Sentiment Classification of Game Reviews on Steam with Semi-supervised Topic Analysis

Course project for CSE 6240: Web Search and Text Mining, Spring 2020

Jingyu Li
School of Industrial and System Engineering
alanli@gatech.edu

Yanxin Ye
School of Industrial and System Engineering
yye79@gatech.edu

Shang-Han Yang
School of Computational Science and Engineering
syang483@gatech.edu

ABSTRACT

Understanding sentiment of game reviews is important for game developers as well as investment companies. However, gaming is a domain in which in-group users have mutual but specific words and expressions which makes text mining harder. Our project focused on sentiment classification problems on game reviews from Steam. Beyond the generic word2vec embeddings, we applied the semi-supervised topic analysis method guided LDA, which can incorporate domain knowledge to generate probability distribution of a review belonging to different topics. We found that performance of Random Forest and Logistic Regression improved when using LDA features or LDA features and word2vec embeddings together, compared with using word2vec only. We also experimented with the popular Gradient Boosting Machine for sentiment classification and applied the Bayesian Optimization method to speed up hyperparameter tuning process. The F1 score of our final model is 0.8997.

ACM Reference Format:
Jingyu Li, Yanxin Ye, and Shang-Han Yang. 2020. Sentiment Classification of Game Reviews on Steam with Semi-supervised Topic Analysis Course project for CSE 6240: Web Search and Text Mining, Spring 2020 . In *Proceedings of* . ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

Gaming industry plays an important role in the global economy and our everyday life. In the annual report released by Newzoo.com, over 2.5 billion gamers contributed 150 billion U.S. dollars of revenue in 2019. Steam, which has over 14 million users active online per day, is the largest platform for selling PC games, as well as the largest community where users can discuss their opinions towards different games. Our project focused on the sentiment classification task upon users reviews on PC games. We applied topic analysis to improve classification performance beyond traditional word2vec embeddings.

The dataset we used is Steam game reviews from Kaggle [9]. We applied a semi-supervised topic analysis approach called guided LDA to generate new numeric features, which is the probability distribution of a review belonging to different topics, upon baseline word2vec embeddings to represent review text. Compared with traditional topic analysis method LDA, we provided several seed words for guided LDA which guide the topics extracted getting closer to the high level natural concept. In our experiment, using LDA features solely outperformed the word2vec embeddings in F1 score, which is stable on different models (e.g. 0.8853 vs. 0.8765 for Random Forest). When using LDA features and word2vec embeddings together, model performance increases more. Compared with baseline, F1, Recall and Precision all went up when using Random Forest or Logistic Regression. Besides exploring topic analysis, we also experimented on Gradient Boosting Machine for classification and Bayesian Optimization method for hyperparameter tuning. We got a better classification model with word2vec+LDA features, whose F1 score on an independent test set is 0.8997, higher than baselines and other models we experimented with.

Our project leveraged semi-supervised topic analysis into sentiment classification on game review. The topic related features improved the classifier's performance. Considering text analysis has the property of domain specificity, the first impact of our project is that we illustrate a way to involve domain knowledge to generate word embeddings which can improve models' performance in a certain domain. Secondly, for many game developers, they see thousands of reviews towards their products, many of which don't have sentiment labels. The model we trained can be used for future sentiment prediction. It can help the game developers better understand users' attitude and preference on their games, and also to help investment companies evaluate the product performance of certain target game developers.

2 LITERATURE SURVEY

Sentiment analysis is a main topic in natural language processing and opinion mining. One basic task is sentiment classification: whether the sentiment to a target is positive or negative. Although human language shares common features broadly, we argued that people may express their attitude and emotion in different ways across different domains. But domain specific features are rarely considered in previous sentiment classification research. However, gaming is a domain in which in-group users have mutual but specific words and expressions. The sentiment analysis in game reviews is challenging and not many researches appeared. The only two working papers we found explored the problem simply, in which the author used the bag-of-word method (e.g. tf-idf) to generate features [1][14]. We proposed that sentiment classification in game reviews need more investigation, especially focusing on domain specific properties. One potential perspective is incorporating topic analysis for sentiment classification. Because users usually evaluate games from different aspects and some domain specific latent topics may exist in game reviews (e.g. game design, hardware support, emotional reaction etc.), which can improve sentiment prediction.

