Recommendations on Steam Games

Taotao Jiang tj2441

Xinyu He xh2469

Xiaohang He xh2509

Abstract

In this paper, we show the whole process of designing and building the Steam platform game recommendation system. We used api to collect data, machine learning models to analyze and predict data, LDA analysis to analyze the results, and HTML5 and JavaScript to build a web interface.

Each of our models has achieved good accuracy. The generated recommended game list and genres are also highly in line with market perceptions.

1. Introduction

As the main form of entertainment for young people, video games have a huge market in today's society. The research object of our project is Steam, which is the largest digital distribution platform for PC gaming, holding around 75% of the market share. The aim of our project is to provide users with predictions and recommendations about video games on Steam.

The existing game recommendation system on the Steam platform is not effective enough. The system pays too much attention to sales, causing some high-quality games to be unable to enter the recommended list because of low sales due to just being launched or low publicity. Not only does the good review rate have a low weight, but also some high-quality games that are not obsolete cannot appear on the recommended homepage because they are not published in recent years.

Our goal is to construct a new system of games recommendation which is influenced by many features. Our system is built based on the data obtained from Steamspy, a website that streams data from Steam.

We constructed four submodels which are popularity, penetration, playability, and feedback as the considered components in our machine learning model in order to make more precise recommendations for users. The data collection block and model training and predicting block are triggered at 2am everyday by Airflow. The generated recommendation list data is linked directly to our web interface. On the web interface, users can choose their desired genre of video games to get a list of suggested games. These recommended games are generated by the score system constructed with our models and algorithms. Also, the score system will also contribute to LDA topic analysis, which will give suggestions to game developers about the popular genres of future games.

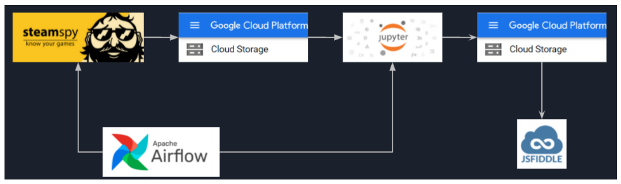


Figure 1: System Architecture

Our recommendation list is in line with the psychological expectations of most players in the market. Our list is similar to the official Steam list, but it is obviously more usable. For example, some series of games with high sales but not good reputation have not appeared in the forefront of our list. In addition, our topic analysis also accurately analyzed the recent popular game types. With our system, users can buy games of a certain theme they want to play, and there is no need to worry about wasting money and time on a bad game.

2. Related Work

Our project encompasses three main goals:

1) game sales prediction;
2) game popularity forecast;
3) recommendation system construction.

We read ten published works, each of which concentrates on segment-analysis in the gaming industry, through journals, books and webpages. However, our project serves as a synthetic study spanning over the above 3 fields.

In 2017, Jruots managed to predict the video game sales with advanced algorithms such as neural networks and gradient boosting with an accuracy score of ~0.5. Our team decides to improve the efficacy by doing more feature engineering and incorporating time series models to capture short-term data trends.

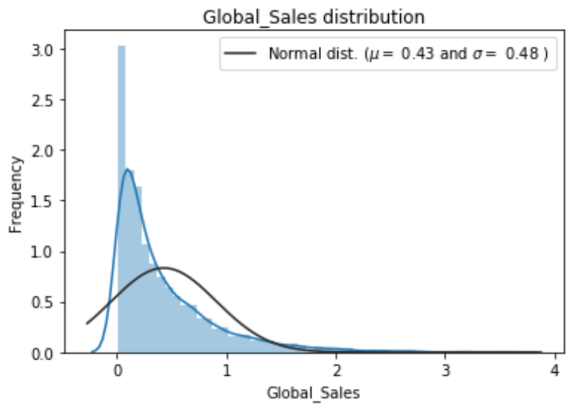


Figure 2: Game Sales Distribution

In terms of game popularity, there's a paper- "Game Popularity Tracking System" (Joseph Alexander, 2018) utilizing heteroscedastic hierarchical folded normal models for predictions. The research is credible in the definition of game popularity. Instead of using the number of players, the simplest and most intuitive operationalization of game popularity in a set period of time, Joseph combined four other metrics to calculate game popularity: user count, unique page views, average time on the page, the difference of unique page views, and average time on the page from the day before. But this paper has some limitations. For instance, it didn't take into account players' feedback, which should be critical in determining the popularity of a game. We plan to include sentiment score as a predictive feature as well by scraping reviews from Twitter.

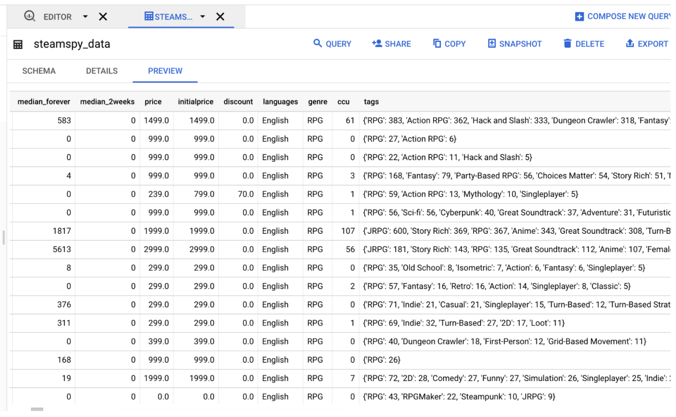
Lastly, as for the recommendation system, most of the past algorithms recommended games based on the game popularity. Although it makes sense to some extent, many good games aren't discovered due to various factors. Our team aims to consider, other than sales, more valuable factors- Originality, Freshness and replayability, etc, as mentioned by Wolfgang, 2000.

