SENTIMENT ANALYSIS ON GAME REVIEWS: A COMPARATIVE STUDY OF MACHINE LEARNING APPROACHES

Jie Ying Tan[1*], Andy Sai Kit Chow[1] and Chi Wee Tan[1]

[1] Faculty of Computing and Information Technology, Tunku Abdul Rahman University College, Kampus Utama, Jalan Genting Kelang, 53300, Wilayah Persekutuan Kuala Lumpur, Malaysia
*Corresponding author: tanjy-wp17@student.tarc.edu.my

ABSTRACT

Sentiment analysis is one of the major topics of natural language processing which is used to determine whether data is positive, negative or neutral. It is often performed on textual data to help businesses monitor brand and product sentiment in customer feedback to understand their customers' needs. This paper explores various machine learning algorithms including Logistic Regression (LR), Multinomial Naïve Bayes (MNB), Support Vector Classifier (SVC), Multi-layer Perceptron Classifier (MLP) and Extreme Gradient Boosting Classifier (XGB) to build sentiment analysis models tailored for the gaming domain to classify reviews into positive, negative and neutral. The models were trained on game reviews obtained from Metacritic and Steam. Various data preprocessing and model optimization techniques have been employed and the performance of the models were evaluated and compared. SVC has been determined as the best-performing model among all the models.

Keywords: Sentiment Analysis, Natural Language Processing, Machine Learning, Support Vector Machine, Game Reviews

1.0 INTRODUCTION

The video game industry has gradually grown to become one of the most profitable segments of the entertainment industry. The advancement of technology has spurred the accessibility of video games and popularized it, with various genres available targeting different audiences. In order to even compete in the market, game developers need to have a clear understanding of customers' opinions and how to retain their user base. By understanding the needs and wants of the users, game developers will be able to more effectively design their games according to the users' satisfaction. Therefore, sentiment analysis is necessary to help game developers uncover the true feelings and opinions of users towards their games. In view of the above, the specific objectives of our project are:

I. To train multiple machine learning models to classify sentiment of game reviews and compare their performance.

II. To investigate whether oversampling and hyperparameter tuning improve the models' performance.

2.0 LITERATURE REVIEW

This section describes the supervised machine learning algorithms used in this project and previous related studies.

2.1 Machine Learning Algorithms

Support Vector Machine (SVM) is a statistical classification approach that determines a hyperplane in an N-dimensional space where N being the number of features, that distinctly classifies the data points. It was considered to be the best text classification method (Xia, Rui, Chengqing Zong, and Shoushan Li, 2011). It is a non-probabilistic binary linear classifier with the ability to separate the classes by a large margin linearly, capable of becoming one of the most powerful classifiers proven by its capability to handle infinite dimensional feature vectors (Al Amrani, Lazaar and El Kadiri, 2018). SVC is developed based on SVM and has various applications which include numerical pattern recognition, face detection, text categorization and protein fold recognition (Lau and Wu, 2003).

LR is a machine learning algorithm that is used to solve classification issues based on the concept of probability. There are a few assumptions that must be met for LR which include the dependent variable must be dichotomous, linear relationship between the dependent and independent variable does not exist, the independent variable must be linearly related, neither normally distributed, nor of equal variance within a group that must be mutually exclusive (Prabhat and Khullar, 2017). Examples of application of LR in various fields include the medical field where it can predict the mortality of injured patients (Boyd et al., 1987).

XGB is a variant of the Gradient Boosting Machine proposed by Chen and Guestrin. The selling point of XGB is the unparalleled scalability in all scenarios which consumes far less resources than existing systems. The system runs more than ten times faster than existing popular solutions on a single machine and scales to billions of examples in distributed or memory-limited settings. There are a few factors that contribute to the scalability of XGB which are related to systems and algorithmic optimizations. For example, handling of sparse data is by a novel tree learning algorithm and handling of instance weights is through a theoretically justified weighted quantile sketch procedure. These outstanding features have made XGB a widely recognized system in machine learning and data mining challenges (Chen and Guestrin, 2016).