Topic analysis aims at solving the problem of what common topics or aspects are referred to in the review corpus. Schouten and Frasincar proposed that the methods used can be divided into frequency-based, syntax-based, supervised machine learning, unsupervised machine learning, and hybrid approaches [13]. Since topic labels are hard to obtain in practice, an unsupervised method called Latent Dirichlet Allocation (LDA) is widely used, which is introduced by Blei et al [2]. LDA is a bag-of-word model. The basic assumption of LDA is that a hidden structure which consists of a set of topics exists in the whole textual dataset. Upon this assumption, it regards each document as a multinomial distribution of a set of topics and each topic as a multinomial distribution of a set of words. By using LDA, three outputs are generated: 1) a set of topics will be extracted from the whole text; 2) for each review, a vector of weight values will be generated which represent the probability of that review containing the corresponding specific topics; 3) for each topic, a set of words which associated to the topic with weights will be generated. In previous research, LDA is used for topic modeling in restaurant reviews [4], doctor reviews [7] and hotel reviews [12].

The shortcoming of LDA is that as an unsupervised approach, the results it generates can be various depending on the corpus we have and sometimes the results are hard to interpret. Thus, a semi-supervised method based on LDA was proposed, called guided LDA [5]. The initial probability of a certain word belonging to different topics is nearly uniformly distributed in LDA; however, by setting some seed words and assigning them to different groups based on our domain knowledge, some extra probability boost is given to each seed word to lie in a specific topic. This is the core improvement in guided LDA, incorporating lexical priors into topic models. We thought guided LDA is a proper and better approach for topic analysis on game reviews, since we can introduce domain knowledge into the model and improve predicting power.

To summarize, our project focused on sentiment classification on game reviews. Besides using word2vec to generate word embeddings as our baselines, we applied a semi-supervised topic analysis method, guided LDA, to calculate the probability distributions of each review belonging to different topics. We explored whether the new features could improve the model performance.

3 DATASET DESCRIPTION AND ANALYSIS

3.1 Dataset Introduction and Data Preparation

The dataset we use in our project is from Kaggle [9], which contains reviews from Steam's best-selling games as of February 2019. The number of reviews in the dataset is 434,891, commenting on 48 popular games and being written by users from December 2010 to February 2019. The main target columns in the dataset that are needed for our project are "review" and "recommendation". The column "review" is the raw text of user reviews, while the column "recommendation" records whether the user is willing to recommend the game to others or not. We propose that the column "recommendation" can be used as the sentiment label in which "recommended" is regarded as positive sentiment and "not recommended" is regarded as negative sentiment.

In our data preprocessing, we first clean each review into lowercase characters only, and then generate token sentences as the input of word2vec training. After word2vec finishes training, we cluster the output vectors by an appropriate number of clusters. The number of clustering is determined by the balance between higher similarity of words within clusters and acceptable minimum words count within clusters (see section 3.3). Then, we clean the review again into only lowercase characters and without stopwords. Finally, we count the frequency of words within each cluster as embeddings for a review and normalize the embedded vectors because we don't want a longer review to have larger values in all the feature spaces while its sentiment polarity might not be stronger than a very short review with obvious sentiment polarity such as "Good!".

Table 1: Basic Raw Data Statistics

Property	Value
Sample size	433375
Ratio of positive sentiment label	0.698
Raw vocabulary size	170451
Final token size (min_word_cnt=40)	9650
Average number of sentences per review	2.6
Average number of tokens per sentences	15.8

Since the first goal of our project is to do sentiment classification of game reviews, we believe this dataset is sufficient to achieve our goals. First of all, the dataset provides an adequate number of reviews and corresponding user sentiment, which meet the need for sentiment analysis. Secondly, the target games of these reviews are all popular games in recent years, like Playerunknown's Battleground, Dead by Daylight and Grand Theft Auto V, which makes the dataset a good representative sample in the game domain. Besides, the target games include different game genres so that the diversity of review corpus increases.