3. Data

We did our data collection via the Steamspy API and stored it as a csv file. The data structure collected from the API's are as follows:

3.1. Columns

- Index: 29,235 games
- Appid: id of the game app (used for collecting data from Steamspy API)
- Name: name of the game
- Developer: developer of the game
- Publisher: publisher of the game
- Score_rank: rank of the game
- Positive: number of positive reviews
- Negative: number of negative reviews
- User scores: user score of the game
- Owners: range of number of game owners
- Average_forever: average number of daily active users
- Average_2week: average number of daily active users for the past two weeks
- Median_forever: median number of daily active users
- Median_2week: median number of daily active users for the past two weeks
- Price
- Initial price
- Discount: discount for the game
- Language: language available for the game settings
- Genre: 30+ genres, each game can have multiple genres
- ccu: number of concurrently connected users
- Tags: number of tags in game review/comments, each game can have multiple tags, 400+ unique tags in total



steamspy_data

median_forever	median_2weeks	price	initialprice	discount	languages	genre	ccu	tags
583	0	1499.0	1499.0	0.0	English	RPG	61	{'RPG': 383, 'Action RPG': 362, 'Hack and Slash': 333, 'Dungeon Crawler': 318, 'Fantasy'
0	0	999.0	999.0	0.0	English	RPG	0	{'RPG': 27, 'Action RPG': 6}
0	0	999.0	999.0	0.0	English	RPG	0	{'RPG': 22, 'Action RPG': 11, 'Hack and Slash': 5}
4	0	999.0	999.0	0.0	English	RPG	3	{'RPG': 168, 'Fantasy': 79, 'Party-Based RPG': 56, 'Choices Matter': 54, 'Story Rich': 51, '
0	0	239.0	799.0	70.0	English	RPG	1	{'RPG': 59, 'Action RPG': 13, 'Mythology': 10, 'Singleplayer': 5}
0	0	999.0	999.0	0.0	English	RPG	1	{'RPG': 56, 'Sci-fi': 56, 'Cyberpunk': 40, 'Great Soundtrack': 37, 'Adventure': 31, 'Futuristic
1817	0	1999.0	1999.0	0.0	English	RPG	107	{'JRPG': 600, 'Story Rich': 369, 'RPG': 367, 'Anime': 343, 'Great Soundtrack': 308, 'Turn-B
5613	0	2999.0	2999.0	0.0	English	RPG	56	{'JRPG': 181, 'Story Rich': 143, 'RPG': 135, 'Great Soundtrack': 112, 'Anime': 107, 'Femal
8	0	299.0	299.0	0.0	English	RPG	0	{'RPG': 35, 'Old School': 8, 'Isometric': 7, 'Action': 6, 'Fantasy': 6, 'Singleplayer': 5}
0	0	299.0	299.0	0.0	English	RPG	2	{'RPG': 57, 'Fantasy': 16, 'Retro': 16, 'Action': 14, 'Singleplayer': 8, 'Classic': 5}
376	0	299.0	299.0	0.0	English	RPG	0	{'RPG': 71, 'Indie': 21, 'Casual': 21, 'Singleplayer': 15, 'Turn-Based': 12, 'Turn-Based Strat
311	0	299.0	299.0	0.0	English	RPG	1	{'RPG': 69, 'Indie': 32, 'Turn-Based': 27, '2D': 17, 'Loot': 11}
0	0	399.0	399.0	0.0	English	RPG	0	{'RPG': 40, 'Dungeon Crawler': 18, 'First-Person': 12, 'Grid-Based Movement': 11}
168	0	999.0	999.0	0.0	English	RPG	0	{'RPG': 26}
19	0	1999.0	1999.0	0.0	English	RPG	7	{'RPG': 72, '2D': 28, 'Comedy': 27, 'Funny': 27, 'Simulation': 26, 'Singleplayer': 25, 'Indie':
0	0	0.0	0.0	0.0	English	RPG	0	{'RPG': 43, 'RPGMaker': 22, 'Steampunk': 10, 'JRPG': 9}

Figure 3: Raw Data in BigQuery

3.2. ColumnsData Preprocessing

- Drop 'na' values
- Response variable: favorable rate = # of positive reviews / (# Positive + # Negative)
- "Genre" column: transforming the genre column into one-hot "genre_ xxx" columns (if one genre is mentioned for one game, the genre_xxx column for that game will be

recorded as 1, otherwise 0)
- "Language" column: transforming the language column into one-hot "language_xxx" columns
- "Tags" column: unfolding the dictionary-liked number of times a tag was mentioned in the reviews into individual "tag_xxx columns", if a tag is not mentioned for a game, the cell entry should be 0

3.3. Data Visualization

I.

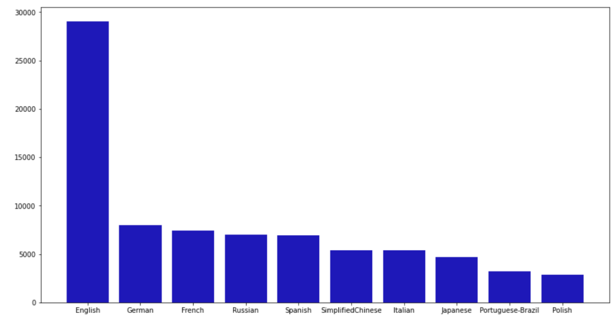


Figure 4: Top 10 Languages of Games

The most common language for Steam Games is English, followed by German, French, Russian, Spanish, ...

II.

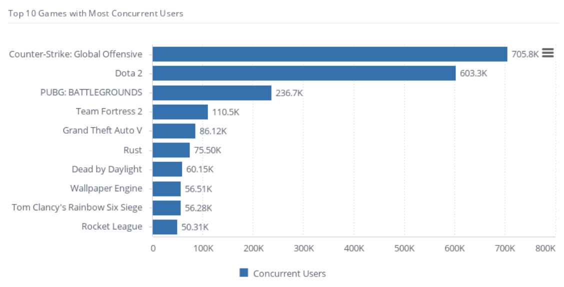


Figure 5: Top 10 Games with Most Concurrent Users

Concurrent users refer to the number of users playing games at the same time. It shows that Counter-Strike is the top performer in this regard, followed by Dota2.

