Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 17, No. 3, 2026

A Three-Channel Adaptive Fusion Model for Game Pre-Order Evaluation Based on Multiple Instance Learning

Wendi Li *

School of Foreign Language, Xiangtan University, Xiangtan, Hunan, China
*** Corresponding author Email:** 1457071672@qq.com

Abstract: Game pre-ordering has become an important marketing method in the game industry; dozens of players choose their beloved games in this method. However, not all times of game pre-ordering come with a pleasant result, many players claimed that they were deceived by the publisher. We are enlightened to develop a tool based on machine learning and NLP to improve such situations and provide a suggestion of buying it or not.

Keywords: Social Media Analytics; Machine Learning; Bert; Regression Task; TF-IDF; NLP; Game Rating Predictions.

1. Introduction

Nowadays, Game pre-ordering has become an important marketing method in the game industry. However, when customers face pre-ordering a new video game, they are facing an awfully imbalanced position, the information presented are normally designed to be attractive and flawless, which makes it almost impossible to identify what games are worth the cost, and what not. In practice, such costly mismatches between pre-order and final quality of the game have repeated themselves, including Cyberpunk 2077, Starfield or Dragon Age: The Veilguard.

To our researches on the existing studies, most studies were focusing on the predictions of sales volume, which means little attention was paid to the side of customers. In that case, A tool that available for the customers to make suggestions on buying games is essentially needed.

Our framework is based on weakly-supervised text learning. First of all, we will be collecting data mostly from "Billbill", a Chinese website. After some pre-work. Aiming to make algorithms learn how to tag the comments without any manual annotations, we applied a weakly-supervised models machine learning model combined with Bert and TF-IDF to extract features and stacking ensemble regression. And our article would be concluded as there parts, 1. the pre-work for data. 2. the extract algorithm. 3. the regression task.

And, to develop a tool like this is not just for the good of the customers, but for the producers to choose their marketing plan wisely.

2. The Pre-Work for Data

Our pre-work for data could be divided as five parts, first is filtering the time of the data while crawling, second is constructing bag-level annotations. Third is clean the data and the fourth is constructing game-level dataset and the last one is some further analysis for data to be sure what techniques should we be using.

2.1. Filtering the Data

Dates we got include the name of the game, the score of the game, the comments and the date of comment. Only comment before the releasing date is valid, according to what gamers were exactly facing the unknown quality of the game.

2.2. Constructing Bag-level Labels

Our data are saved in a CSV file. Data are divided under three labels, the name of the game, the comments and the score of the game. The score of the game would be the bag-level label for the game, and label the comment.

2.3. Cleaning the Data

As for the raw date we collected, we still need to clean them. For example, many content creators would have a giveaway event in their comment section, that would make those comments out value in predicting. So we have introduced Noise Keywords to block such data, including "subscribed, first one, followed, already repost".

2.4. Constructing game-level Dataset

We have collected over 170,000 valid comments from 38 different games. And divide 10 games into the test game collections. In order to improve the skills model using to learn how to label comments, we put the game posses extreme score into the training set.

3. Technical Basis

In this study, the entire framework is framed as Multiple Instance Learning (MIL) [1] problem. Semantic features of Chinese game reviews are obtained by using BERT (Bidirectional Encoder Representations from Transformers) [2], while local phrases are extracted by TF-IDF (Term Frequency-Inverse Document Frequency) [3]. And for the High-dimensional features are compressed by PCA (Principal Component Analysis) [4]. Our model consists of three channels, Semantic Channel, Structural Channel and Ensemble Channel. The Ensemble Channel is formed by four regressors, Ridge [5], Random Forest [6], Gradient Boosting Decision Trees [7] and Multi-Layer Perceptron [8]. The weights of each regressor and each channel are adaptively learned by Non-Negative Least Squares (NNLS) [9]. All models are implemented using scikit-learn [10]. (Identify applicable sponsor/s here. *(sponsors)*)

3.1. Weakly-Supervised MIL

- In traditional MIL, bags are labeled as positive or negative, while all the instances inside remain unlabeled. A positive bag, it contains at least one positive instance, and the others could be negative or neutral. Formally, bag $B = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}, Y \in \{0,1\}$, follows standard MIL assumptions:

$$Y = \begin{cases} 1, & \exists \boldsymbol{x}_i \in B \textit{is positive}, \\ 0, & \forall \boldsymbol{x}_i \in B \textit{is negative}. \end{cases} \quad (1)$$