3.2 Raw Data Statistics

After preprocessing the data, we explore some basic statistics of our dataset, shown in Table 1.

3.3 Data Analysis

Beyond the raw data statistics, we explored the data to help our modeling. First of all, the sentiment label is imbalanced. The proportion of positive labels is around 70%, which causes accuracy invalid in model evaluation. We used F1 score instead as the main metrics to evaluate models.

Second, to transform the text to numeric features, we clustered the vectors trained by word2vec. We monitored the total similarity of words within a cluster by the averaged distance for a word vector to its centroid of the belonging cluster. Then, we averaged all the similarities in all the clusters to find a balanced cluster number such that the words within clusters are more similar and the minimum number of words within a cluster is not too small. Our final decision is using K=80.

Third, after we embedded the word count in each review with the clustered mapping and normalized the embedded review vector, we calculated the averaged value of each feature for sentiment label y=1 and y=0 respectively, and then calculated the absolute

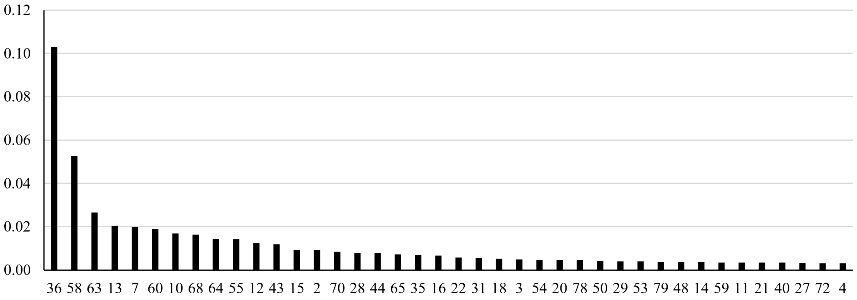

Figure 1: Estimated Top 40 Feature Importance vs ClusterID (featureID)

Table 2: Evaluation of the Number of Clusters

Number of Clusters	10	20	40	80	100
Average distance to cluster center	0.87	0.84	0.81	0.78	0.77
Average count of words in a cluster	965	483	241	121	97
Minimum count of words in a cluster	543	236	78	48	44

Table 3: Sample words within top 5 features

Top 5 features	Sample words
Cluster 36	’fun’, ’good’, ’great’, ’well’, ’nice’
Cluster 58	’but’, ’not’, ’so’, ’as’, ’there’
Cluster 63	’product’, ’devs’, ’modding’, ’developers’, ’support’
Cluster 13	’waste’, ’trash’, ’stupid’, ’hell’, ’garbage’
Cluster 7	’bad’, ’toxic’, ’terrible’, ’broken’, ’buggy’

difference between the two average values for each feature to represent whether this feature is valid to distinguish the sentiment polarity. We sorted these absolute average value’s differences for all the features in descending order to get an impression of relative feature importance.

Fourth, we further investigated the sample words in the clusters with higher importance. We found that these words somehow meet our expectations for sentiment polarity, especially cluster 36, 13, and 7. We also thought the cluster number selection mechanism we discuss above helps us obtain such intuitive features. (See Table 3)

Fifth, from the sample words of each cluster, we also observed some hints that the proposed semi-supervised topic analysis can be helpful. For example, “potato” is a meme in the game domain which means the game server is not stable. But it wasn’t assigned to the same cluster with the more explicit words like “server”, “internet” and “ping”. In our proposed method, we tried to incorporate such domain knowledge to better generate features.

4 EXPERIMENT SETTING AND BASELINES

4.1 Data Split, Model Evaluation and Metrics

We split the whole dataset as the training set (80% of the whole dataset) and the test set (remaining 20%). The training set is for model training and validation purposes. We used 5-fold cross validation on the training set to train each model, perform hyperparameter tuning and compared different models so that we can select the model producing the best predictions. After we chose the best model, we used the whole training set to retrain the model with the selected hyperparameters and then used the test set to evaluate the performance of the selected model.