III.

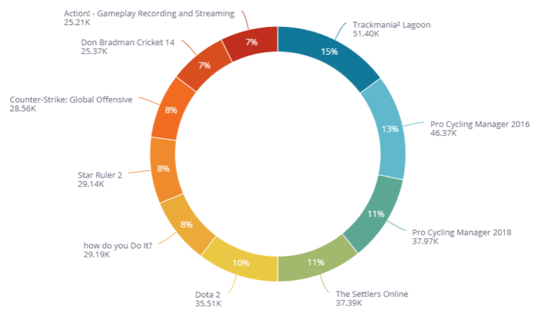


Figure 6: Top 10 Most Frequently Played Games

TrackMania 2: Lagoon is the most frequently played game with around 51.4k minutes played in the last 2 weeks.

4. Methods

Our team decided to examine four dimensions- *popularity, penetration, playability and feedback,* to evaluate game performance. Thus, we constructed four submodels representing the above dimensions, each of which generates a score of 0-25, to design a game scorecard.

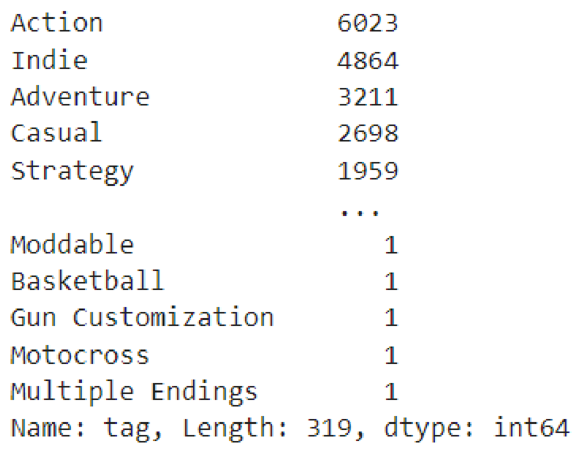
4.1. Feature Engineering

To make the analysis more interesting and robust, we conducted preliminary statistical analysis and feature engineering to add additional variables.

- n_lang: number of languages supported
- favorate_rate: positive/(positive + negative)
- if_dis: dummy variable, if any discount available for the game
- n_owner: number of game owners. It's the average value of the owners range
- top10_dev: dummy variable, if the game is developed by top 10 developers
- top10_ pub: dummy variable, if the game is published by top 10 publishers

4.2. Sentiment Analysis

First, we tried to do sentiment analysis for further twitter data analysis. However, we found the data involves 319 unique tags, making it impossible to conduct twitter analysis.



```
Action                   6023
Indie                    4864
Adventure                3211
Casual                   2698
Strategy                 1959
                         ...
Moddable                    1
Basketball                  1
Gun Customization           1
Motocross                   1
Multiple Endings            1
Name: tag, Length: 319, dtype: int64
```

Figure 7: Sentiment Analysis Results

4.3. Model Design

4.3.1 Optimization Techniques

- Scaling & Normalization: It's important to scale data when dealing with distance-sensitive algos, such as linear regression.
- Principal Component Analysis: Since our dataset has 400+ tags, we utilized PCA to reduce data dimensionality. 10 features are finally kept to account for 90% variance.
- Algo Comparison: our team compared 3 models: linear regression, random forest and gradient boosting regression. We expect boosting algorithms could lift model performance. Specifically, pca and scaling were applied for linear regression, while only pca was adopted for random forest and gradient boosting since tree models aren't sensitive to distance.

4.3.2 Evaluation Metrics

- MAE: mean absolute error is often used in small-scale dataset, which makes it inappropriate to be used in our case. Also, MAE isn't straightforward enough to do comparisons. We try to find more representative metrics.
- SMAPE: SMAPE is symmetric mean absolute percentage error. SMAPE isn't scale dependent and ranges well between 0-100%, easy for comparison. Although this metric treats over-forcast vs under-forecast unequally, it'll eventually be balanced out by normalization. Its formula is as follows.

$$\text{SMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|F_t - A_t|}{|A_t| + |F_t|}$$

4.3.3 Scoring Methodology

Each prediction will be normalized between 0 and 25. Using adjusted Min-Max normalization to score each dimension/submodel between 0-25, and finally adding them up.

i.e. score = 25 * (value - min)/(max - min)

4.4. Submodels & Accuracies

4.4.1 Popularity

- Independent variable: Average_forever (as a proxy for daily active users)
- dependent variables: n_owner, n_lang, if_dis, average_2week, price…
- the baseline random forest has the lowest test smape of ~4%

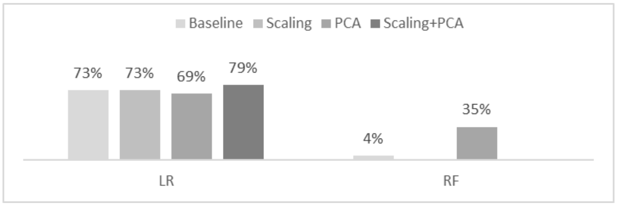


Figure 8: Accuracy Plot for Popularity Model

4.4.2 Penetration

- independent variable: N_owner
- dependent variables: top10_dev, top10_pub, n_lang, average_forever…
- The baseline random forest model shot the best performance of ~19% test smape.

