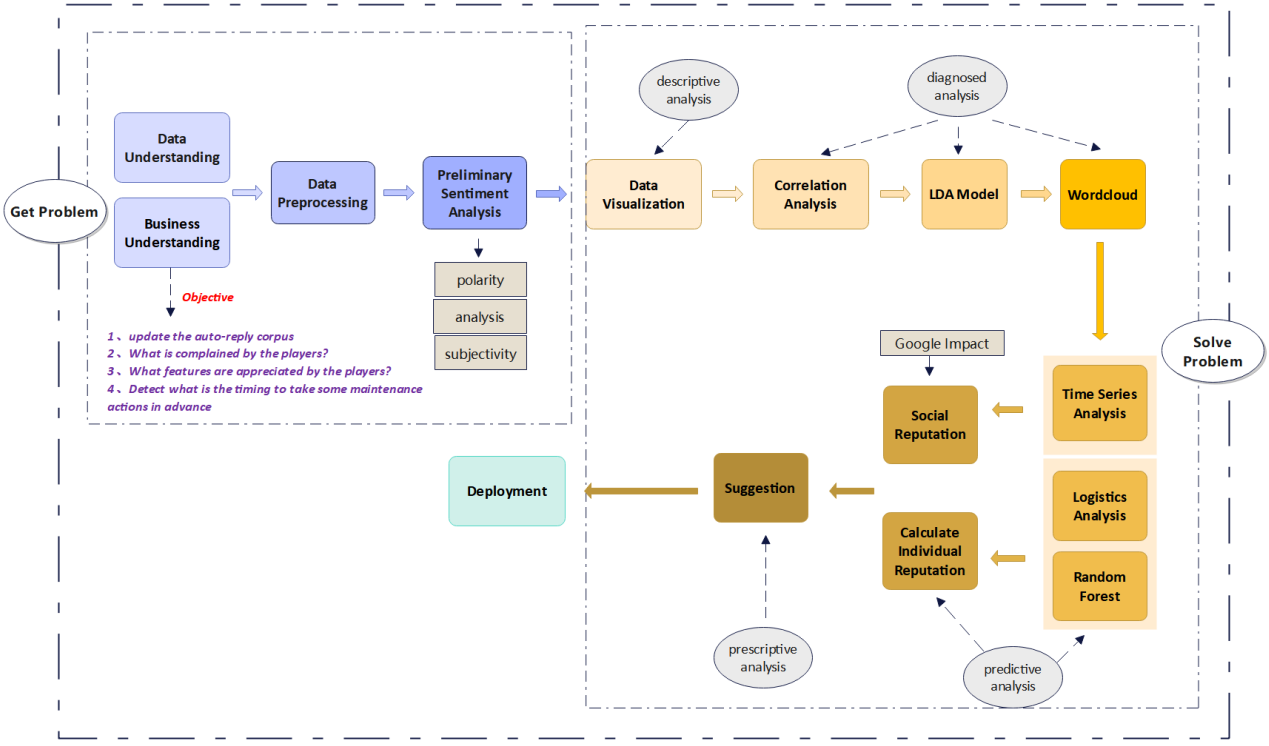

Figure 2. Analytical pathway

6 Data Set

6.1 Data Description

6.1.1 Players' Comments from Google Play Store

Our data was crawled from the Google Play store. Originally, there are 8 variables, which include both numerical and categorical variables: "reviewID", "content", "score", "thumbsUpCount", "reviewCreatedVersion", "at", "replyContent", "repliedAt". We newly created 7 additional variables: "Polarity", "subjectivity", "ABS", "attitude", "review_length", "usefulness", "days of comment". In total, we have 15 variables (See Appendix 1).

There are 1273 observations in the dataset. We also have 642 fresh comments dated after 14 Oct as our deployment dataset. When building the classification model, we used 70% of historical data as the training set and 30% as the testing set.

6.1.2 Google Trends

We crawled weekly google impact data from the official website of google trends for Genshin from September 2020 to October 2022. There are 2 variables "week" and "google impact". We use this data to compute the time series analysis and prediction (See Appendix 2) .

6.1.3 Performance Data of Genshin Impact

On the other hand, we scrabbled some variables to find whether they have correlation with the performance of the Genshin Impact, which includes numerical variables: "App_revenue", "app_downloads", "Avg_google_impact", "Num_hours_watched_Twitch", and "Average_Monthly_Players", and "google_impact"(See Appendix 3).

6.2 Data Preparation

6.2.1 Data Cleaning

We tackled the following cleaning challenges: (1) missing data (version number is missing for some comments); (2) data dislocation; (3) emoji in the comments; (4) duplications

6.2.2 Data Construction

- **ABS**

In the score of reviews, high-scoring reviews are as important as low-scoring reviews, because they will reflect the player's attitude towards the game. Therefore, we define the deviation degree of each score from the mean by constructing ABS index, to judge the extreme degree of score. The specific equation is as follows:

$$\text{ABS} = |\text{score} - \text{mean score}|$$

- **Days of comment**

The difference between the date the review was published and the crawler date can reflect the time variable, which is conducive to studying the relationship between the number of days the review was published and the number of likes.

- **Usefulness**

For the operation team, useful reviews have practical value for decision-making, so it is necessary to judge the usefulness of reviews. We set the quartile (6) of the historical data as the threshold, and when the thumbs up number of the review is bigger than 6, the review is regarded as useful (1), otherwise it is regarded as useless (0).

- **Polarity**

Quantify the sentiment of reviews from negative to positive. The range of Polarity attribute is from –1(extremely negative) to 1(extremely positive).

- **Subjectivity**

Quantify "private states" (opinions, emotions, sentiments, beliefs, speculations) from the reviews, range from 0 to 1.

- **Review_length**

The total length of the reviews was calculated based on the number of words.

- **Attitude**

In preventing the player attitude decline, we constructed an indicator as the base measure of the pre-warning model for operation team to detect when to take maintenance actions in advance. We leveraged score and sentiment polarity to construct player attitude from comments, giving weights of 0.3 and 0.7 separately. The range for attitude is from -0.4 to 2.2.

$$Attitude = 0.3 * Score + 0.7 * Polarity$$

Variable	Range	Type
Attitude	[-0.4, 2.2]	Decimal
Polarity	[-1, 1]	Decimal
Score	[1, 5]	Integer

Table 1. Relevant data dictionary

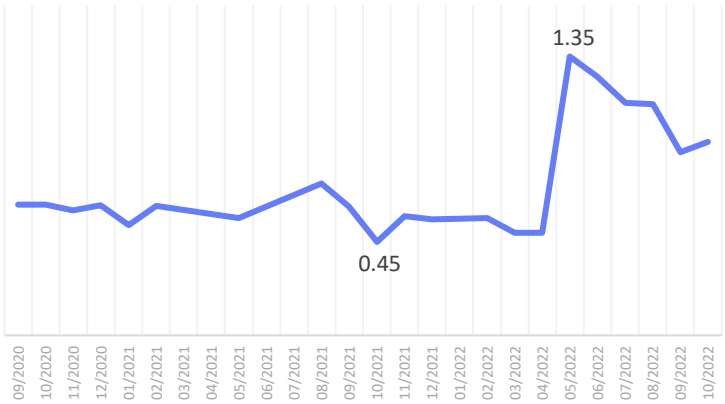

Figure 3. Monthly attitude

We took polarity into account because there were cases where the review is with a high score but negative sentiment, suggesting the inconsistency of score and sentiment polarity. So, the score alone may not truly represent the reviewer's attitude. Hence it is essential to consider the sentiment polarity when constructing the attitude.

Score	Polarity	Content
5	-0.4	I was going to play this game, **but** the download resource data is 18 GB
5	-0.15	This game is very fun, has been playing it ever since the game release. The few **problems** I had been when I played in mobile it lag all the time. There's also an **issue** in the cut scenes. Hope mihoyo can fix these problems for all of us mobile user.
5	-0.13	**Update:** Issue resolved! **Sad** that they didn't address the mobile **issue** in their recent development notes. **Stuck in** the white screen of death.
1	0.78	Very **nice** size.
1	0.42	Happy 2nd anniversary, great rewards, exciting events especially the 10% only win welkin, **LOVE IT**!

Table 2. Reviews with high(low) score but negative(positive) sentiment

Secondly, we focus more on negative or low-score reviews from players since they reflect the attitude drop lie in players which may ultimately lead to a reputation drop. Among all useful comments (number of thumbs > 6) in our data set, if we group them by score, there are 59% low/medium comments. If we group them by polarity, there are 79% negative/neutral comments. So, clearly, polarity captured more comments of our interest. Therefore, we give polarity more weight than score. By giving more weight to polarity, the indicator will be more sensitive to negative and neutral polarity therefore effectively capturing the attitude drop. This is in line with the concern of the operation team.