The evaluation metrics we used are F1 score, Recall, and Precision. And F1 score was the main indicator for our project to compare, select and evaluate models.

4.2 Baseline and System Setting

As discussed above, we used word2vec embedding as the baseline features and 80 features (clusters) were included. We followed previous research to include Gaussian Naïve Bayes, Decision Tree, Logistic Regression and Random Forest as baseline classification models [14]. The reason why these baselines are suitable for our problem is that we planned to do binary sentiment classification upon the game reviews and all these models are typical classification approaches. In the meanwhile, our project intended to improve the sentiment prediction by leveraging topic analysis on the text. Thus, in our baseline models we only used word2vec to train the corpus and generated embeddings for a review as the features so that the model performance can be the baseline to which we can compare when topic related features were added. Table 4 illustrates our parameter settings on baseline models. We didn’t perform any hyperparameter tuning on baseline models.

We worked on Python 3.6.5 environment, and the RAM of the laptop is 16.0 GB. The code repository we used for our baselines is scikit-learn. And we set random state as 0 for all models, which we didn’t write out explicitly in parameter setting columns of Table 4.

5 PROPOSED METHOD

5.1 Guided LDA

As mentioned in literature review, we used guided LDA approach to extract topics from the corpus and the probability distribution of a review belonging to different topics form a set of new numeric features to represent the text. By leveraging this method, we can

Table 4: Model Settings and Model Performance

Experiment	Model	Feature	Module	Paramter	F1	Recall	Precision
Baseline	Gaussian Naïve Bayes	word2vec	sklearn		0.7527	0.6542	0.8862
	Decision Tree	word2vec	sklearn		0.8333	0.8466	0.8203
	Logistic Regression	word2vec	sklearn	max_iter=10000	0.8615	0.9257	0.8057
	Random Forest	word2vec	sklearn	n_estimators=500	0.8765	0.9131	0.8427
Proposed	Logistic Regression	LDA	sklearn	max_iter=10000	0.8622	0.8934	0.8331
	Random Forest	LDA	sklearn	n_estimators=500	0.8853	0.9084	0.8633
	Gradient Boosting	LDA	lightGBM	n_estimators=500	0.8853	0.9053	0.8662
	Logistic Regression	w2v+LDA	sklearn	max_iter=10000	0.8746	0.9041	0.847
	Random Forest	w2v+LDA	sklearn	n_estimators=500	0.892	0.9189	0.8666
	Gradient Boosting	w2v+LDA	lightGBM	n_estimators=500	0.8937	0.9152	0.8733

incorporate game-specific domain knowledge in generating numeric representations of text. We proposed that these features can be more efficient in predicting sentiment than generic word2vec embeddings in baseline.

We used python module guidedLDA [10] to implement the semi-supervised topic analysis. The seed words we set are shown in Table 5. The topics we assumed have two categories: one is the generic aspect that influences player experience (except topic 8 and 9); and another one is players' emotional reactions to the game. Our first step was generating bag-of-word embeddings for each cleaned review based on word count. We selected the top 7000 words by term frequency. We assigned 0 to 6999 as the id for each word. The input of guidedLDA model is bag-of-word embeddings and a dictionary of topic id to seed words' id. Although we only assumed 11 topics in our seed, we set the number of topics to be generated as 15 in the model so that some latent topics may also be extracted. The model outputted the probability distribution of each review belonging to one of the 15 topics, and we used this 15-dimension output as features.

Besides, we defined another set of features which we called them "weighted special word probability". Since the guided LDA model can also generate the probability for a word of belonging to a certain topic. We explored the top words in each topic in terms of probability. We got an interesting finding that the top words in each topic can be divided into two groups. One group we called "pan words". A typical example is "game". Since this word appears simultaneously with different words from different latent topics, it's in the top list of almost all the topics. Another group we called "special words", which only appears in one or two topics' top list and is in accordance with the topic closely, like "potato" in topic 2 and "amd" in topic 4. So we selected the top 500 words in each topic and identified the "special words", which appeared no more than twice. We retained the probability of "special words" then calculated the "weighted special word probability" as the following equation. In this way, we generated another 15 features.