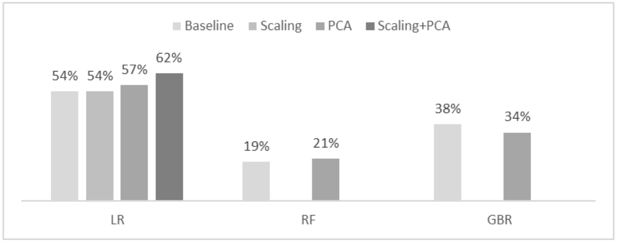


Figure 9: Accuracy Plot for Penetration Model

4.4.3 Playability

- independent variable: CCU (concurrent users)
- dependent variables: price, if_dis, average_2week, median_forever…
- Due to data limitations of ccu, all 3 models didn't get great results. Relatively speaking, the baseline random forest has the lowest smape of ~57%

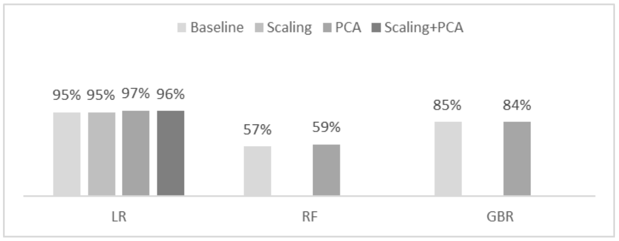


Figure 10: Accuracy Plot for Playability Model

4.4.4 Feedback

- independent variable: Favorable_rate
- dependent variables: ccu, positive, negative, n_lang, if_dis…
- the baseline random forest is the best with the lowest test smape of ~0%

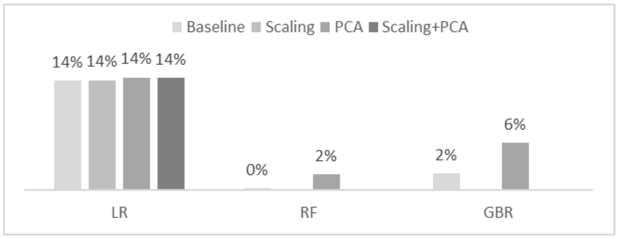


Figure 11: Accuracy Plot for Feedback Model

4.5. Scorecard Result

According to the scorecard analysis, Counter-Strike: Global Offensive ranks first, followed by Dota2. The result will be further improved by incorporating the LDA model in a later stage. Below is an overview of the top 10 recommended games based on the current analysis.

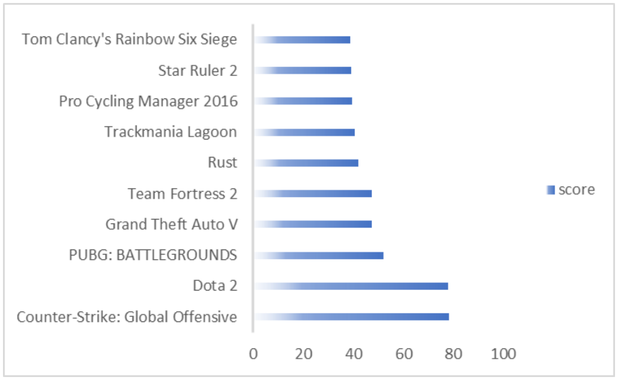


Figure 12: Game Scorecard Result

5. Experiments

5.1. Sentiment Analysis

After investigating the game pool, we recognized ~30k games. It's no easy task to scrape all reviews for these games from Twitter. Even from the perspective of dominating tags, the tag volume is also too big to proceed analysis with sentiment score.

5.2. Construct Algorithm

We tried to experiment with different algorithms with different optimization techniques. The accuracy score was out of our expectation/hypothesis. For example, we would expect gradient boosting should perform better than random forest, which is simply based on bagging algo. Also, scaling didn't help improve the model performance of linear regression, which appears to be unexpected as well facing such varied dataset. The model result showed that baseline random forest reached consistent good performance on the models, except the prediction of playability. Our team contemplated that such consistent performance could be explained by the nature of the dataset- big volatility & many categorical variables.

5.3. Build System Web Interface

To make the project result more accessible to our readers, we designed a web-based game searching platform using HTML5 and JavaScript. The functionality is to offer game rankings by genre. All game recommendations under that genre will pop up once users select a genre they're interested in.

6. System Overview

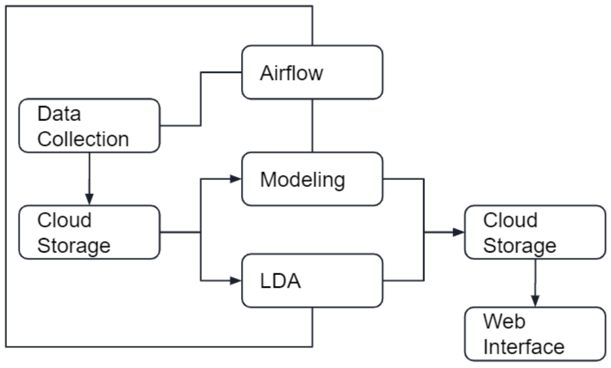


Figure 13: System Structure Overview

6.1. Data Collection

With the python API application in the data collection block, the system will collect data from SteamSpy and do the preprocessing part to modify the raw data. Then the processed data will be stored in Cloud Storage for further analysis.