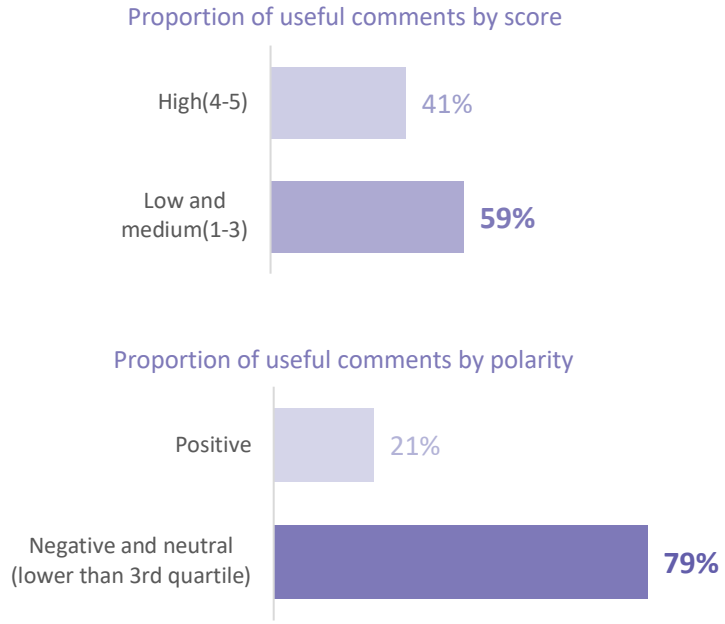

Figure 4. Players pay more attention to reviews with negative and neutral sentiment polarity

7 Objective 1 — Update the Auto-Reply Corpus

7.1 Topic Modeling – LDA

First, data pre-processing was conducted for LDA. During the data pre-processing, we also derived additional comment-based stop words which have high frequency but play trifling roles in defining the topics.

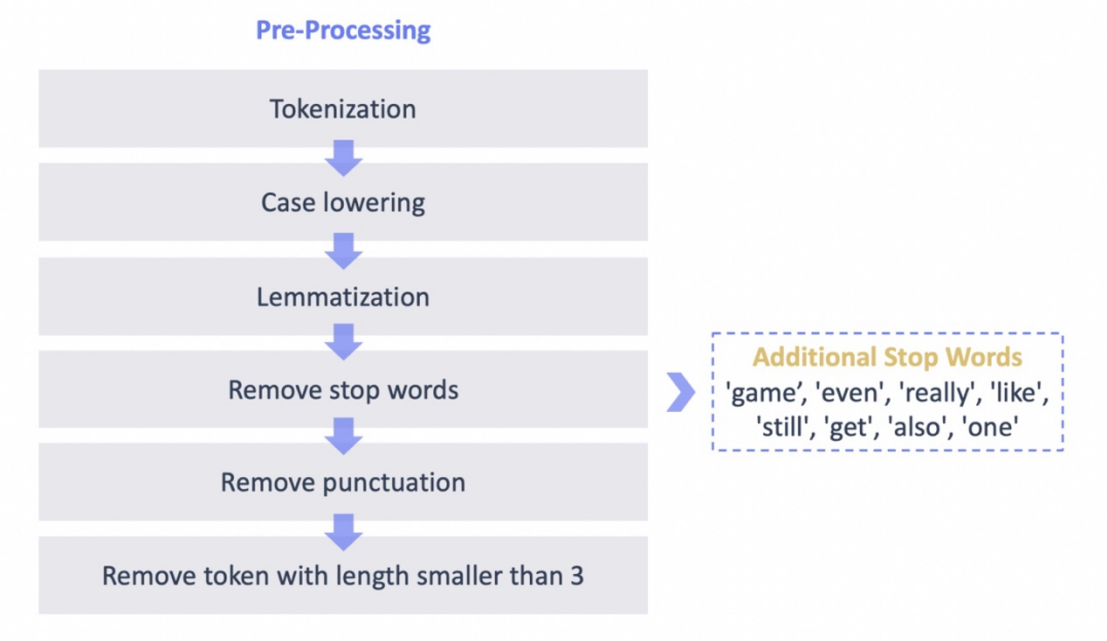

Figure 5. Data pre-processing for LDA & additional stop words

Considering both model performance and contextual information about the game, we came up with an acceptable model (chunk size = 4, passes = 10, UMass coherence = -2.5639, CV coherence = 0.4093) displaying four topics.

4 Topics

Topic 1	43.7% of tokens
topic 2	23.4% of tokens
Topic 3	18.8% of tokens
Topic 4	14.0% of tokens

Table 3. Percentage of Tokens within each topic

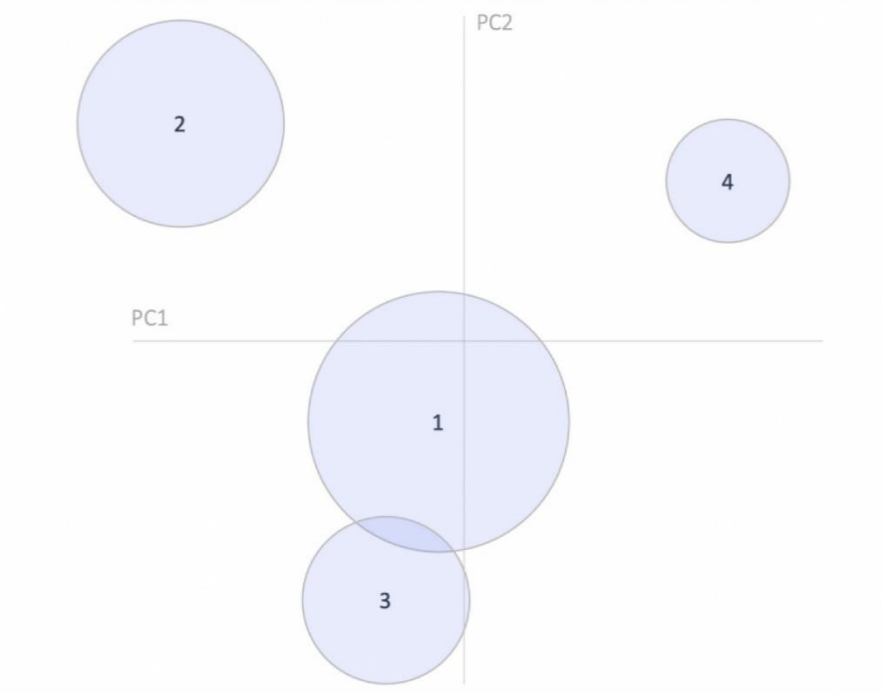

Figure 6. Intertopic distance map (via multidimensional scaling)

Summarizing the most relevant terms, we named the four topics.

Story Design character, story, great, player, time, good, fun	**Mobile Technical Problems** phone, play, mobile, problem, storage, controller, space, fix
Graphic good, update, graphic, character, play, new	**Character Design** character, best, amazing, look, love, fighting

Figure 7. Most relevant terms for the 4 topics (sorted by estimated frequency)

The first topic, which covers the greatest proportion of tokens, is about the story design. The related comments show appreciation to the story design where characters are elaborately involved.

The second topic is about the technical problems encountered when playing the game on mobile devices. The main problems include: the game is occupying too much storage, the controller sometimes does not work as smoothly as expected, or there are some bugs reported.

The third topic is about the Graphic. The game is excellent in its Graphic of both the world and the characters. Each update brings a surprise.

The fourth topic is about character design. Besides an amazing look and well-designed outfit, the motions of the characters, especially the effects during the fighting, are highly appraised.

	Topic	Review Number
1	Story Design	404
2	Technical Problem	131
3	Graphic	570
4	Character Design	168
	Total	1273

Table 4. Number of comments for each topic

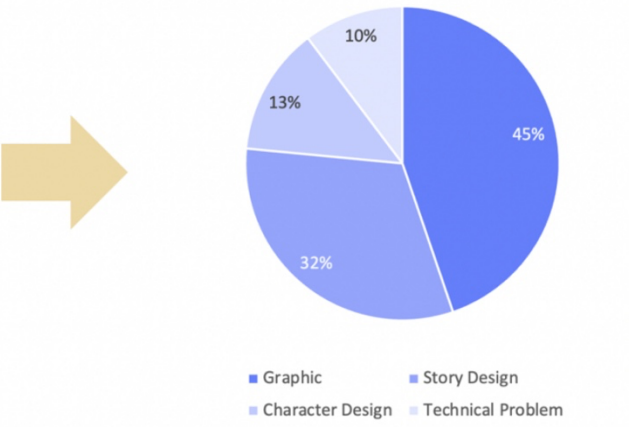

Figure 8. Topic distribution

Among all the comments, topics about Graphic and story design occupy a lion's share. Next follows the character design and technical problems. Therefore, we suggested the operation team redesign the auto-reply corpus into such a distribution of topics.

7.2 Distribution of Sentiment (Polarity)

Besides topic, sentiment distribution, especially polarity, is also a crucial factor to consider. Within the range from -1 to 1, we divided the polarity of sentiment into 4 parts: strong negative, moderate negative, moderate positive and strong positive, with -0.5, 0 and 0.5 as the separations.