- in the training set $\mathcal{D} = \{(B_1, Y_1), \dots, (B_N, Y_N)\}$, MIL would be to learn from bag-level supervised learning, and mark the key instances without any instances-level labels.
- In our prediction project, A game could be regarded as a bag, the comments are the instances and the score of the games are the labels of the bags. Since we are unable to knowing the exact contribution of single comment, we applied MIL to deal with such circumstances.

3.2. Text Representation Models

- In order to comprehend the semantic features from comments, we introduced BERT-Base-Chinese. By transforming non-structural Chinese comments into a fixed 768-dimensional vector to provide efficient features for MIL. In the purpose of deduce the effect of the noise from vast dimensions, after preliminary extractions, we introduced PCA (Principle component Analysis) to reduce the dimensions. In that case, key dimensions would be saved to a 200-dimensional vector and significantly ease the stress for the hardware. However, BERT is incapable of catching local key phrases, in order to make use of them, we introduced the TF-IDF (Term Frequency-Inverse Document Frequency).
- TF-IDF Would collect Term Frequency, the more frequent the term appears in the current bag, the higher TF it get. And for those terms that have higher frequency in all the bags, it possesses lower IDF (Inverse Document Frequency). However, it is incapable of understanding the meaning of sentences, which is why it could be combined with BERT. In order to cooperate with BERT, it would also provide a 200-dimensional vector for further work.

3.3. Regression Models and Ensemble Fundamentals

- We ensemble four models, Ridge Regression, RF (Random Forest), GBDT (Gradient Boosting Decision Tree) and MLP (Multi-Layer Perceptron) to form a Instance-Level Multi-View Prediction Ensemble Channel and combine it with Structural Channel and Semantic Channel to form a Prediction layer and apply NNLS (Non-Negative Least Squares) to calculate the weight of each channel and produce the final prediction. the Semantic Channel is formed by Ridge, and produce prediction using 25-dimension vector generated by BERT. And Structural Channel is formed by Ridge, produce prediction by construction features.

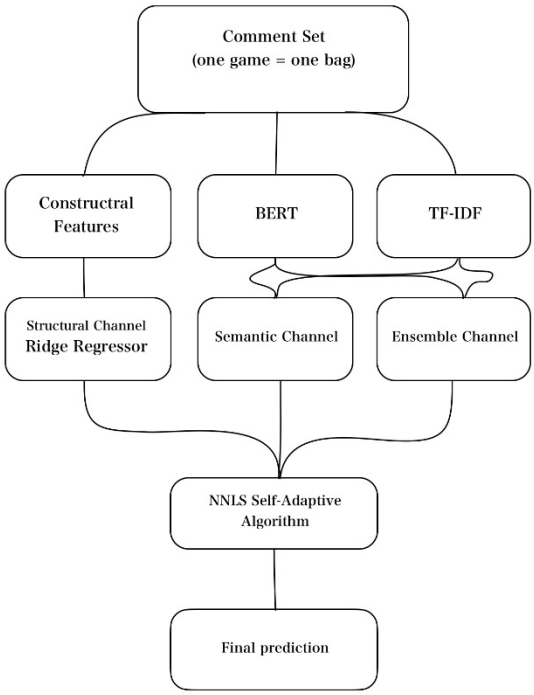

Fig 1. Framework of the multi-channel multi-view prediction model based on NNLS self-adaptive weight fusion

- Ridge Regression is a variation based on Linear Regression, written as

$$\min_{\boldsymbol{w}} \| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} \|_2^2 + \alpha \| \boldsymbol{w} \|_2^2 \quad (2)$$

which has great tolerance in noise. $\| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} \|_2^2$ is to fit the error and the $\alpha \| \boldsymbol{w} \|_2^2$ is applied for punishing the model when overfitting. With such design, it is capable of dealing with circumstances with few data. Which is why we apply it in Structural Channel and Semantic Channel for they are based on bag-levels that possess few data.