Weighted special word probability for text i in topic j:

$$\frac{\sum_{k=1} p_k count_k}{\sum_{k=1} count_k}.$$

5.2 Gradient Boosting Machine

In baselines, Random Forest performed the best. Beside the bagging model, we also proposed to use another category of ensemble learning: boosting model. Boosting is an ensemble learning approach that a sequence of models are learned aiming at decreasing the bias of previous weak models. Gradient Boosting Machine (GBM) is one of the most popular boosting models. It adopted the concept of gradient descent, which is used to update the parameters to minimize loss function of a model. In Gradient Boosting, the whole fitting function is regarded as the parameter of the loss function and we calculate the steepest negative descent by using gradient descent method. The descent is often called pseudo residual. Then a base learner function is fitted with the pseudo residual. In each iteration, this base learner function will be added to the former weak models to get a new model. Many research and practical projects in different domains showed that GBM outperformed in machine learning problems including classification. So in our project, we also planned to experiment on GBM to investigate its performance and build a better model. We use LightGBM module, which is an efficient and scalable gradient boosting tree framework created by Microsoft [8].

5.3 Bayesian Optimization

Another problem in text mining is that the feature space of word embeddings is generally in large scale, which makes the training process long. As a result, the traditional grid search method in hyperparameter tuning becomes time consuming. In order to tackle this problem in our project, we adopted the Bayesian Optimization method for hyperparameter tuning. Bayesian Optimization is a optimization method designed for objective functions that take a long time to evaluate and is widely used for black-box derivative-free global optimization [3]. The objective in our project is maximizing the F1 score of a classifier and the parameters are the model hyperparameters. In each iteration, it first models the objective function using Gaussian Process to provide posterior probability distributions of the objective function for any given set of parameters. The algorithm then uses an acquisition function to decide which hyperparameter combination to sample next. The acquisition function is a function of distribution properties (e.g. mean,variance) of the updated posterior distribution we get in the former step. The design of acquisition is a balance between exploration (mean)

Table 5: Seed Words for Topics

ID	Seed words	Assumed topic
0	gameplay, mechanics, combat, fps, survive, shooting, online, single, multiplayer	gameplay
1	money, free, price, pay, dlc, skins	price
2	server, fix, bugs, lag, potato, connection	server
3	cheat, hackers, aimbot	cheat
4	cpu, gpu, laptop, ram, hardware, crash	hardware
5	friends, teammates	cooperation
6	story, experience, sound, physics, music	art design
7	naked, nudity, blood, racist, idiots, noobs	offensive
8	happy, recommend, favorite, great, nice, amazing, awesome, perfect, simple, fantastic	praise
9	sick, tired, disappointed, worst, trash, stupid, hell, garbage	criticise
10	alpha, early, new, future, patch	new game

Table 6: Bayesian Optimization on GBM

Parameters	Sample range	Optimization results
n_estimators	(500, 10000)	9924
learning_rate	(0.001, 0.1)	0.03616
num_leaves	(20, 70)	69
min_data_in_leaf	(100, 1000)	110
max_depth	(3, 12)	8
reg_alpha	(0, 3)	1.631
reg_lambda	(0, 3)	0.05826
min_split_gain	(0.001, 0.1)	0.06095
min_child_weight	(5, 50)	8.65
colsample_bytree	(0.1, 0.9)	0.4008
subsample	(0.5, 1)	0.8351

and exploitation (variance). Large mean means the value is more likely to be the true value of objective function, while large variance means we are more likely to find a larger point on objective function. So by maximizing the acquisition function, we find the ideal hyperparameter sample point for next round. We used a popular acquisition function called Expected Improvement in our project. The final hyperparameters we used were the one with the largest F1 score in all iterations.