Naïve Bayes (NB) algorithm is a classification technique based on the Bayes' Theorem assuming that there is independence among predictors. It is mostly used for document level classification. The general idea of the algorithm is that through the joint probabilities of words and categories, the calculation of the probabilities of categories given a test document can be performed. The decision-making time for NB classifiers is computationally short and learning can be started without a large amount of data (Ashari, Paryudi and Min, 2013). There are a few variations of the NB classifier, namely Multinomial Naive Bayes (MNB), Bernoulli Naive Bayes (BNB) and Gaussian Naive Bayes (GNB).

MLP is a type of feed-forward artificial neural network made up of neurons called perceptrons. Neurons are hierarchically arranged in multiple connected layers which are made up of three kinds of layers, namely the input layer, output layer and hidden layer. The input signal is passed through the input layer while the output layer performs prediction and classification with the hidden layer providing computational processing in the network to produce the network outputs. The objective of training MLP networks is to determine the best set of connection weights and biases to minimise the prediction error (Alboaneen, Tianfield and Zhang, 2017).

2.2 Related Work

Previous studies have been done for performing sentiment analysis using machine learning techniques. Chakraborty et al. (2018) performed sentiment analysis on game reviews obtained from Amazon and Twitter. The algorithms which include NB, SVM, LR and Stochastic Gradient Descent (SGD) were used to train sentiment analysis models and the models were evaluated in terms of their accuracies. The feature extraction method used was the Bag-of-Words method.

Zuo (2018) performed sentiment analysis on game reviews collected from Steam. The algorithms used were NB and Decision Tree classifiers. Feature selection using information gain was carried out, followed by feature extraction through Term Frequency-Inverse Document Frequency (TF-IDF) and hyperparameter tuning of the models through grid search.

Britto and Pacifico (2020) conducted a study on video game acceptance by performing sentiment analysis on game reviews. The dataset used was game reviews written in Brazilian Portuguese language extracted from Steam. Feature extraction was performed using the Bag-of-Words method. The algorithms implemented were Random Forest classifier, SVM and LR.

Based on the previous studies, there exists several research gaps for sentiment analysis on game reviews using machine learning techniques including the lack of implementation of XGB and MLP algorithms. Besides, there is a lack of exploration on sentiment analysis for more professional and complex reviews written by game critics such as the reviews on Metacritic. Furthermore, the effect of resampling techniques such as oversampling along with hyperparameter tuning of TF-IDF have not been studied before.

3.0 METHODOLOGY

This section presents the project framework, datasets, text preprocessing, data labelling, feature extraction, handling of imbalanced classes, model applications and hyperparameter tuning.

3.1 Project Framework

The framework of this project is shown in Figure 1.

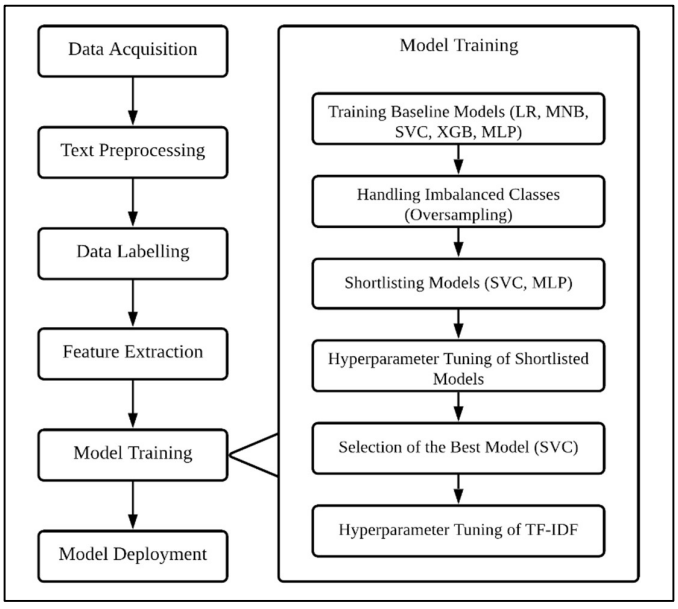

Figure 1. Project Framework.

3.2 Dataset

The data used for training the models contains 17543 game reviews, 8363 of which were critic reviews scraped from Metacritic's website (https://www.metacritic.com/) and 9180 were user reviews collected from Steam (https://store.steampowered.com/) by using its web API in July 2021. Table 1 shows the sample reviews obtained.