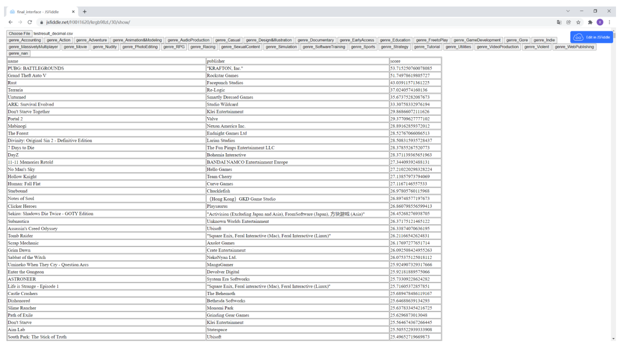
6.2. Modeling

Using the stored data in the cloud storage, the modeling block will first create new dimensions such as 'n_lang', 'favorable_rate'. Then machine learning models of the ‘sklearn’ package will be utilized to train models of four features of our final score. The best model of each feature will be selected among LinearRegression, GradientBoostingRegressor, RandomForestRegressor, and so on. Finally, the modeling block will predict and generate the scorecard result of each game in our dataset and export it to the cloud storage.

6.3. LDA

Parallel to the model module, the LDA module also uses preprocessed data in cloud storage for topic analysis. After generating the new columns of data for LDA analysis, the block will first do clustering for tags of games. Then the LDAModel of the ‘gensim’ package will generate the LDA model which will be visualized by the ‘pyLDAvis’ package.

6.4. Web Interface



name	publisher	score
PUBG: BATTLEGROUNDS	"KRAFTON, Inc."	53.715250760078085
Grand Theft Auto V	Rockstar Games	51.74978619805727
Rust	Facepunch Studios	43.03911571361225
Terraria	Re-Logic	37.0240574160136
Unturned	Smartly Dressed Games	35.67375282087673
ARK: Survival Evolved	Studio Wildcard	33.30758332976194
Don't Starve Together	Klei Entertainment	29.8686607211626
Portal 2	Valve	29.37709627777102
Mabinogi	Nexon America Inc.	28.89162859372012
The Forest	Endnight Games Ltd	28.5276706608651З
Divinity: Original Sin 2 - Definitive Edition	Larian Studios	28.508315935728437
7 Days to Die	The Fun Pimps Entertainment LLC	28.37855267520773
DayZ	Bohemia Interactive	28.371139365651963
11-11 Memories Retold	BANDAI NAMCO Entertainment Europe	27.34409392488131
No Man's Sky	Hello Games	27.210220298328224
Hollow Knight	Team Cherry	27.13857973794069
Human: Fall Flat	Curve Games	27.1167146557533
Starbound	Chucklefish	26.97805760115968
Notes of Soul	（Hong Kong）GKD Game Studio	26.89748577197673
Clicker Heroes	Playsaurus	26.860798556599413
Sekiro: Shadows Die Twice - GOTY Edition	"Activision (Excluding Japan and Asia), FromSoftware (Japan), 方块游戏 (Asia)"	26.45268276938705
Subnautica	Unknown Worlds Entertainment	26.37175121465122
Assassin's Creed Odyssey	Ubisoft	26.33874070636195
Tomb Raider	"Square Enix, Feral Interactive (Mac), Feral Interactive (Linux)"	26.21166542624831
Scrap Mechanic	Axolot Games	26.17697277651714
Grim Dawn	Crate Entertainment	26.092508424955263
Sabbat of the Witch	NekoNyan Ltd.	26.075375125018112
Umineko When They Cry - Question Arcs	MangaGamer	25.924907329317666
Enter the Gungeon	Devolver Digital	25.92181889575066
ASTRONEER	System Era Softworks	25.73309228624282
Life is Strange - Episode 1	"Square Enix, Feral interactive (Mac), Feral Interactive (Linux)"	25.71605372857851
Castle Crashers	The Behemoth	25.689478486119167
Dishonored	Bethesda Softworks	25.64688639134293
Slime Rancher	Monomi Park	25.637833454216725
Path of Exile	Grinding Gear Games	25.6296873013048
Don't Starve	Klei Entertainment	25.564674367266445
Aim Lab	Statespace	25.505522939333908
South Park: The Stick of Truth	Ubisoft	25.49652719669873

Figure 14: Web Interface Screenshot

Directly connected to the scorecard result stored in the cloud storage, the web interface can show users the recommended game lists of different genres. The input button is generated by the ‘button’ function of HTML5. Each time the user selects a desired genre, the data filtering and sorting process is implemented through JavaScript and the related list is finally presented on the web page.

6.5. Airflow

Airflow is the platform that allows us to schedule, manage and trigger each block of our workflows. We schedule the whole workflow triggered at 2am everyday. It will first update the new data to the storage, and then trigger the build module and LDA block to obtain the new scorecard result. Therefore, the list of recommended games on our web interface is updated daily.

6.6. Bottlenecks and improvements

Although we planned the whole system rooted on cloud storage, we failed to link the url of Google Cloud Storage cannot to JavaScript. As a result, our current web page data can only be displayed in the form of local upload. So in the future, we can look for new cloud storage platforms to load our data and systems.

7. Game Scores and Tags Analysis with Latent Dirichlet Allocation Model

After we finished calculating scores for each game, we further investigated the relationship between game scores and game tags’ decomposition (tags’ decomposition tells us what distributions of tags could describe a game; for example, game CSGO (Counter Strike Global Offensive) may be described by 30% of tag - FPS, 30 % of tag - Multiplayer, 20% of the tag - Shooter, and 20% of tag - Action; the percentages are calculated by the number of times a tag appeared in the game reviews).

Since we have more than 400 game tags, we utilized the Latent Dirichlet Allocation Algorithm, an unsupervised topic modeling algorithm to transform the 400 game tags (recorded as number of times one tag appears in one game’s reviews) into 20 topics.

7.1. Overview of LDA

The latent dirichlet allocation algorithm - LDA - is an unsupervised topic modeling algorithm. It is always used in newspaper article classification tasks. To put it simply, the algorithm basically imagines a fixed set of news topics in the first place. Each topic represents a set of words. And the goal of the algorithm is to map all the articles to the topics in a way, such that the words in each newspaper article are mostly captured by those imaginary topics.