- RF is an ensemble regression method based on Bagging. In a given training set, through T times sampling with replacement, with sampled data we could form a Decision tree by minimizing the weighted sum of the left MSE (Mean Square Error) and the right after the split, written as:

$$\hat{y}_{RF} = \frac{1}{T} \sum_{t=1}^{T} h_t(\boldsymbol{x}) \quad (3)$$

where h_t is the t regression tree.

RF is robust to noisy features. In our channel, it could capture TF-IDF key words and reduce variance.

- GBDT is based on Boosting, aiming to fit the residuals and construct a residual function. By adding the residual function to the current function, we would acquire a more accurate function, written as:

$$F_m(\boldsymbol{x}) = F_{m-1}(\boldsymbol{x}) + \eta \cdot h_m(\boldsymbol{x}) \quad (4)$$

where F_m is the model after m times learning. η is the learning rate and the $h_m(x)$ is the residual function. In our channel, it could reduce bias. With the combination between GBDT and RF, we have strong model to deal with the variance and bias.

- MLP is a feedforward neural network. In its model, each input features is connected to neurons in the first

layer and weighted. Through linear transformations and activated by Activation Function, they would have been propagated through successive layers to produce prediction.

$$\boldsymbol{h}^{(l)} = \sigma(\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)}) \quad (5)$$

Where $h^{(l)}$ is the output of the layer l, and the $h^{(l-1)}$ is the output of the previous layer. $w^{(l)}$ is the weight matrix of current layer, σ is the Activation Function, which is ReLU in this article.

- To integrate the four regressors, we introduced Non-Negative Least Squares (NNLS), it would apply the Lawson-Hanson algorithm to calculate the proper weight for each regressors for better prediction. Written as:

$$\min_{\boldsymbol{w}\geq 0} \| \boldsymbol{Aw} - \boldsymbol{y} \|_2^2 \quad (6)$$

Where $A \in \mathbb{R}^{N\times M}$ is a meta-feature matrix, N is the bags (games in this article), and M is the number of channel, each row of A contains the prediction of the scores of the bags made by each channels. w is the weight vector to weight each channels. And $y \in \mathbb{R}^N$ is the actual score of the bag. w is a weight matrix to be learned, after solving, the sum of w is normally not equal to 1, we still need to yield interpretable contribution ratios for each channel and final prediction, written as:

$$w_j^{norm} = \frac{w_j}{\sum_{k=1}^{M} w_k} \quad (7)$$

4. Experiment and Data

In this section, we would present what our experiment shows in this project. In order to prove the efficiency of each method we apply in this model, we did some ablation experiment. Also, to provide suggestions for players to decide buying the or not, we set a standard as 75, for those games higher than it, they would be assigned as worthy of buying. And for those games predicted between 65 and 80, we set a Buffer Zone to warn players pre-order it with caution.

4.1. Experiment Setup

To better evaluate our model, we split the games as described at chapter II. 40 games including those with prior release (e.g. The Legend of Zelda: The Tears of The Kingdom) and those new IPs (e.g. Black Myth: Wukong).

To assess our model, we introduce three regression metrics for Comment Level, MAE, RMSE and R^2. The classification threshold for classifying the positive (worthy of pre-ordering) is set at 75, with a Buffer Zone between 65 and 80 for cautious recommendations. And three metrics to assess our model in game level, Game-Level Accuracy, weighted F1, High-Risk Misclassification Rate and the Buffer Zone Proportion. Weighted F1 is the weighted average of the per-class F1 scores, where the weight of each class is its proportion of samples in the test set. It is used to evaluate classification performance under class imbalance: a high weighted F1 indicates that the model performs well on both the majority and minority classes, rather than achieving high accuracy by simply favoring the dominant class. And The High-Risk Misclassification is predicting a game of higher score than 75 under 65, vice versa.

For comparison, we include the four individual regressors applied in our ensemble channel—Ridge, Random Forest, GBDT, and MLP—along with their equal-weight ensemble and a constant baseline that always predicts the training set mean score. All experiments are conducted on a consumer-grade laptop with an Apple M-series chip, using BERT-Base-Chinese for semantic feature extraction and PCA for reducing the dimensions.