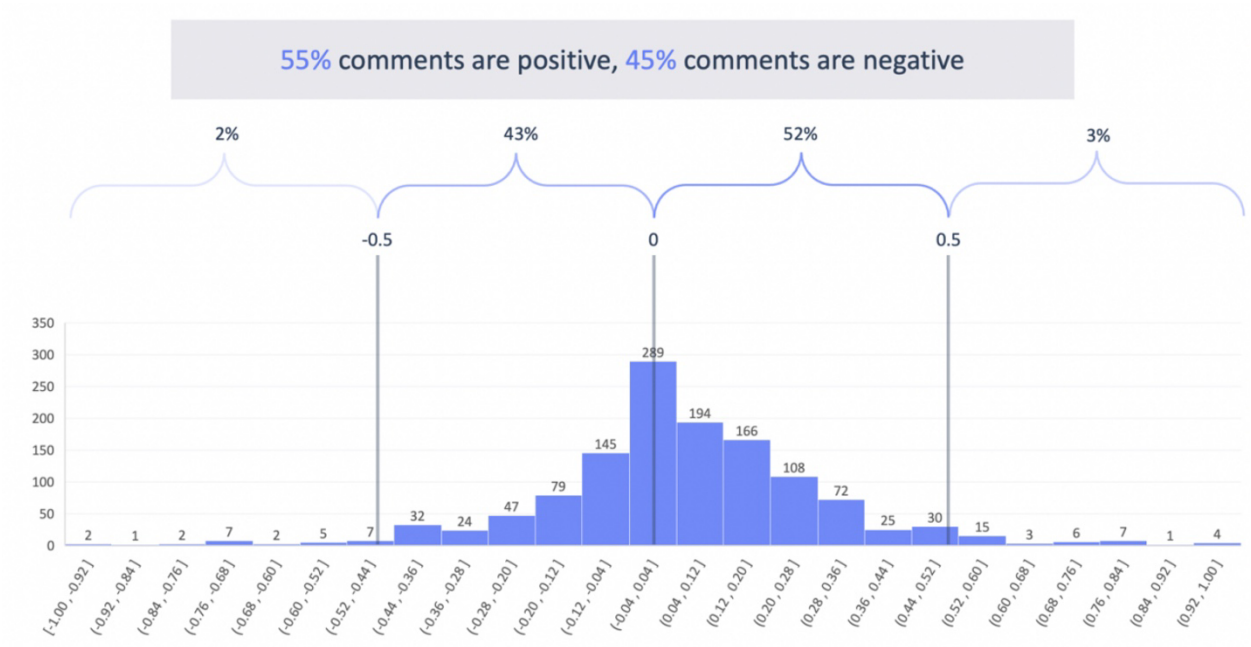

Figure 9. Sentiment (Polarity) distribution

Following a relative normal distribution, there are 55% positive comments, and 45% negative comments. In re-designing the autoreply corpus, such a sentiment distribution can also serve as a guide for the operation team.

8 Objective 2 — Pros & Cons of Genshin Impact

8.1 Word Cloud for Features need to be improved

We divided the comments into two categories based on sentiment score and calculated the frequency of specific tokens in these two types of comments.

Figure 10. Word cloud of negative comments

From the word cloud, the potential problems that we derived from negative comments are system bugs, time-consuming to load the game and to go through some quests in the game. Meanwhile, the game occupies a large amount of storage.

8.2 Word Cloud for Features Appreciated by Players

Figure 11. Word cloud of positive comments

The most appreciated features of Genshin Impact are the elaborate design of characters and storylines. Most players recognize the game incredibly fun and beautifully designed.

9 Objective 3 — A Pre-Waring Model for Alerts

In preventing the player attitude decline, we built a pre-warning model for operation team to detect when to take maintenance actions in advance. Firstly, we constructed an indicator for

attitude by leveraging score and sentiment. Then we determined a pre-warning model by calculating the percentage of useful reviews with poor attitude and comparing it with the historical performance. We utilized useful reviews predicted from classification model in the previous part as deployment data to show the function of the pre-warning model.

9.1 Alerting Mechanism

After the attitude construction for each review, we built a pre-warning model for the operation team to detect when to take maintenance actions in advance. To begin with, we set the first quantile of historical attitude as the benchmark unchanged for a game version, and focused on the reviews with poor attitude. Quantile is an appropriate way to separate the data set into 4 layers, and the first quantile of attitude represents extremely poor attitude. Poor attitude means that our game reputation has reached a low point from the perspective of players.

Then, we calculated the percentage of useful reviews with poor attitude for every two-week data as a window. The cycle of game maintenance and update is usually once every two weeks, which is also a more suitable time span for the operation team to design and take intervening actions compared to daily and monthly time span.

Finally, we compared the percentage in the deployment window to the median percentages in historical windows. If the percentage of reviews with poor attitude in deployment window is higher than the historical performance, the warning is triggered.

In historical data of $thumbsUpCount > 6$:	**In deployment data of model predicted:**
1.Calculate $\boldsymbol{R_1}$: the first quantile of attitudes (unchanged for a version) 2.Set every two-week data as a window 3.Calculate $\boldsymbol{B_n}$: the number of reviews whose attitude is lower than $\boldsymbol{R_1}$ in $\boldsymbol{Window_n}$ 4.Calculate $\boldsymbol{A_n}$: the number of reviews in $Window_n$ 5.Calculate $\boldsymbol{P_n = B_n/A_n}$: the percentage of the review number with poor attitude in $\boldsymbol{Window_n}$ 6.Set the median $\boldsymbol{M}$ of $\boldsymbol{P_1}$, …, $\boldsymbol{P_n}$ as the triggered line	7.Calculate $\boldsymbol{P = B/A}$ in deployment data set (a window) 8.If $\boldsymbol{P > M}$, a warning is triggered. Actions are required to prevent an attitude drop.

Table 5. The algorithm of pre-warning model

The Figure illustrates the steps of how we built the classification model of predicting useful reviews. For data which dated after 14 Oct, we took them in the deployment stage.

And we trained Random Forest and Logistic Regression model based on historical data. Then evaluate them in the measure of confusion matrix or other metrics.

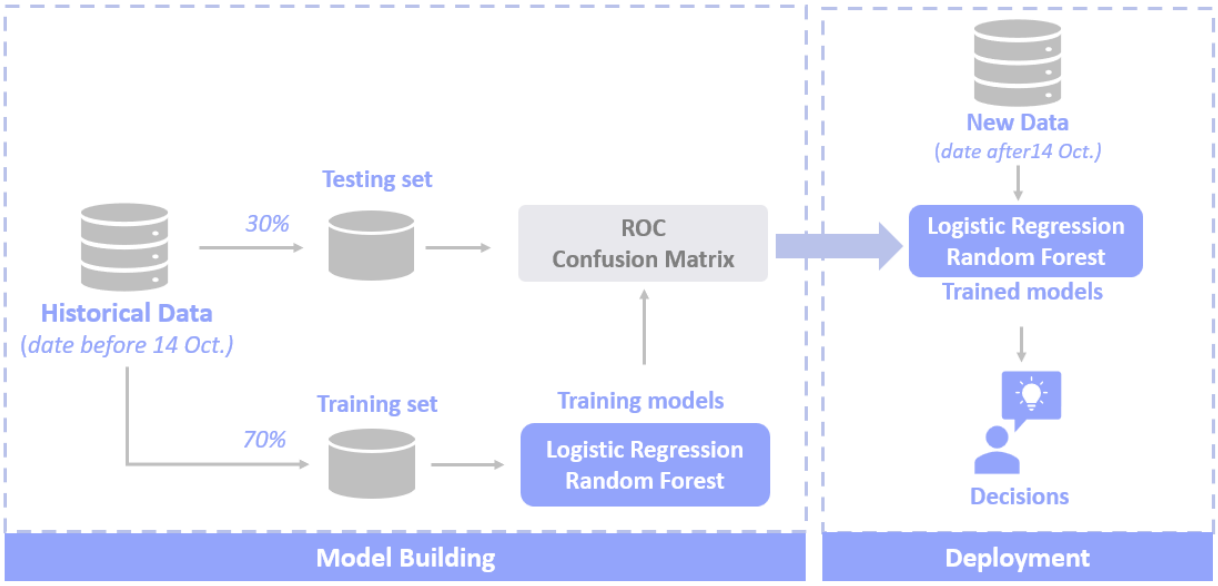

Figure 12. Classification model building pathway

9.2 Random Forest Classification

We first applied Random Forest for classification. The result shows that Length, Days of comments and subjectivity are the top 3 most key features of deciding the usefulness of a comment. Also, the accuracy score tends to be 0.783.