Thus, in our project, we just treat each game’s tag description as “newspaper articles" (for example if game CSGO has 1688 “first-person” tag and 1585 “Tactical” tag, the “article” for CSGO is just: “First-Person, First-Person, … 1688 times of First-Person; Tactical, Tactical, Tactical, … 1585 times of Tactical".). And then, we treat the 400+ unique tags as “words” in the “articles”. In this way, we were able to utilize the LDA algorithm to classify a game into topics, and then analyze a game based on its topic distribution, as well as the tag distribution within each topic.

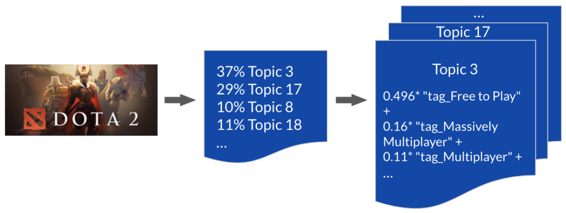

DOTA 2
37% Topic 3
29% Topic 17
10% Topic 8
11% Topic 18
...
...
Topic 17
Topic 3
0.496* "tag_Free to Play"
+
0.16* "tag_Massively Multiplayer" +
0.11* "tag_Multiplayer" +
...

Figure 15: DOTA2 Topic/Tag Distribution

For example, DOTA2 can be explained by 37% of Topic 3, 29% of Topic 7, and etc. The topics can be further decomposed into distribution of tags, the two most important tags that contribute to topic 3 are: ‘free to play’, ‘massive multiplayer’, and the weight for tag ‘free to play’ on topic 3 is 0.49, the weight for tag “Massively Multiplayer” on topic 3 is 0.16.

7.2. LDA Result Visualization

In order to visualize the result of the LDA model better, we utilized the Gensim package to train the model and plot the classification result.

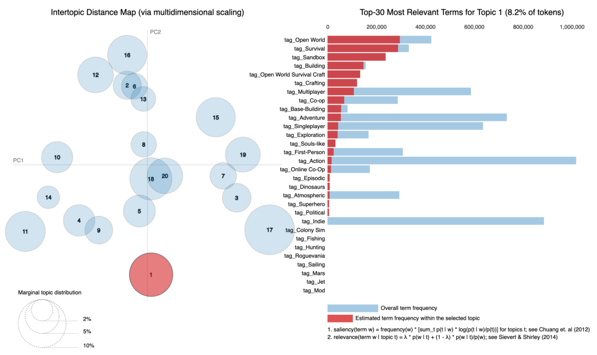


Figure 16: LDA Topics Visualization

The gensim package allows us to interact with the topic modeling result. Each circle on the left hand side of the graph represents a topic‘s position by taking the first two principal components as x and y axis after a PCA dimension reduction. The relative position of the topics can show us the distance between different topics, for example, the topic 18 and 20 should be very similar in terms of their tag’s decomposition, while the topic 1 should be quite different from all the other topics. On the right hand side is the tag decomposition for topics, when we highlight the topic circle using cursor, the tag’s decomposition for that topic will be shown on the right, ordered from the most important to the least important tag.

7.3. Topics With Top Three Key Tags (and Weights)

20 topics and the weights (the weights reflect how important a keyword is to that topic) for the top three key tags for each topic are listed below (by implementing the LDA algorithm on the Steam data we collected before):

- Topic 1: 0.172*"tag_Open World" + 0.167*"tag_Survival" + 0.138*"tag_Sandbox"
- Topic 2: 0.220*"tag_Pixel Graphics" + 0.154*"tag_2D" + 0.093*"tag_Rogue-like"
- Topic 3: 0.497*"tag_Free to Play" + 0.164*"tag_Massively Multiplayer" + 0.107*"tag_Multiplayer"
- Topic 4: 0.253*"tag_Horror" + 0.110*"tag_Survival Horror" + 0.096*"tag_Psychological Horror"
- Topic 5: 0.151*"tag_Realistic" + 0.093*"tag_Racing" + 0.076*"tag_Sports"
- Topic 6: 0.137*"tag_Point & Click" + 0.074*"tag_Mystery" + 0.056*"tag_Hidden Object"
- Topic 7: 0.224*"tag_Third Person" + 0.160*"tag_Post-apocalyptic" + 0.155*"tag_Action"
- Topic 8: 0.262*"tag_Sci-fi" + 0.130*"tag_Space" + 0.113*"tag_3D"
- Topic 9: 0.132*"tag_Classic" + 0.120*"tag_Stealth" + 0.104*"tag_Mature"
- Topic 10: 0.157*"tag_Anime" + 0.099*"tag_Visual Novel" + 0.090*"tag_Choices Matter"
- Topic 11: 0.466*"tag_Adventure" + 0.129*"tag_Story Rich" + 0.122*"tag_Atmospheric"
- Topic 12: 0.146*"tag_Puzzle" + 0.096*"tag_Casual" + 0.082*"tag_Family”
- Topic 13: 0.691*"tag_Simulation" + 0.270*"tag_Early Access" + 0.039*"tag_Steampunk"
- Topic 14: 0.478*"tag_Difficult" + 0.191*"tag_Great Soundtrack" + 0.116*"tag_Replay Value"
- Topic 15: 0.111*"tag_Action" + 0.096*"tag_Platformer" + 0.070*"tag_Arcade"
- Topic 16: 0.633*"tag_Indie" + 0.270*"tag_Casual" + 0.042*"tag_VR"

- Topic 17: 0.169*"tag_FPS" + 0.144*"tag_Multiplayer" + 0.118*"tag_Shooter"
- Topic 18: 0.287*"tag_Strategy" + 0.071*"tag_War" + 0.058*"tag_Military"
- Topic 19: 0.418*"tag_Action" + 0.133*"tag_Funny" + 0.123*"tag_Zombies"
- Topic 20: 0.380*"tag_RPG" + 0.145*"tag_Fantasy" + 0.110*"tag_Action RPG"

From the result, we can see that the LDA model did successfully classify some typical types of Steam game. For example, if we look at Topic 4: 0.253*"tag_Horror" + 0.110*"tag_Survival Horror" + 0.096*"tag_Psychological Horror", this is a very typical horror game description; if we look at Topic 11: 0.466*"tag_Adventure" + 0.129*"tag_Story Rich" + 0.122*"tag_Atmospheric", is again a very typical description for an adventure game. If we go through the topic list one by one, we will come up with different games that match up with each topic quite easily.