Table 1. Sample reviews

	Metacritic critic reviews	Steam user reviews
1	While the AI has a few problems, and there is an occasional rough spot, the finished product is one of the most complete and compelling games we've played, and easily the best MechWarrior game in the series.	EA sucks at their customer service and they manage to crash their game upon every launch, they like grabbing your money. But the game is very fun yes.
2	With plenty thatâ€™s new to see and do, even without seeing what the new continent has to offer, The Burning Cruisade looks like a fantastic expansion set and comes highly recommended for players that have gone into hibernation (like me), or newbies looking for something fun to take up 88 hours of their week.	I mean, don't get me wrong, I like the game and all, a nice classic shooter game which just kind of works, but there a lot of hackers and downright rude people on there. Last time I was on the game, all I heard was trash talk, and racism. Maybe that's why it's called, "Global offensive". But hey, if you don't mind that, it's a great game.
3	Destined to be a classic. This is one of those simulations that reminds you why you love the genre. It has all the fidelity, immersion, playability, polish, and graphical splendor that enriched classics like "Red Baron," "Aces over the Pacific," "Falcon 3.0," and "European Air War."	So if your just here for the solo campaign then id recommend it as its a lot of fun however if you want the online as well then i can't recommend it as R* is very money hungry and as such will ONLY fix a bug if it eats into their profits.

3.3 Text Preprocessing

The dataset was preprocessed before being used to train the models. Firstly, HTML tags and hyperlinks were removed. Next, the texts were converted into lowercase and contractions were expanded. Besides that, special characters were removed. This is followed by removal of numbers, single character words, extra whitespaces and stopwords, except for negations such as "no" and "not" because removal of such words would invert the sentiment of the reviews. Then, tokenization and part-of-speech (POS) tagging were performed. The POS tags were passed on to the lemmatizer so that lemmatization can be carried out based on the context of the tokens.

3.4 Data Labelling

The sentiments of the reviews were labelled as positive, negative or neutral by using pretrained sentiment analysis models of three libraries. The models used were NLTK's VADER Sentiment Intensity Analyzer, Textblob's Pattern Analyzer and Flair's TARS Classifier. A majority voting approach was used to determine the final sentiments of the game reviews. There were a total of 10426 positive reviews, 2975 neutral reviews and 2017 negative reviews.

3.5 Feature Extraction

The TF-IDF approach has been applied by using Scikit-learn's TfidfVectorizer to perform feature extraction. The "max_features" hyperparameter was set to 2500 while default values were used for other hyperparameters.

3.6 Handling Imbalanced Classes

Since the data contains a significantly greater number of positive reviews than neutral and negative reviews which may affect the performance of the models, the Synthetic Minority Oversampling Technique (SMOTE) was applied to adjust the distribution of the classes so that all classes have the same number of samples.

3.7 Model Applications

The machine learning algorithms used in this project are as follows:

a) Logistic Regression (LR)

LR is by default used for binary classification but it is extended by the Scikit-learn library to also perform multi-class classification.

b) Multinomial Naïve Bayes (MNB)

MNB is a probabilistic learning method used for classification with discrete features. Scikit-learn's MNB algorithm not only allows the use of integer feature counts, but also fractional counts obtained from TF-IDF.

c) Support Vector Classifier (SVC)

SVC is a classification algorithm that can be used to solve binary and multi-class problems. Scikit-learn's SVC algorithm uses a one-vs-one scheme to support multi-class classification.

d) Extreme Gradient Boosting Classifier (XGB)

XGB, is a decision-tree-based ensemble machine learning algorithm that implements gradient boosting. The XGBoost library provides a Scikit-learn wrapper class that allows the XGB algorithm to be used the same way as other Scikit-learn algorithms.

e) Multi-layer Perceptron Classifier (MLP)

MLP, is a feedforward Artificial Neural Network (ANN) algorithm that consists of multiple fully connected layers. Scikit-learn's MLP algorithm provides a regularization term that can be used to constraint the size of the weights in the neural network to prevent overfitting.