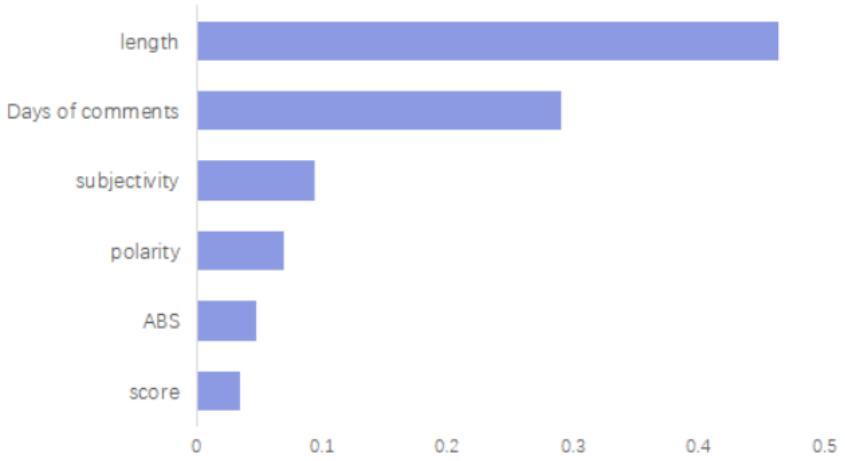

Figure 13. Importance of the variables

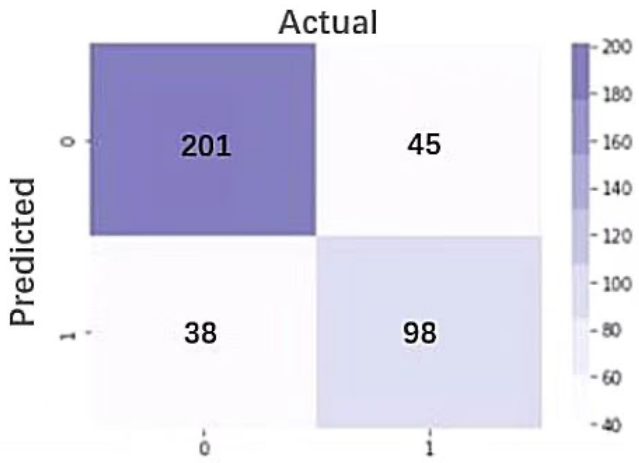

Figure 14. Confusion matrix of Random Forest model

9.3 Logistic Regression

In addition to random forest, we also choose logistic regression method as other classifier. In model 1, usefulness is regarded as dependent variables, while score, subjectivity, polarity, length, ABS and Days of comments are used as independent variables.

Variable	Variable Name
X1	score
X2	subjectivity
X3	polarity
X4	length of comment
X5	ABS
X6	days of comment

Table 6. Variables used in Logistic Regression

The regression result of model 1 is shown in Figure 15:

```
Call:
glm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6, family = binom
ial(),
    data = review)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3991  -0.7117  -0.3141   0.8794   2.8516

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.2271434  0.4549186  -9.292  < 2e-16 ***
x1           0.0507355  0.0692573   0.733 0.463824
x2           0.3966685  0.6275227   0.632 0.527310
x3           0.3545991  0.4967350   0.714 0.475314
x4           0.0485059  0.0035548  13.645  < 2e-16 ***
x5           0.4109869  0.1057496   3.886 0.000102 ***
x6          -0.0007599  0.0007217  -1.053 0.292379
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1179.80  on 890  degrees of freedom
Residual deviance:  858.15  on 884  degrees of freedom
AIC: 872.15

Number of Fisher Scoring iterations: 5
```

Figure 15. Regression result of model 1

And then we got the equation of model 1:

$$\log\frac{p}{1-p} = -4.227 + 0.05 \times score + 0.4 \times subjectivity + 0.354 \times polarity + 0.048 \times length + 0.41 \times ABS - 0.001 \times Days\ of\ Comment$$

Next, we looked at how well each independent variable fits into model1 using a marginal model plot. Marginal model plot can be used to judge the fit degree of independent variables and regression model. The red line represents the model, and the blue line represents each independent variable. If the red line and the blue line coincide, no transformation is required for the independent variable.

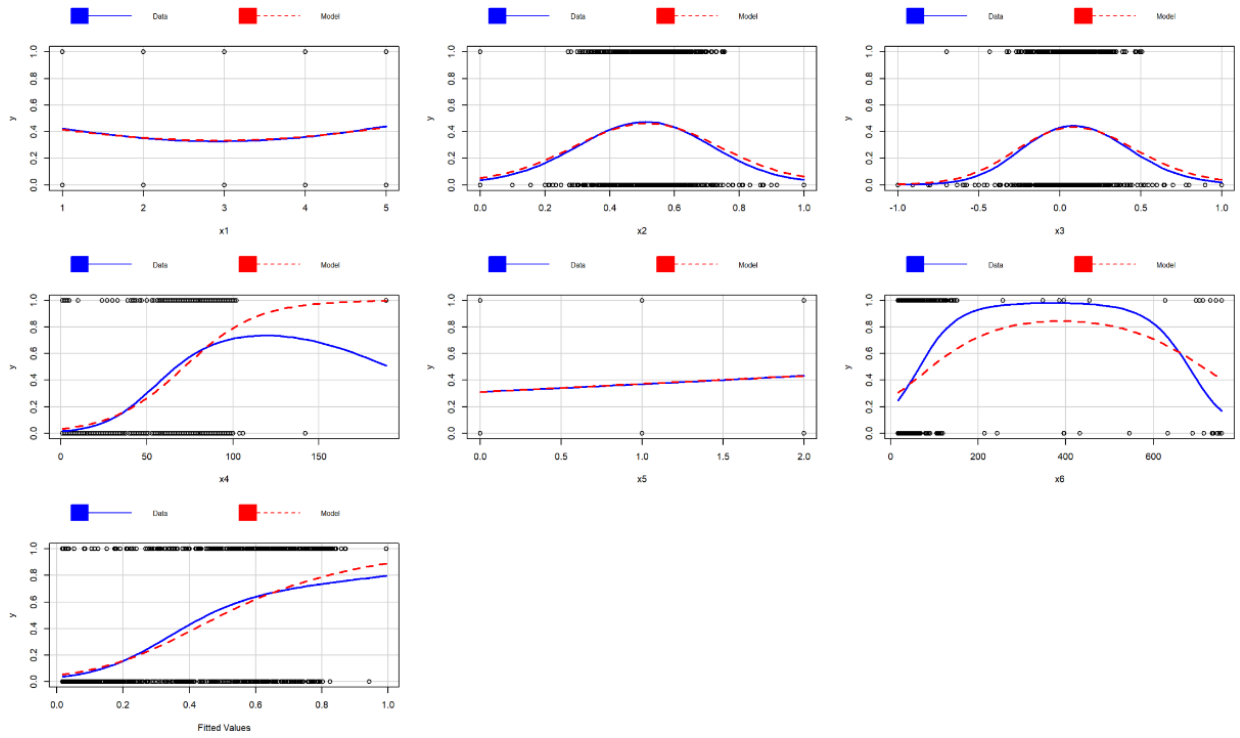

Figure 16. Marginal model plots of model 1

In the results of model 1, the fit effect of x4 and x6 was not good, so we calculated the skewness of these two indicators, respectively.

Variable	Skewness
X4	-0.23
X6	5.41

Table 7. Skewness of variables

X6 had the problem of right skewness, so we did log transform for this indicator and added it to the new independent variable, and then we got model 2.

The regression result of model 2 is shown in Figure 17:

```
Call:
glm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + log(x6), family = bi
nomial(),
    data = review)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2370  -0.6757  -0.3066   0.7902   2.8805

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.091713   0.939395  -9.678  < 2e-16 ***
x1          -0.012352   0.071164  -0.174    0.862
x2           0.281186   0.637691   0.441    0.659
x3           0.465797   0.511655   0.910    0.363
x4           0.043648   0.003670  11.894  < 2e-16 ***
x5           0.521120   0.110761   4.705 2.54e-06 ***
x6          -0.008871   0.001568  -5.656 1.55e-08 ***
log(x6)      1.547165   0.255088   6.065 1.32e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1179.80  on 890  degrees of freedom
Residual deviance:  818.84  on 883  degrees of freedom
AIC: 834.84

Number of Fisher Scoring iterations: 5
```

Figure 17. Regression result of model 2

And then we got the equation of model 2:

$$log\frac{p}{1-p} = -9.092 - 0.012 \times score + 0.281 \times subjectivity + 0.466 \times polarity + 0.044 \times length + 0.521 \times ABS - 0.009 \times Days\ of\ Comment + 1.547 \times log(Days\ of\ Comment)$$

From model 1 to model 2, we can see that the fit of x6 has been improved.

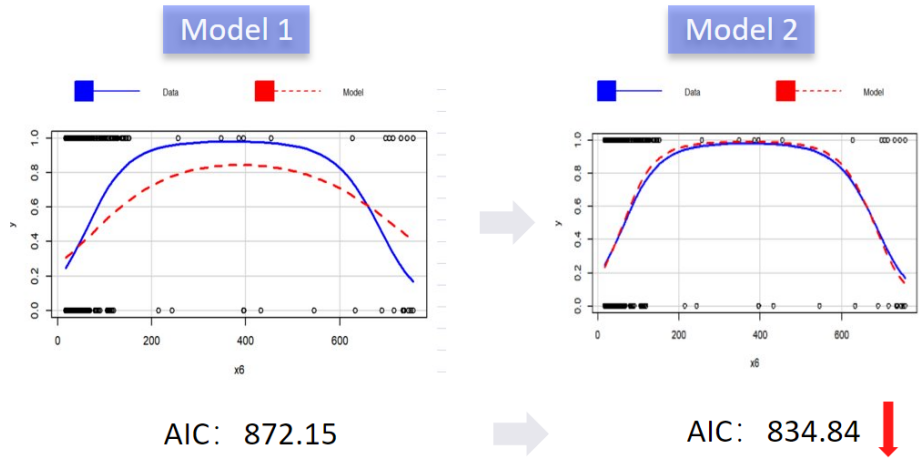

Figure 18. Marginal model plot of model 2

We then transformed x4 to add the square of x4 to the logistic regression independent variables, and we got model 3.