7.4. Top 50 Rated Games vs Their Topic Decomposition

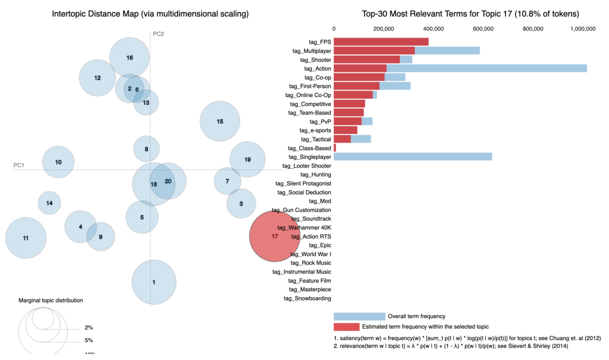


Figure 17: Topic 17 Tag Distribution

For the top 50 rated games from our previous modeling result, more than 20% of games (11 out of 50) has Topic 17 as their major component (major component means that this topic has the highest percentage in the topic distribution for the game). As we can see from the graph, tag "FPS" has the highest weight for Topic 17, followed by tag "Multiplayer", tag "Shooter", tag "Action", tag "Co-op" and etc. One typical game for this topic is Counter Strike Global Offensive.

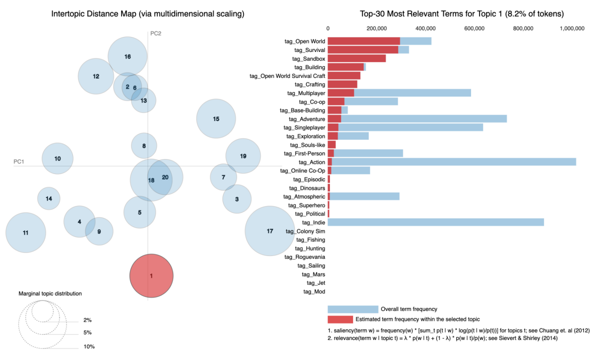


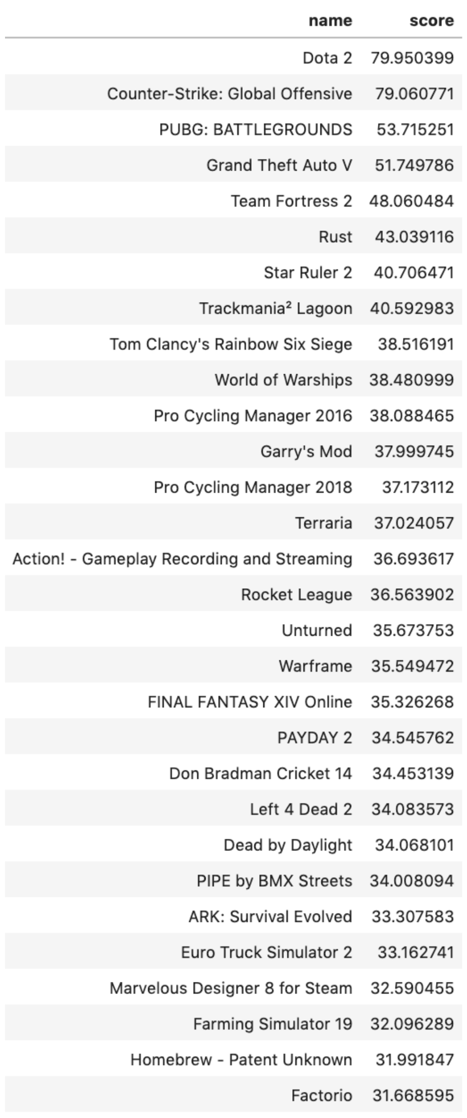
Figure 18: Topic 1 Tag Distribution

And another 20% of games have topic 1 as their major component. As we can see from the graph, tag "Open World" has the highest weight for topic 1, followed by tag "Survival", tag "Sandbox", tag "Building", …, etc. And one typical game for this topic is Grand Theft Auto V.

7.5. Business Value

Based on our analysis, we can suggest that game companies could consider the tags and tag distributions for Topic 17: 0.169*"tag_FPS" + 0.144*"tag_Multiplayer" + 0.118*"tag_Shooter" + 0.094*"tag_Action" + 0.090*"tag_Co-op" + 0.081*"tag_First-Person" + 0.069*"tag_Online Co-Op" + 0.056*"tag_Competitive" + 0.053*"tag_Team-Based" + 0.050*"tag_PvP" and Topic 1: 0.172*"tag_Open World" + 0.167*"tag_Survival" + 0.138*"tag_Sandbox" + 0.085*"tag_Building" + 0.078*"tag_Open World Survival Craft" + 0.070*"tag_Crafting" + 0.063*"tag_Multiplayer" + 0.040*"tag_Co-op" + 0.032*"tag_Base-Building" + 0.032*"tag_Adventure" when designing new games in order to produce high quality and successful games in the future (games like Counter Strike Global Offensive, and Grand Theft Auto V, even better if they can combine these two types). And our analysis provides a more detailed suggestion for game design than simply suggesting a general trend in the game industry.