All the models were trained with their default hyperparameters to obtain their baseline performances except for MLP. Scikit-learn's MLP algorithm has a default architecture that consists of one input layer, one hidden layer with 100 neurons and one output layer, which causes the model to be computationally expensive to train. Therefore, a smaller value for the "hidden_layer_sizes" hyperparameter was set. The MLP model trained comprised 2 hidden layers, with 10 neurons in the first hidden layer and 5 neurons in the second hidden layer. The number of hidden layers and neurons were set arbitrarily as the model only acts as a baseline model before hyperparameter tuning was performed.

3.8 Hyperparameter Tuning

In order to improve the performance of the models, a Randomized Search Cross Validation with 3 splits was carried out to find the best combination of hyperparameters. In addition, a Grid Search Cross Validation with 3 splits was also performed on TF-IDF to select the best hyperparameters for it to further improve the performance of the models.

4.0 RESULTS AND DISCUSSION

Table 2 and Table 3 show the baseline performance of the models trained on the imbalanced dataset and oversampled dataset obtained through cross validations. Weighted precision, weighted recall and weighted F1-score were used as the metrics as they take into account the number of instances in each class.

Table 2. Baseline performance of all models trained on imbalanced dataset

	Accuracy	Weighted Precision	Weighted Recall	Weighted F1-Score		
				Negative	Neutral	Positive
LR	74.8%	71.7%	74.8%	71.6%	71.1%	72.2%
MNB	68.3%	63.3%	68.3%	57.2%	56.8%	56.9%
SVC	72.9%	69.9%	72.9%	66.6%	67.4%	67.6%
XGB	75.9%	73.5%	75.9%	74.1%	72.6%	73.3%
MLP	69.4%	70.1%	69.4%	70.1%	69.1%	70.0%

Table 3. Baseline performance of all models trained on oversampled dataset

	Accuracy	Weighted Precision	Weighted Recall	Weighted F1-Score			Status
				Negative	Neutral	Positive	
LR	79.3%	79.6%	79.3%	79.1%	79.5%	79.6%	Rejected
MNB	67.4%	67.5%	67.4%	67.2%	66.7%	67.1%	Rejected
SVC	87.7%	88.7%	87.7%	87.4%	87.4%	88.0%	Accepted
XGB	79.7%	80.1%	79.7%	80.0%	79.9%	79.5%	Rejected
MLP	86.8%	87.0%	86.8%	86.5%	86.8%	86.9%	Accepted

Based on the results in Table 2 and Table 3, oversampling has significantly improved the performance of all the models except MNB which was observed to have a drop in accuracy and weighted recall. The improved performance was due to the class distribution being balanced after performing duplication of data to synthesize new data from the minority classes.

MNB performed poorer on the oversampled data and was the worst-performing model most likely due to its assumption that all features are independent which is rarely true in real-world use cases where there are a large number of features.

The most significant improvement of performance was observed in SVC and MLP. These two models worked well with the larger, balanced dataset and they were also the two best-performing models. Therefore, they have been shortlisted for hyperparameter tuning and their performances after hyperparameter tuning were evaluated through cross validation. The fine-tuned models' performances are shown in Table 4.

Table 4. Performance of the fine-tuned models

	Accuracy	Weighted Precision	Weighted Recall	Weighted F1-Score			Status
				Negative	Neutral	Positive	
SVC	89.7%	90.0%	89.7%	89.2%	89.7%	90.1%	Accepted
MLP	87.0%	87.1%	87.0%	86.7%	87.0%	87.0%	Rejected

Table 4 shows that hyperparameter tuning has improved both models' performance. Hyperparameter tuning is able to improve the models' performance because it determines the best combinations of hyperparameters which produce optimal models that minimize the loss functions.

SVC outperformed MLP in terms of accuracy, weighted precision, weighted recall and weighted F1-score after the hyperparameter tuning. Hence, SVC as the best-performing model among all the models, was selected to test the effect of hyperparameter tuning of TF-IDF on its performance. The tested values of the TF-IDF hyperparameters and the best values determined by Grid Search Cross Validation are shown in Table 5.