The regression result of model 3 is shown in Figure 19:

```
Call:
glm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + log(x6) + I(x4^2),
    family = binomial(), data = review)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0802  -0.7146  -0.2469   0.8022   3.0564

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.3864693  0.9496476  -9.884  < 2e-16 ***
x1          -0.0101486  0.0722868  -0.140  0.88835
x2           0.0818581  0.6792084   0.121  0.90407
x3           0.5043766  0.5388696   0.936  0.34928
x4           0.0719140  0.0118709   6.058 1.38e-09 ***
x5           0.5191741  0.1107629   4.687 2.77e-06 ***
x6          -0.0081280  0.0015208  -5.345 9.07e-08 ***
log(x6)      1.4610007  0.2527319   5.781 7.43e-09 ***
I(x4^2)     -0.0002414  0.0000920  -2.624  0.00869 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1179.80  on 890  degrees of freedom
Residual deviance:  812.39  on 882  degrees of freedom
AIC: 830.39

Number of Fisher Scoring iterations: 5
```

Figure 19. Regression result of model 3

And then we got the equation of model 3:

$$\log\frac{p}{1-p} = -9.386 + 0.082 \times score + 0.082 \times subjectivity + 0.504 \times polarity + 0.072 \times length + 0.519 \times ABS - 0.008 \times Days\ of\ Comment + 1.461 \times log(Days\ of\ Comment) - 0.0002 \times length^2$$

The x4's fit has also improved from model 2 to model 3.

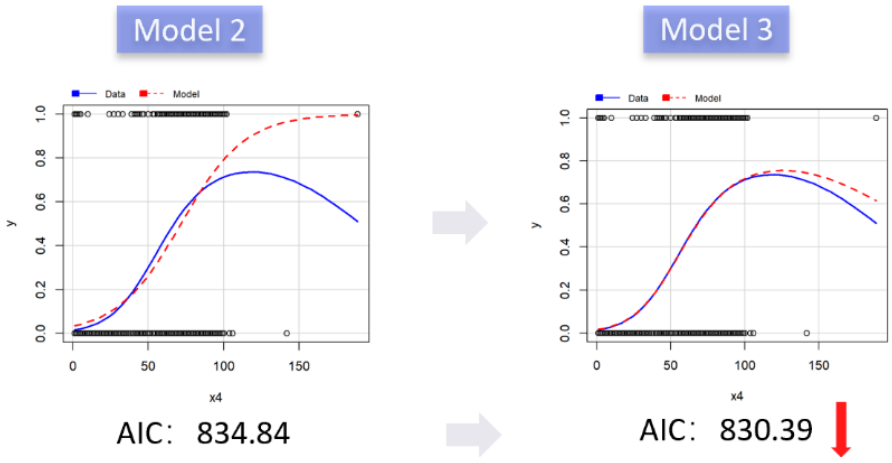

Figure 20. Marginal model plot of model 3

The next step was to remove the insignificant variables. According to the p values of the regression results of model 3, we removed score, polarity and sensitivity. We used the step function for the model 3 and then got the model 4.

The regression result of model 4 is shown in Figure 21:

```
Call:
glm(formula = y ~ x4 + x5 + log(x6) + x6 + I(x4^2), family = binomial(),
    data = review)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0835  -0.7338  -0.2492   0.7885   3.0364

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.3681123  0.9077063 -10.321  < 2e-16 ***
x4           0.0721989  0.0118078   6.115 9.68e-10 ***
x5           0.5221103  0.1103544   4.731 2.23e-06 ***
log(x6)      1.4647592  0.2502500   5.853 4.82e-09 ***
x6          -0.0081322  0.0014982  -5.428 5.70e-08 ***
I(x4^2)     -0.0002446  0.0000916  -2.671  0.00757 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1179.80  on 890  degrees of freedom
Residual deviance:  813.43  on 885  degrees of freedom
AIC: 825.43

Number of Fisher Scoring iterations: 5
```

Figure 21. Regression result of model 4

And then we got the equation of model 4:

$$\log\left(\frac{p}{1-p}\right) = -9.368 + 0.072 \times length + 0.522 \times ABS - 0.008 \times Days\ of\ Comment + 1.464 \times \log(Days\ of\ Comment) - 0.0002 \times length^2$$

From model 3 to model 4, the AIC of the model decreases, and the independent variables of model 4 are all significant. At this point, we have completed all the variable selection steps.

We put all the models together and compared them. From model 1 to model 4, AIC value gradually decreases.

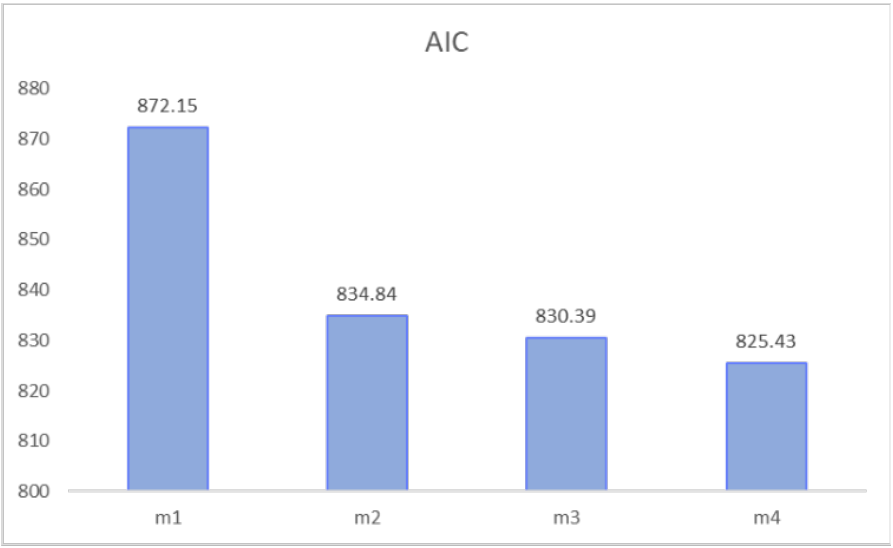

Figure 22. AIC of models

Meanwhile, we drew the ROC curve. It can be seen from the Figure 23 that model 4 has the maximum AUC value.

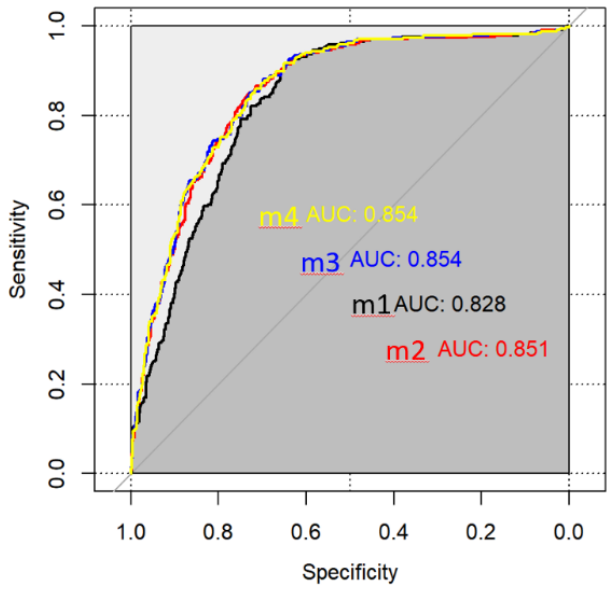

Figure 23. ROC Curve of models

Therefore, model 4 has the best model fitting effect, and we choose model 4 as the logistic regression model.

To calculate the accuracy of model 4, we drew the roc curve of the training set and got the optimal threshold point of 0.378, which was applied to the model.

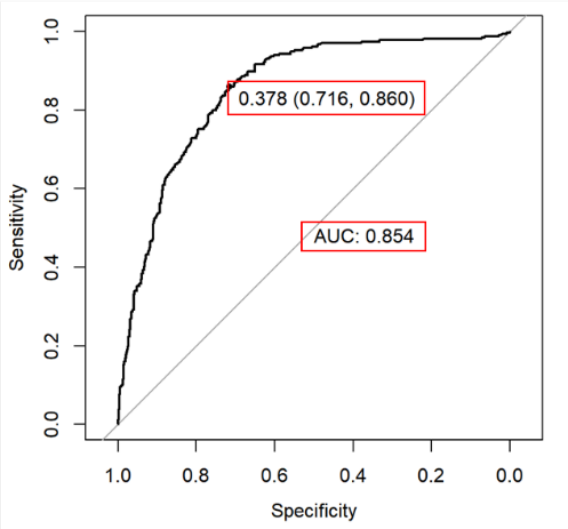

Figure 24. ROC Curve of train set in model 4

Then we predicted the test set and calculated the confusion matrix.