We used Bayesian Optimization on tuning hyperparameters of the GBM model. The module we use is bayesian-optimization [11]. The hyperparameters we tuned and the corresponding sample range is shown in Table 6. We set to get 5 initial samples and then do 15 rounds of optimization.

6 EXPERIMENT

6.1 Experiment Result

We used 5-fold cross validation on the training set to compare the performance of all baseline models and proposed models. All results of baseline models and proposed methods are illustrated in Table 4.

The baselines are several classification models with word2vec embeddings. Table 4 shows that Random Forest and Logistic Regression performed significantly better than Naive Bayes and Decision Tree. The F1 score of random forest baseline and logistic regression baseline have already exceeded 0.85.

The first target of our proposed method is to investigate whether the features generated by guided LDA can improve the model prediction. We followed the process described in section 5.1 to generate 30 topic features. We firstly used Random Forest and Logistic Regression only on the LDA features. They all outperformed the corresponding baseline in terms of F1 score. Further comparison revealed that the improvement in F1 score mainly resulted from large increases in Precision with small decreases in Recall, indicating that the model using LDA features has a lower error rate when predicting positive.

Then we used both word2vec embedding and LDA features as predictors and ran Random Forest and Logistic Regression. Compared with baselines and models only using LDA features, the F1 scores of both models are higher. And Recall and Precision both increased upon baseline performance. So combining the model performance with word2vec, LDA and word2vec+LDA, we can conclude that the semi-supervised LDA features we generate can improve the sentiment classification on game review and it mainly raises Precision score. Then, using these features along with word2vec can outperform more on both Recall and Precision.

The second target of our proposed method is to explore GBM's performance. Results show that its F1 score is a bit higher than that of Random Forest, due to better performance on Precision. In our project, GBM didn't show a significant advantage which is widely proved in other research. One possible reason is that some research showed that Gradient Boosting performed relatively bad in high dimensional and sparse space [6].

The third target of our proposed method is experimenting with the Bayesian Optimization method in hyperparameter tuning. We applied GBM on word2vec+LDA features. The F1 scores of GBM models with each parameter sample is shown Figure 2. The best F1 score we achieved was 0.8968, which outperformed all other models in Table 4. The corresponding parameter set is shown Table 6. We got increases in hyperparameter tuning, but the degree is not significant. There can be three reasons which need more investigation in the future. First of all, we could sample more times so that we are more likely to achieve the global optimum. Second, we may need to adjust the sample range of each parameter. Third, it's possible that we have already met the ceiling of predicting power by using these features and the GBM method. Other embedding methods or modeling approaches are needed.

As a final evaluation, we used the whole training set to train this selected model again and then evaluated it on the test set. The F1, Recall and Precision on test set was 0.8997, 0.9211 and 0.8793, which proved this model is a good classifier in game review sentiment classification.

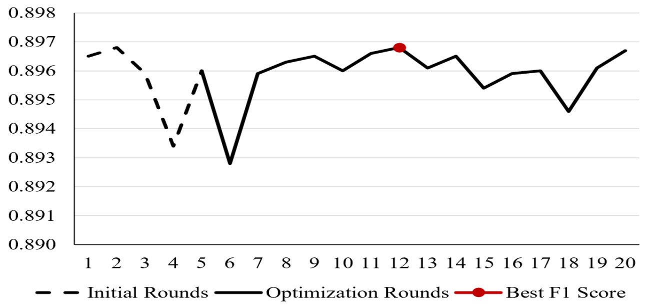

Figure 2: F1 Score of Sampled Models in Bayesian Optimization

6.2 Discussion: Why LDA works

We proposed a way to explain why topic features can improve the performance. In word2vec we used in baseline models, the vector of a center word was trained by the neighbor word pairs within a certain window. This training mechanism makes the created word vector incorporate the local semantic relations, and usually results in generated similar vectors with similar part-of-speech tagging. Thus, words with opposite polarity in sentiment can be clustered into the same group. However, compared to word2vec which incorporates more local information, the LDA algorithm incorporates more global statistical information because it assigns an undecided word to a topic whose words more frequently come along with the undecided word within all reviews, regardless of their positions in the reviews. And the pre-assigned seed words boost the prior probability more. From the feature importance of GBM in Figure 3, we found that all topic features were in the top (t0 to t14), which is a further proof of the aforementioned limitation of word2vec and our good choices of pre-assigned words for guided LDA.