4.2. Comment-level Regression Analysis

We analyze the results of our experiment, which is shown below:

As shown in the table 1, we could conclude that their performances show a number of differences. Random Forest achieves best MAE (11.469) and R^2, followed by GBDT with its MAE (11.891) and R^2(0.3763). While the other two, MLP and Ridge were lag behind. As a conclusion, RF and GBDT are better suited in the mixed feature space of the dense semantic representations (from BERT) with sparse keyword features.

Table 1. Comment-Level Regression Analysis

Method	MAE ↓	RMSE ↓	R² ↑
Constant Baseline (Training Set Mean)	14.286	19.258	−0.0156
Ridge (Single Model)	13.623	17.839	0.1285
Random Forest	11.469	15.066	0.3784
GBDT	11.891	15.091	0.3763
MLP	12.766	16.251	0.2768
Equal-Weight Four-Model Average	11.788	15.316	0.3576
Three-Channel NNLS Fusion (Ours)	**8.668**	**11.389**	**0.6448**

According to the table 1, we conclude that our model did have some advantages over those models above. Our Three-Channel NNLS Fusion model displays better MAE, RMSE and R^2 compare to all the other single models. NNLS effectively reduces the MAE (11.788 to 8.668) and RMSE (15.316 to 11.389) and R^2 is improved from 0.3576 to 0.6448.

4.3. Game-level Classification

Table 2. Game-Level Classification

Metric	Equal-Weight Four-Model Average	Three-Channel NNLS Fusion (Ours)
Number of Test Games	10	10
Correct Predictions	6	**7**
Game-Level Accuracy	60.0%	**70.0%**
Weighted F1	0.5556	**0.6818**
High-Risk Misclassification Rate	30.0% (3/10)	**0.0% (0/10)**
Buffer Zone Proportion	30.0% (3/10)	40.0% (4/10)

After analyzing how our models perform in regression tasks, we put our attention to the Game-level classification. We introduced game-level accuracy, weighted F1, high-risk misclassification rate and Buffer Zone Proportion to assess our model.

As shown in the table 2, we compare the prediction

accuracy between Equal-weight Four-Model Average and Three-Channel NNLS Fusion. We can conclude that our method is ahead of the equal-weight one.

According to this table, game-level accuracy is 70%, weighted F1 is 0.6818, High-Risk Misclassification is 0.0% and the Buffer Zone Proportion is 40.0%. It is inferable that our model has the above-average accuracy in classifying positive and negative.

The key metric is the High-Risk Misclassification, which increases when model produce opposite predictions to the real game. Our model achieve 0.0% High-Risk Misclassification (30.0% for Equal-Weight Four-Model Average)means no games were predicted as the opposition, Which could be concluded as two reasons: first, the high R^2 the Three-Channel NNLS Fusion achieve; second, the Buffer Zone (65-80) absorbs many of those uncertain games and avoid such misclassification:the three wrongly predicted games received negative predictions while they are positive, the model classified them as "wait-and-see" instead of predicting them as unworthy of buying.

As for the Buffer Zone Proportion, our model is 10% higher than the Equal-Weight Four-Model Average. Considering 0.0% of High-Risk Misclassification, the model achieved better use when giving a suggestion on pre-ordering in the practice.

According to what we have analyzed, our model achieves higher accuracy and reliability.

4.4. NNLS Weight Analysis

In this section, we would analysis how NNLS weight each channel:

Table 3. NNLS Weight When with or Without Prior Releases

Channel	With Prior Releases	Without Prior Releases
Structural Model	0.3500 (35%)	—
Game-Level Text Model (Semantic)	0.2000 (20%)	1.0000 (100%)
Ensemble Model (Four Regressors)	0.4500 (45%)	0.0000 (0%)
Sum	**1.0000 (100%)**	**1.0000 (100%)**

As shown in the table 3, when having prior releases, the weight of Structural Model, Game-Level Text Model (Semantic) and Ensemble Model (Four Regressors) is 0.35, 0.20 and 0.45. The weight for the Ensemble Model is the highest (0.45), which indicates its high ability in predicting with four ensemble regressors. The weight of Structural Model is 0.35, which indicates that the prior releases serve as an important factor for predicting sequel how releases behave. As for the Game-Level Text Model, which weights the least (0.20). We can infer that it possesses some missing information from the Ensemble Model.