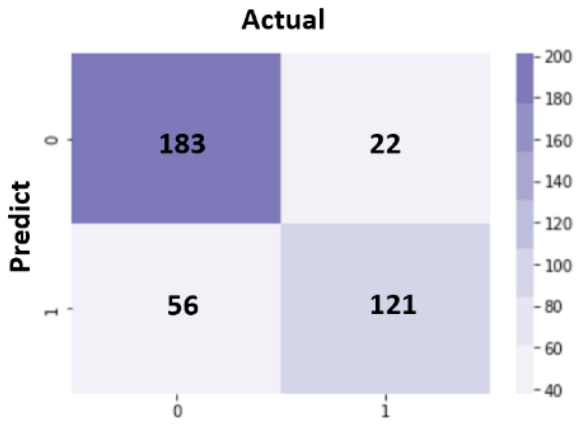

Figure 25. Confusion matrix of test set in model 4

The accuracy score of model 4 is 0.796, and the sensitivity is 0.846.

9.4 Final Model

Now we have two classification models, how to decide the final label for each comment? Since the operation team put a higher penalty on misclassifying useful comments as useless, we compare the leakage rate of useful comments between the models.

$$Leakage\ Rate = 1 - Recall = \frac{FN}{TP + FN}$$

For logistic model, 15.38% useful comments are classified as useless, for random forest model, 31.47% useful comments are classified as useless. However, if taking the union of the classification results of both models, the rate dropped to 13.9%.

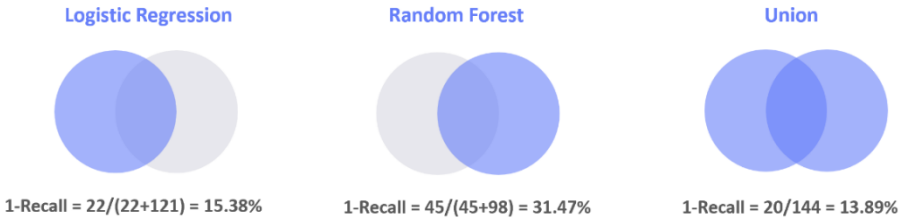

Figure 26. Leakage rate of different models

We union the results of the two models to label the comments. If either model classified a comment as useful, we regard that comment as useful. The operation team regard missing the timing to intervene as more costly.

Correspondingly, we construct a cost function: -50 before False Negative and -10 before False Positive.

$$Cost\ Function = (-50) \times\ FN + (-10) \times FP$$

9.5 Model Deployment

We utilized the deployment data set prepared before to illustrate the mechanism of the pre-warning model (Table 5). The result demonstrates that there are 15% of reviews with poor attitude in deployment window, which is higher than the median of historical performance (8%). Thus, the pre-warning is triggered, and maintenance actions are required to prevent the attitude drop (Figure 27).

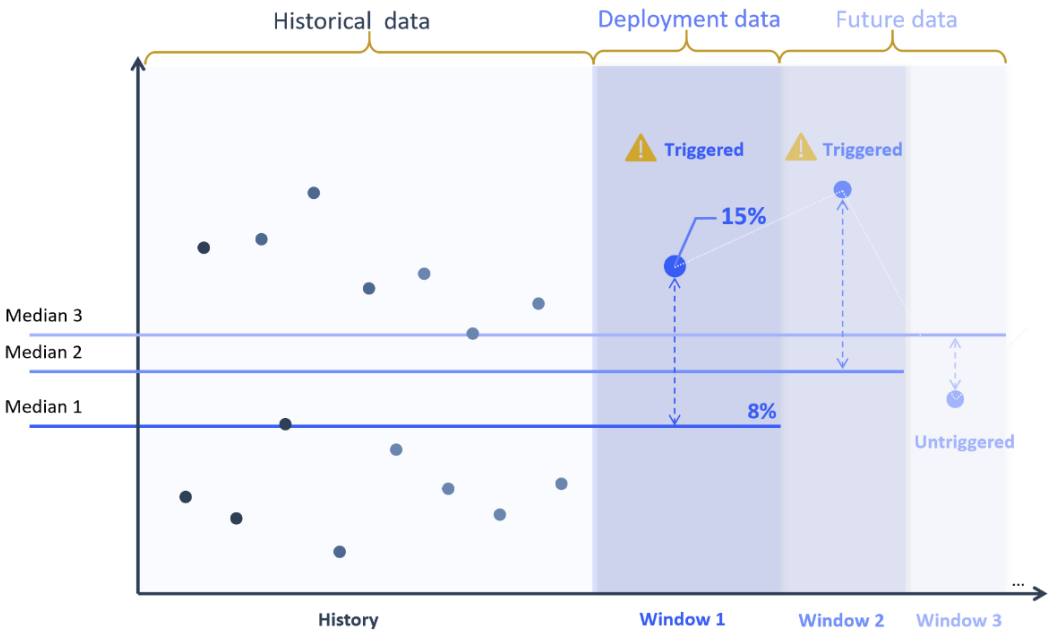

Figure 27. Comparison on percentage of the useful reviews with poor attitude

10 Explorative Analysis – Time Series of Social Impact

In this section, besides the three technical objectives, we also use Google trends to figure out Genshin's popularity in society and generate suggestions for better operations based on Google trends time Series Analysis.

10.1 Correlation Matrix

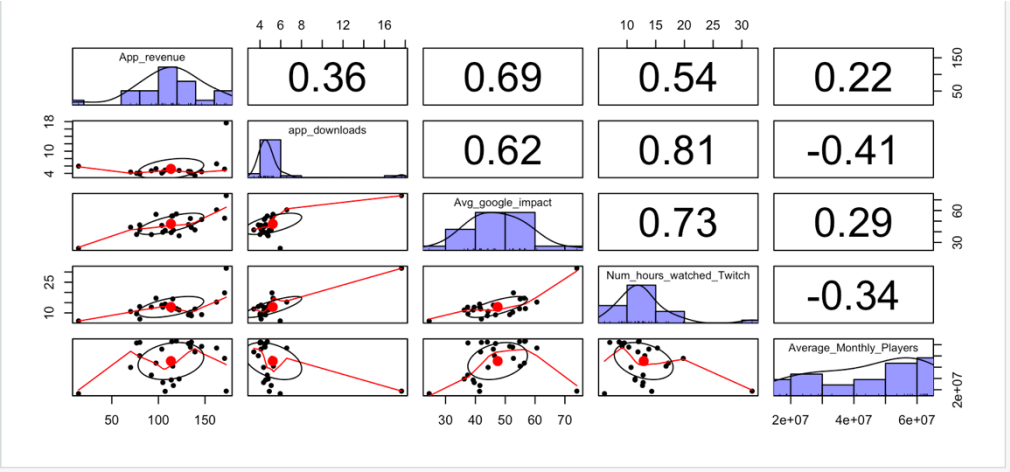

Figure 28. Correlation matrix

To determine whether we should use Google Trends to explore Genshin's popularity in society, we found its correlations with other indicator variables that might represent Genshin's popularity

in society, including variables app revenue, app downloads, hours people spent watching Twitch, and the average number of monthly players. From correlation matrix (Figure 28), we can find that Google impact has a strong and positive correlation with app revenue, app download, and the average number of hours players watch Twitch. Since Google trends has a strong correlation with those indicator variables of Genshin's popularity, we use Google trends information to make further analysis and prediction.

10.2 Lag Plot and Decomposition

A lag plot was used to help evaluate whether the values in a dataset or time series are random.

Figure 29. Lag plot

From the above lag plot (Figure 29), we can find that the data set or time series is random as it does not exhibit any identifiable structure or pattern. Also, there is a small autocorrelation in the data because the data is clustered around the diagonal, and there are outliers.