7 CONCLUSION

Our project incorporated semi-supervised topic analysis into the sentiment classification on game reviews and proved that the numeric features from topic analysis can improve the prediction performance. The limitation of our work is that since guided LDA is not a fully supervised approach, its results may change along with different corpus samples. So to investigate whether guided LDA is a valid way to create numeric representation of text and whether it can improve sentiment classification stably, more experiments on different dataset and different domains are needed.

A potential extension of our work is that instead of evaluating on the review level, we can dive into sentence level or topic level. In reality, people's sentiment towards a certain target can be mixed. In game reviews, this phenomenon is also significant. Conducting sentiment analysis on topic level can help game developers understand players' reactions in a smaller granularity, providing more business insight.

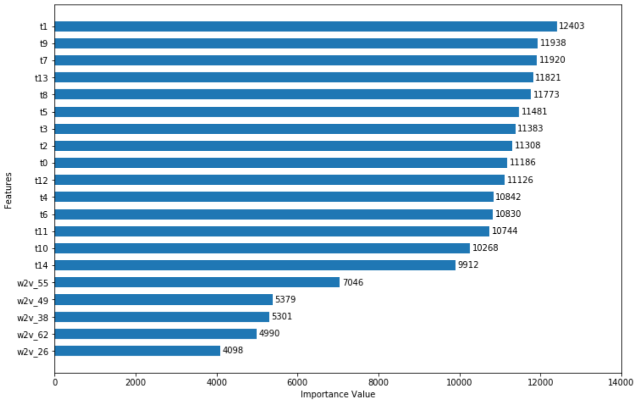

Figure 3: Feature Importance of GBM

8 TEAM CONTRIBUTION

All team members have contributed a similar amount of effort.

REFERENCES

[1] Rohan Bais, Pasal Odek, and Seyla Ou. 2017. Sentiment Classification on Steam Reviews.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[3] Peter I Frazier. 2018. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811* (2018).

[4] James Huang, Stephanie Rogers, and Eunkwang Joo. 2014. Improving restaurants by extracting subtopics from yelp reviews. *iConference 2014 (Social Media Expo)* (2014).

[5] Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. 2012. Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 204–213.

[6] Jiawei Jiang, Bin Cui, Ce Zhang, and Fangcheng Fu. 2018. Dimboost: Boosting gradient boosting decision tree to higher dimensions. In *Proceedings of the 2018 International Conference on Management of Data*. 1363–1376.

[7] K. Kavya and C. Sreejith. 2018. Know Your Doctor Topic Modeling and Sentiment Analysis Based Approach To Review Doctor. *International Journal of Computer Sciences and Engineering* 06 (07 2018), 37–42. https://doi.org/10.26438/ijcse/v6si6.3742

[8] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*. 3146–3154.

[9] Luthfi Mahendra. 2019. Steam Reviews Dataset: Collection of Steam's Best Selling Games Reviews. *https://www.kaggle.com/luthfim/steam-reviews-dataset* (2019).

[10] GuidedLDA module. [n.d.]. *https://github.com/vi3k6i5/GuidedLDA* ([n. d.]).

[11] Bayesian optimization module. [n.d.]. *https://github.com/fmfn/BayesianOptimization* ([n. d.]).

[12] Isidoros Perikos, Argyro Tsirtsi, Konstantinos Kovas, Foteini Grivokostopoulou, Ioannis Daramouskas, and Ioannis Hatzilygeroudis. 2018. Opinion Mining and Visualization of Online Users Reviews: A Case Study in Booking. com. In *2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE, 1–5.

[13] Kim Schouten and Flavius Frasincar. 2015. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering* 28, 3 (2015), 813–830.

[14] Zhen Zuo. 2018. Sentiment analysis of steam review datasets using naive bayes and decision tree classifier. (2018).