7.6. Recommended Games List (updated)



name	score
Dota 2	79.950399
Counter-Strike: Global Offensive	79.060771
PUBG: BATTLEGROUNDS	53.715251
Grand Theft Auto V	51.749786
Team Fortress 2	48.060484
Rust	43.039116
Star Ruler 2	40.706471
Trackmania² Lagoon	40.592983
Tom Clancy's Rainbow Six Siege	38.516191
World of Warships	38.480999
Pro Cycling Manager 2016	38.088465
Garry's Mod	37.999745
Pro Cycling Manager 2018	37.173112
Terraria	37.024057
Action! - Gameplay Recording and Streaming	36.693617
Rocket League	36.563902
Unturned	35.673753
Warframe	35.549472
FINAL FANTASY XIV Online	35.326268
PAYDAY 2	34.545762
Don Bradman Cricket 14	34.453139
Left 4 Dead 2	34.083573
Dead by Daylight	34.068101
PIPE by BMX Streets	34.008094
ARK: Survival Evolved	33.307583
Euro Truck Simulator 2	33.162741
Marvelous Designer 8 for Steam	32.590455
Farming Simulator 19	32.096289
Homebrew - Patent Unknown	31.991847
Factorio	31.668595

Figure 19: Updated Game Scorecard

After adding the topic feature (from the LDA model) into the original model, the game lists are shown above. The overall ranking does not differ too much when compared to the previous result.

8. Conclusion

8.1. Process Recall

In conclusion, we started our project because of the ineffectiveness of the existing game recommendation system. We completed our goal, which is constructing a new system of games recommendation system that is influenced by many features from Steamspy data.

In the model part, we constructed four submodels: popularity, penetration, playability, and feedback; and combined these four sub-scores to get our final game score. We let our system update everyday at 2am using Airflow, and we also built a web interface for better visualization of the result.

At the last part of our project, we also investigated the relationship between game scores and game tag decompositions. We looked at the 50 top ranked games and interpreted the main tag decompositions for these games using the LDA model. This kind of interpretation could help us provide a more detailed game design suggestion for the game companies.

8.2. Drawbacks and Future Improvements

The cloud storage we tried to use is Google Cloud Storage. However, the provided url cannot be used for JavaScript data input. Thus, for future work, a new cloud platform should be found for our website and data storage.

As for the web interface, the version we now use is only a demo built based on JavaScript. More optimization improvements can be made with CSS.

References

[1] FirstName Alpher, , and J. P. N. Fotheringham-Smythe. Frobnication revisited. Journal of Foo, 13(1):234–778, 2003.

[2] FirstName Alpher, , FirstName Fotheringham-Smythe, and FirstName Gamow. Can a machine frobnicate? Journal of Foo, 14(1):234–778, 2004.

[3] FirstName Alpher. Frobnication. Journal of Foo, 12(1):234–778, 2002.

[4] Actual Author Name. The frobnicatable foo filter, 2014. Face and Gesture (to appear ID 324).

[5] Actual Author Name. Frobnication tutorial, 2014. Some URL al tr.pdf.

[6] "6 Ways to Predict the Sales Success of a Game - Expert Blog for Professionals in the Video Game Industry." Expert Blog for Professionals in the Video Game Industry, 11 Apr. 2016, https://codeswholesale.com/blog/6-ways-to-predict-the-sales-success-of-a-game/.

[7] Budiarto, Joseph Alexander. "Game Popularity Tracking System." International Journal of Industrial Research and Applied Engineering, no. 2, Petra Christian University, Oct. 2018. Crossref, doi:10.9744/jirae.3.2.79-85.

[8] Davis, Nik. "Steam Data Exploration in Python | Nik Davis." Home | Nik Davis, https://nik-davis.github.io/posts/2019/steam-data-exploration/.

[9] jruots. "Forecasting Video Game Sales | Kaggle." Kaggle: Your Machine Learning and Data Science Community, Kaggle, 9 Dec. 2017, https://www.kaggle.com/jruots/forecasting-video-game-sales.

[10] Kang, Ha-Na. "A Study of Analyzing on Online Game Reviews Using a Data Mining Approach: STEAM Community Data." International Journal of Innovation, Management and Technology, EJournal Publishing, 2017, pp. 90–94. Crossref, doi:10.18178/ijimt.2017.8.2.709.

[11] Muller, Andrew. "Video Game Review Analysis. NLP Classification with and Neural… | by Andrew Muller | Medium." Medium, Medium, 11 Apr. 2021, https://andrew-muller.medium.com/video-game-review-analysis-3c7602184668.

[12] "Steam Community :: Steam Web API Documentation." Steam Community, https://steamcommunity.com/dev.

[13] "Twitter API Documentation | Docs | Twitter Developer Platform." Use Cases, Tutorials, & Documentation | Twitter Developer Platform, https://developer.twitter.com/en/docs/twitter-api.

[14] Weber, Ben G. "Recommendation Systems in Games." Game AI Pro 3, A K Peters/CRC Press, 2017,pp.461–70,http://dx.doi.org/10.4324/9781315151700-39.

[15] "What Makes a Game Good?" The Games Journal, http://www.thegamesjournal.com/articles/WhatMakesaGame.shtml.

[16] "Gensim: topic modeling for humans", https://radimrehurek.com/gensim/models/ldamodel.html

[17] Prabhakaran, Selva. "Topic Modeling with Gensim (Python)", https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/#13viewthetopicsinldamodel

Individual Student Contributions in Fractions

	xh2469	tj2441	xh2509
Last Name	He	Jiang	He
Fraction of (useful) total contribution	1/3	1/3	1/3
What I did 1	Data Streaming	Literature Review	Data Wrangling
What I did 2	LDA Analysis	Model Construction	WebPage Design
What I did 3	Report Writing	Report Writing	Report Writing