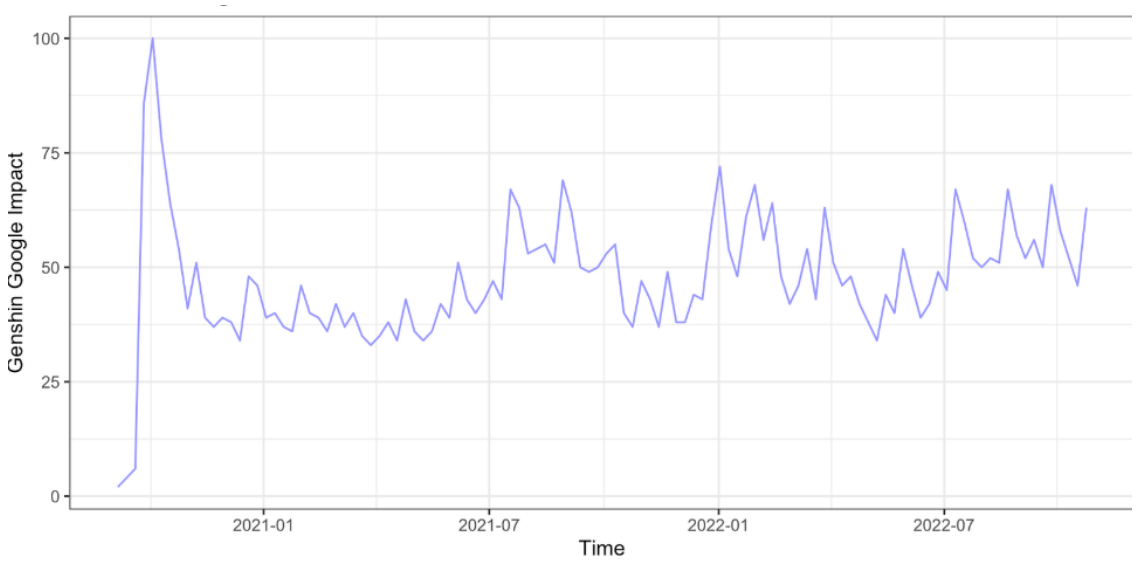

Figure 30. Google Trends time series of Genshin Impact

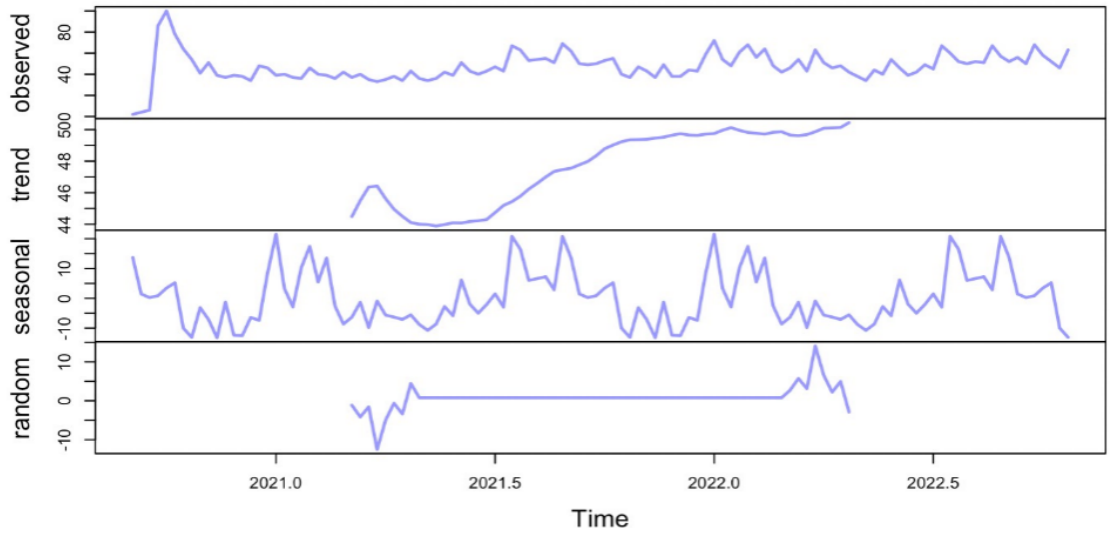

Figure 31. Decomposition of Additive Time Series

Through the time series plot (Figure 30) for Google Trends above, we find that seasonal fluctuations of time series are basically unchanged over time, so we choose the additive model to decompose one time series into multiple series with a combination of level, trend, seasonality, and noise components and obtained the decomposition plot below (Figure 31). We find there is an increasing trend. Also, there seems to be some seasonality in terms of years.

10.3 Seasonal Difference of Google Trend

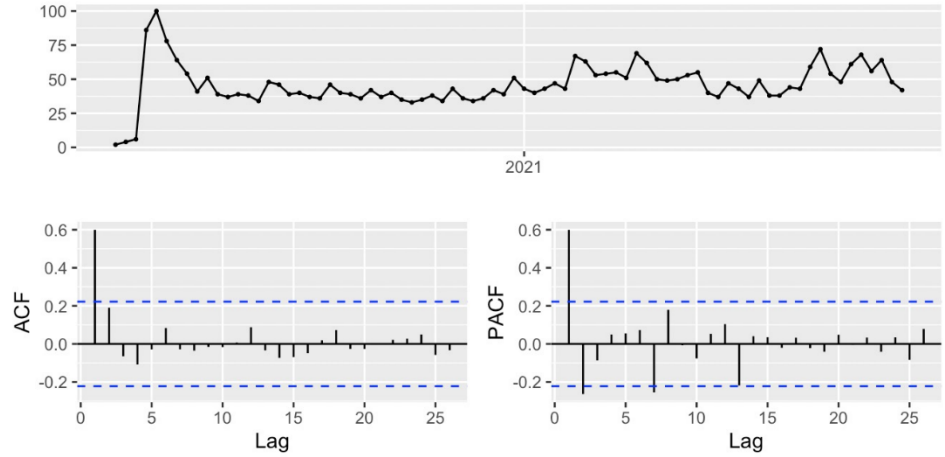

Figure 32. ACF & PACF plots of Google Trends time series

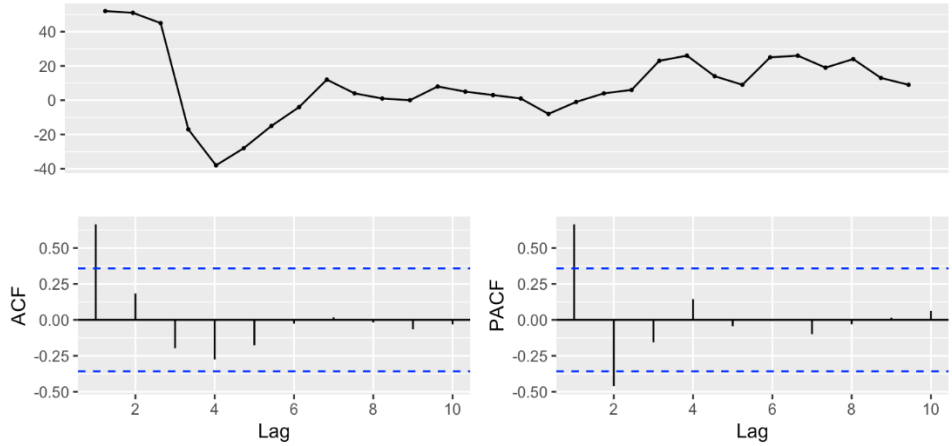

Figure 33. ACF & PACF plots of Google Trends time series with seasonal difference=1

For model building, instead of using train/test split to split the data randomly, as there is no dependence from one observation to the other, we want to use values at the rear of the dataset for testing and everything else for training. The "training" data set is the general term for the samples used to create the model, while the "testing" data set is used to qualify performance. We have 112 observations records at weekly intervals, so we keep the first 78 observations for training and the last 34 observations for testing, and we plot a time series for Google trends along with its ACF and PACF for model building.

From the original time series plot (Figure 32), it shows a little seasonality, but it is not obvious, and it might become obvious with more data. So, a seasonal difference analysis is added for further analysis (Figure 33)

10.4 ARIMA Model for Google Trends

Model complexity

Small

Large

	ARIMA Model	AIC	RMSE (training)	RMSE (testing)	p-value (Ljung-Box test)
1 variable	(1,0,0)(0,0,0)	610.65	11.626	8.701	0.0962
	(0,0,1)(0,0,0)	614.93	11.96	9.647	0.0385
2 variables	(1,0,0)(1,0,0)	608.69	10.98	8.333	0.1978
	(0,0,1)(1,0,0)	614.97	11.622	9.494	0.0387
	(1,0,1)(0,0,0)	608.17	11.290	9.466	0.0360
	(1,0,1)(0,1,0)	245.3	7.906	NA	0.1123
	(2,0,0)(0,0,0)	607.41	11.235	9.484	0.0361
3 variables	(1,0,1)(1,0,0)	605.28	10.480	9.063	0.1807
	(2,0,1)(0,0,0)	608.82	11.190	9.447	0.0280

Table 8. Time series ARIMA model result

Here, we tried and ran different Arima models with or without seasonal differences at different model complexity levels. From model result (Table 8), we can figure out that ARIMA model with seasonal difference (1,0,1) (0,1,0) gives the best result with the smallest AIC. However, the data was too small to be tested. Also, with the increase of testing data, the training becomes less, and the seasonality will be less significant, so there is no need to do seasonal difference.

Principles of model selection:

1. The smaller the AIC and test RMSE, the better
2. The less complexity the model, the better

From the overall comparison of RMSE for both training and testing data and AIC, we find that although ARIMA (1,0,0) (1,0,0) result is not the best in training data, it is the best in testing and the model complexity is small. On the other hand, we learned that if the test is significant (P<0.05), the model is inadequate, then we need to go back and consider the other Arima Model. Here for ARIMA (1,0,0) (1,0,0) result, the test is insignificant with p-value = 0.1978, which indicates that the model is adequate and is good enough.