Table 4. Analysis of Games Without Prior Releases

Game	True Score	Fusion Prediction	Error	Text Model	Ensemble
Expedition 33	94.0	76.79	−17.2	76.79	68.32
Mindseye	38.0	45.66	+7.7	45.66	67.45
Unknown 9: Awakening	59.5	58.46	−1.0	58.46	68.27

As shown in the table 4, we could see that when having no prior releases, the Ensemble Model produces highly concentrated predictions around the dataset mean, which explains why NNLS dropped the weight of Ensemble Model—for the Text Model demonstrates clearly superior to Ensemble Model

4.5. Game-level Classification Details

In this section, we examined the detailed predictions:

For the ten predicted games, seven of ten games were predicted rightly, and without any High-Risk Misclassification. But three of ten games were underestimated by the model.

For the Zelda and Resident Evil 9, model rightly predicted them as recommended games, but for the other four positive games were into the Buffer Zone (only Expedition 33 was predicted as positive). And the three games of the four were predicted in opposition (positive to negative).

Table 5. Detailed Predictions for each game

Game	True	Label	Pred	P-Label	Corr	Verdict
Zelda: Tears of the Kingdom	97.0	Pos	97.56	Pos	✅	Rec
Resident Evil 9	92.5	Pos	86.96	Pos	✅	Rec
Expedition 33	94.0	Pos	76.79	Pos	✅	Wait
Lies of P	87.0	Pos	65.76	Neg	❌	Wait
Death Stranding 2	93.0	Pos	73.43	Neg	❌	Wait
The Outer Worlds 2	82.5	Pos	69.63	Neg	❌	Wait
Dragon Age: Veilguard	72.0	Neg	59.06	Neg	✅	Not Rec
Battlefield 2042	58.0	Neg	58.94	Neg	✅	Not Rec
Unknown 9: Awakening	59.5	Neg	58.46	Neg	✅	Not Rec
Mindseye	38.0	Neg	45.66	Neg	✅	Not Rec

And the negative games were all predicted rightly, it appears that model achieved ability in identifying negative games, though the accuracy of predictions was not satisfying (12.94 of distance for Dragon's Age: Veilguard)

According to the analysis above, we could conclude that the model is avoiding high-risk predictions, the buffer zone and the three-channel NNLS fusion together enable a conservative yet safe prediction strategy (three games were into the buffer zone) because the model defers judgement on uncertain games by placing them into the Buffer Zone instead of predicting them wrongly and providing harmful suggestions, and demonstrates strong ability to identify negative games.

4.6. Conclusion and Future Direction

In this section, we summarize the key findings and reflect the limitations.

1) Our main framework is based on the Three-Channel design, Structural Channel provides a steady anchor for predictions, Semantic Channel provides supplemental information for the Ensemble Channel, and the Ensemble Channel provides signals with multi-view and fusion, but in certain circumstances, it would fail to work—that is when NNLS play its function, it enables the three channels organized organically and when Ensemble Channel fails to work, it would turn to the Semantic Channel and keep the model works.

2) We did not do ablation experiments. First, NNLS distribution itself proves channel utility (one channel is useful when its weight is not zero); Second, by comparing the single models and the Ensemble model, we have already proved its efficiency.

3) However, due to the hardware constraints us to do with more data and bigger pre-trained model. 38 games and the number of comments were not sufficient, data resources should have been multiple and not just one single platform. Structural features should have been more various—type of

the games, the scale of the workshop—and not just the score of the prior releases. And examine the model with rigorous ablation experiment.

References

[1] Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence, 89(1–2), 31–71. https://doi.org/ 10.1016/S0004-3702(97)00034-3.

[2] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. https: //arxiv.org/abs/1810.04805.

[3] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5), 513–523.

[4] Jolliffe, I. T. (2002). Principal component analysis (2nd ed.). Springer.

[5] Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55–67. https://doi.org/ 10.1080/ 00401706. 1970. 1048 8634.

[6] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324.

[7] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451.

[8] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536. https://doi.org/10.1038/323533a0.

[9] Lawson, C. L., & Hanson, R. J. (1974). Solving least squares problems. Prentice-Hall.

[10] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.