Table 5. Tested hyperparameter values and the best values

Hyperparameter	Tested values	Best value
max_features	2500, 5000, 10000	2500
max_df	0.25, 0.5, 0.75	0.25
ngram_range	(1, 1), (1, 2), (1, 3)	(1, 1)

Based on the result in Table 5, SVC had the best performance under the condition in which the top 2500 terms across the corpus ordered by term frequency were considered, terms that occurred in more than 25% of the documents were ignored and only unigrams were extracted. The performance of SVC trained with the features extracted by the fine-tuned TF-IDF is shown in Figure 2.

	precision	recall	f1-score	support
negative	0.97	0.91	0.94	2044
neutral	0.91	0.87	0.89	2082
positive	0.87	0.96	0.91	2130
accuracy			0.91	6256
macro avg	0.92	0.91	0.91	6256
weighted avg	0.92	0.91	0.91	6256

Figure 2. Classification report of SVC with fine-tuned TF-IDF.

Based on the classification report in Figure 2, hyperparameter tuning on TF-IDF has improved SVC's performance. The model has achieved an accuracy of 91%, precision of 92%, recall of 91% and F1-score of 91%.

5.0 CONCLUSION

In conclusion, five machine learning models have been trained with game reviews obtained from Metacritic and Steam. It was shown that performing oversampling on the imbalanced dataset significantly improved the performance of most models. Besides, it was also shown that performing hyperparameter tuning on the models and TF-IDF resulted in better performance. Furthermore, we have determined Support Vector Classifier as the best-performing model among the five models with an accuracy of 91 percent. Its excellent performance could be attributed to the way it performs classification, which is based on hyperplanes instead of probabilities. It is suitable for text classification tasks with a large number of features such as sentiment analysis.

Through this project, game developers and studios would be able to see the value in performing sentiment analysis on users' opinions to make better decisions in game development. Sentiment analysis allows them to understand the needs and wants of their users and design their games to conform to it. Not only that, they would be able to identify and resolve their users' pain points. In the long run, their user base will be guaranteed to increase.

Future work should focus on analyzing the sentiment of emoticons and emojis as they are widely used by users in gaming platforms to express their feelings. In addition, ensemble methods can be experimented to build more robust sentiment analysis models.

6.0 ACKNOWLEDGEMENTS

Authors thank the Faculty of Computing and Information Technology, Tunku Abdul Rahman University College for financial support and resources to carry out this project.

REFERENCES

Al Amrani, Y., Lazaar, M., & El Kadiri, K. E. (2018). Random forest and support vector machine based hybrid approach to sentiment analysis. *Procedia Computer Science*, 127, 511-520.

Alboaneen, D. A., Tianfield, H., & Zhang, Y. (2017, December). Sentiment analysis via multi-layer perceptron trained by meta-heuristic optimisation. In *2017 IEEE International Conference on Big Data (Big Data)* (pp. 4630-4635). IEEE.

Ashari, A., Paryudi, I., & Tjoa, A. M. (2013). Performance comparison between Naïve Bayes, decision tree and k-nearest neighbor in searching alternative design in an energy simulation tool. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 4(11).

Boyd, C. R., Tolson, M. A., & Copes, W. S. (1987). Evaluating trauma care: the TRISS method. Trauma Score and the Injury Severity Score. *The Journal of trauma*, 27(4), 370-378.

Britto, L. F., & Pacıfico, L. D. (2020) Evaluating Video Game Acceptance in Game Reviews using Sentiment Analysis Techniques. In *Proceedings of SBGames 2020* (pp. 399-402).

Chakraborty, S., Mobin, I., Roy, A., & Khan, M. H. (2018, December). Rating Generation of Video Games using Sentiment Analysis and Contextual Polarity from Microblog. In *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)* (pp. 157-161). IEEE.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

Lau, K. W., & Wu, Q. H. (2003). Online training of support vector classifier. *Pattern Recognition,* 36(8), 1913-1920.

Prabhat, A., & Khullar, V. (2017, January). Sentiment classification on big data using Naïve Bayes and logistic regression. In *2017 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-5). IEEE.

Xia, R., Zong, C., & Li, S. (2011). Ensemble of feature sets and classification algorithms for sentiment classification. Information sciences, 181(6), 1138-1152.

Zuo, Z. (2018). Sentiment analysis of steam review datasets using naive bayes and decision tree classifier. *Student Publications and Research - Information Sciences*. http://hdl.handle.net/2142/100126