10.5 Google Trends Forecast Results

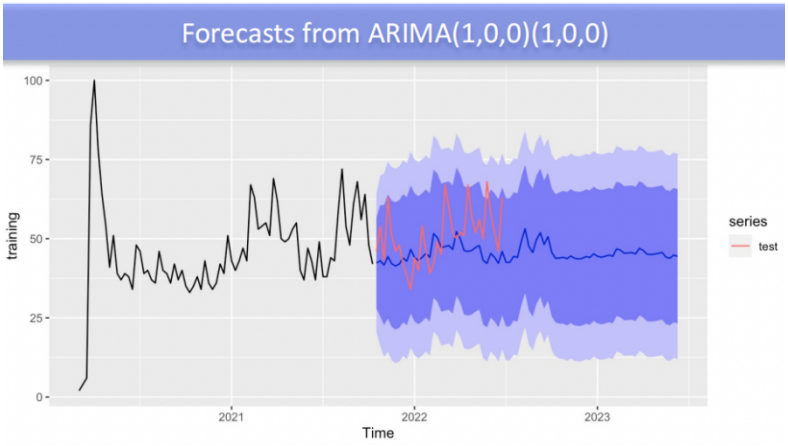

Figure 35. Time series forecast

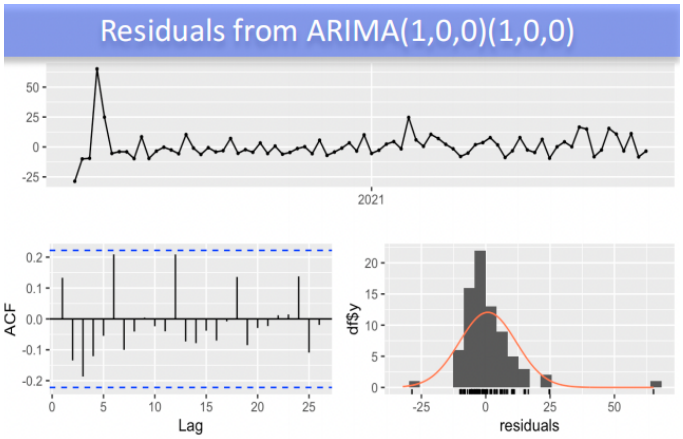

Figure 36. Time series residual plot

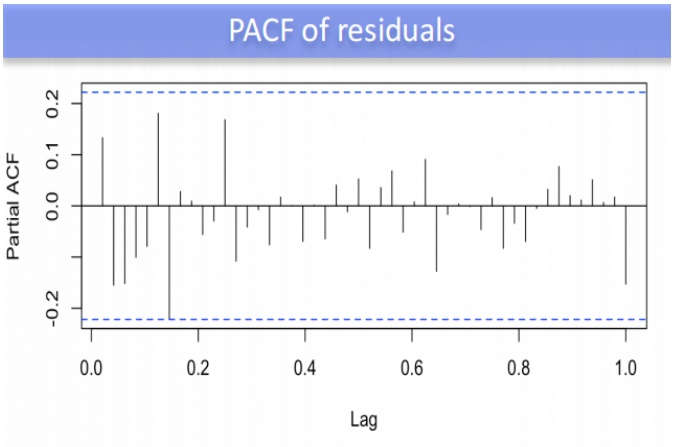

Figure 37. Time series residual (PACF)

Here, we checked the model result (Figure 35) and residual (Figure 36, 37) and conclude that:

1) The residuals are distributed evenly around 0.

2) From the ACF and PACF, anything within the blue area (two blue dash lines) is statistically close to zero and within 95% significance threshold. All the autocorrelation coefficients lie within these limits, confirming that the data are white noise.

3) Residual plots are seemingly normally distributed.

4) So, this is a good model for further prediction and forecast.

Google trends are an indicator of searching popularity. If the game is widely discussed/searched in society, the game is regarded as more popular. From the results of our Google trends forecast for the next six months, we can see a significant decline in Google trends from around October 2022, but a leveling off over the next six months. Overall, we suggest that when the operation team finds that the whole society is paying high attention to the game, it should be a good timing for the game to attract new players.

11 Suggestions

Across the whole analysis process, we have derived several suggestions for Genshin Impact according to business objectives.

To prevent the attitude drop, initially we suggest that Genshin Impact endeavors to optimize issues complained mostly by players, including system bugs, time for service, game loading and storyline, and occupying a large amount of storage.

Furthermore, we propose to update the auto-reply corpus by focusing more on topics regarding Graphic and story design. The sentiment polarity distribution (55% positive and 45% negative) can also serve as a guide for the operation team. As for the timing of interfering, when the percentage of useful reviews with poor attitude in the deployment window is higher than that of the median level in history, the warning is triggered, and the operation team should take the maintenance actions.

Besides, from the perspective of social reputation, when there is a growing attention to the game in the whole society, the operation team can seize the chance to attract new players.

Lastly, we recommend new games to focus on elaborating design of characters and storylines, which are mostly appreciated by the players.

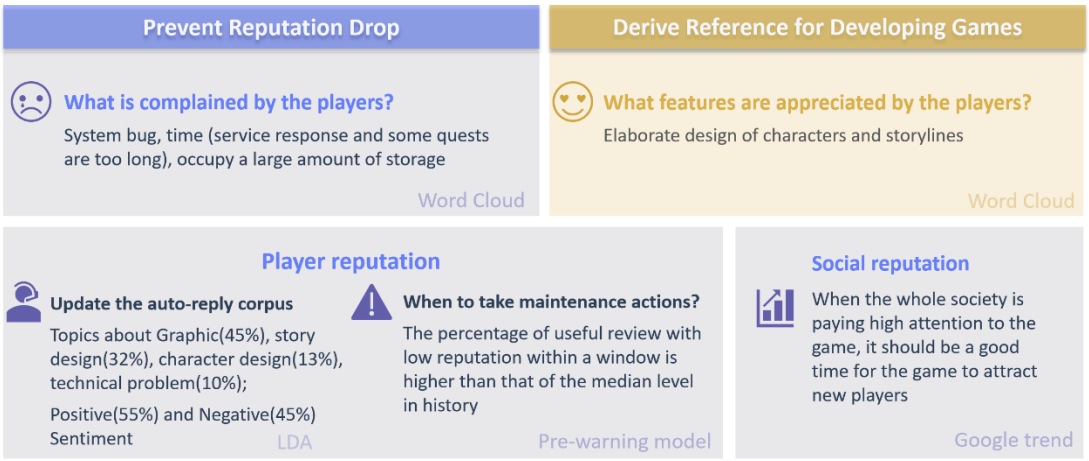

Figure 38. Suggestion summary

12 Future Recommendations

- Google Play Store displays a limited number of comments every day. If the data collection could be conducted at a higher frequency (e.g, twice a week), then there would be more comments for better analysis and prediction.
- The coverage of windows and thresholds of the pre-warning model could be recalibrated according to the feedback from the operation team after the initial deployment of the model.

References

Top grossing mobile games worldwide for December 2020. Sensor Tower - Market-Leading Digital & Mobile Intelligence. (n.d.). Retrieved October 15, 2022, from https://sensortower.com/blog/top-mobile-games-by-worldwide-revenue-december-2020

Top grossing mobile games worldwide for December 2021. Sensor Tower - Market-Leading Digital & Mobile Intelligence. (n.d.). Retrieved October 15, 2022, from https://sensortower.com/blog/top-mobile-games-by-worldwide-revenue-december-2021

Top grossing mobile games worldwide for June 2022. Sensor Tower - Market-Leading Digital & Mobile Intelligence. (n.d.). Retrieved October 15, 2022, from https://sensortower.com/blog/top-mobile-games-by-worldwide-revenue-june-2022

Appendices

Appendix 1 – Comments Data Dictionary

Variables	Type		Description
reviewID	Categorical	Char	ID of the comments
content	Categorical	Char	Comment text
score	Numerical	Integer	Players' rating of the game
thumbsUpCount	Numerical	Integer	number of likes from other players for each comment, quantile 1(25%) is 6
reviewCreatedVersion	Categorical	Char	Game version
at	Day Time	Day Time	time of the review created
replyContent	Categorical	Char	game's official reply to players' comments
repliedAt	Numerical	Decimal	time of the game official's reply
polarity	Numerical	Decimal	polarity of players' comment based on sentiment scoring
subjectivity	Numerical	Decimal	subjectivity of players' comment based on sentiment scoring
ABS	Numerical	Integer	=|Score – average Score|
attitude	Numerical	Decimal	Calculated from score and sentiment
review_length	Numerical	Integer	the length of the review (count of words)
usefulness	Categorical	Char	binary variable, 1 if noteworthy comment, and 0 if not noteworthy (just normal)
days of comment	Numerical	Integer	How many days the review has been published

Appendix 2 – Google Trends Data Dictionary

Variables	Type		Description
week	Date	Date	Weekly data from Sep 2020 to Oct 2022
google_impact	Numerical	Integer	monthly App revenue from Sep 2020 to Oct 2022 for Genshin

Appendix 3 – Performance Data Dictionary

Variables	Type		Description
google_impact	Numerical	Integer	indicator of searching popularity
app_revenue	Numerical	Decimal	monthly App revenue from Sep 2020 to Oct 2022 for Genshin
app_downloads	Numerical	Decimal	monthly App downloads from Sep 2020 to Oct 2022 for Genshin
avg_google_impact	Numerical	Integer	monthly google trends from Sep 2020 to Oct 2022 for Genshin
num_hours_watched_Twitch	Numerical	Decimal	average number of hours players watch Twitch
average_Monthly_Players	Numerical	Integer	monthly players from Sep 2020 to Oct 2022 for Genshin

Appendix 4 – Gantt Chart