Context-Aware Comment Prediction: Sentiment, Score, and Critic Classification from Reviews

Kaiwen Zhu
Department of ECE
University of Toronto
kaiwen.zhu@mail.utoronto.ca

Abstract

This project applies machine learning to analyze movie reviews by predicting sentiment, estimating review score and identifying top critics. By using BERT embedding for feature extraction and XGBoost and CatBoost models for prediction, the proposed model achieved 80% accuracy in sentiment classification, a mean squared error of 2.82 for score prediction and 77% accuracy in professional critic identification.

1 Introduction

We are living in an era where sharing information, especially opinions and experiences, has become easier and more widespread than ever. Nowadays, online reviews play a crucial role for people when deciding whether to watch a movie or buy a product. However, there are thousands of reviews online, and it is difficult to read and understand them all. In this project, we focus on movie rating and reviews, and aim to use machine learning to perform three tasks: predicting review sentiment (positive/negative), predicting review score and identifying whether the review is written by a top critic or regular user. The project goal is to turn long, opinion-rich reviews into more direct sentiment labels, positive or negative, and also use a score to quantify the review. The motivation behind our project is to help the users to make sense of the public and expert opinions quickly in this information-overwhelmed society and support their decision-making efficiently.

ECE 1513: Introduction to Machine Learning (Winter 2025).

2 Preliminaries and Problem Formulation

In this project, our goal is to build a system that can automatically analyze movie reviews and predict useful information from the users. Specifically, the problem is a supervised machine learning task where the inputs are movie reviews in plain text format, and outputs are:

- Sentiment Classification — A binary classification task to predict if a review is positive or negative.
- Score Prediction — A regression task to predict the score on a scale from 0 to 10.
- Critic Identification — A binary classification task to determine whether the review is written by a top critic or an ordinary reviewer.

To successfully complete this project, we had to learn and apply following concepts:

- **Bidirectional Encoder Representations from Transformers (BERT)**
  BERT is a pre-trained language model developed by Google. It is designed to understand the context of words in a sentence by looking at both the left and right sides (bidirectional). We will use BERT to convert review text into numerical vectors called embeddings, which capture the semantic meaning of the text. These embeddings serve as input features for our machine-learning models.
- **XGBoost and CatBoost**
  XGBoost and CatBoost are machine learning models that build many small decision trees in a row. Each of these trees tries to fix the errors made by the previous ones and hence learn complex patterns. XGBoost is fast and accurate, while CatBoost is good at handling data with categories like "yes/no" or "A/B". In this project we will use XGBoost and CatBoost in sentiment classification, regression models to predict review scores and critics identification tasks.

3 Solution via Deep Learning

The proposed solution uses BERT embeddings for feature extraction, followed by several ML models for multi-task predictions.

3.1 Dataset

We use a real-world dataset of movie reviews from Rotten Tomatoes, which contains reviewer comments, sentiment labels, critic status and various formats [1]. The dataset will be cleaned, mapped to labels, normalized and filtered to ensure data quality. In addition, the dataset was randomly split into 80% for training and 20% for testing. Details will be explained in section 4.2.

3.2 Embedding Layer

We then use the bert-base-uncased model from Hugging Face Transformers to extract deep semantic features from the processed review dataset. Each review is tokenized and passed through BERT, and the output from the final hidden layer is mean-pooled to produce a fixed-size embedding. These embeddings are able to capture the context and meaning of the text and serve as input features for downstream prediction tasks.

3.3 Models

The BERT-generated embeddings will be used as input features to train three separate machine learning models, each targeting a specific task identified in section 2. For sentiment classification, we use XGBoostClassifier to predict whether a review was positive or negative. For score prediction, we use XGBoostRegressor to estimate the normalized review score on a 0–10 scale. In addition, we use a CatBoostClassifier to identify whether a review was written by a top critic or a regular reviewer.

3.4 Testing and Evaluation

For each targeted task, the model performance is assessed based on following metrics and threshold: Sentiment Classification: Accuracy > 0.7 [2], F1 Score > 0.8 [3]; Score Prediction: Mean Squared Error as close to 0 as possible [4] and Critic Identification: Accuracy > 0.7 [2], F1 Score > 0.8 [3].

4 Implementation

This section describes the machine learning approach used for integrating data preparation, feature extraction, model building, training and testing with visualization. Data flow as shown in Figure 1.

4.1 Data Preparation and Feature Extraction

Data Loading and Sampling: The process begins by loading the dataset from a CSV file containing movie reviews. To ensure manageability without sacrificing representativeness, a random sample of 40% of the data is selected. This sampling step reduces the overall dataset size, making subsequent processing and experimentation more efficient.

Data Cleaning: After sampling, the data is cleaned by removing records with missing values in key fields such as review content, review type, and critic status, and by filtering out entries with empty scores. This step ensures that only records with complete and relevant information are used for analysis, thereby improving the reliability of the results.

Field Standardization: Next, the review scores are normalized to a consistent 0–10 scale through a custom function that can handle different formats, for example scores expressed as fractions, letter grades, or direct numerical values. In addition, any entries where the sentiment conflicts with the normalized score are removed to further enhance data quality.

Text Feature Extraction with BERT: Finally, the cleaned review texts are converted into numerical features using a pre-trained BERT model. The texts are tokenized, padded, and truncated to a fixed maximum length before being processed by the model, and then the output of the last hidden layer is averaged to get the final review representation. These BERT embeddings, along with the derived labels and scores, will be the input of subsequent modeling tasks.

4.2 Model Building and Task Definition

Sentiment Classification: For sentiment classification, an XGBoost classifier is employed to predict whether a review is positive or negative. The model is trained using the BERT embeddings as features, and its performance is evaluated through accuracy metrics, confusion matrices, and detailed classification reports.

Score Regression: For predicting review scores, an XGBoost regressor is applied. This model is trained to minimize the mean squared error (MSE) between the predicted scores and the actual normalized scores. The performance of the regression model is also visualized with scatter plots comparing actual values against predictions.

Critic Identification: In addition, a CatBoost classifier is used to determine whether a review was written by a top critic. The classifier is trained on the same BERT-based features, and its performance is measured by calculating accuracy and reviewing confusion matrices and classification reports, which provide insights into its ability to correctly identify reviews from top critics.

4.3 Training and Tuning

For training, we use 80% of the data as the training dataset. As a baseline, we built and trained a simple regression model using TF-IDF features with a Random Forest Regressor. Though it performed okay on the training set, it showed signs of overfitting and produced a high MSE. Then we moved

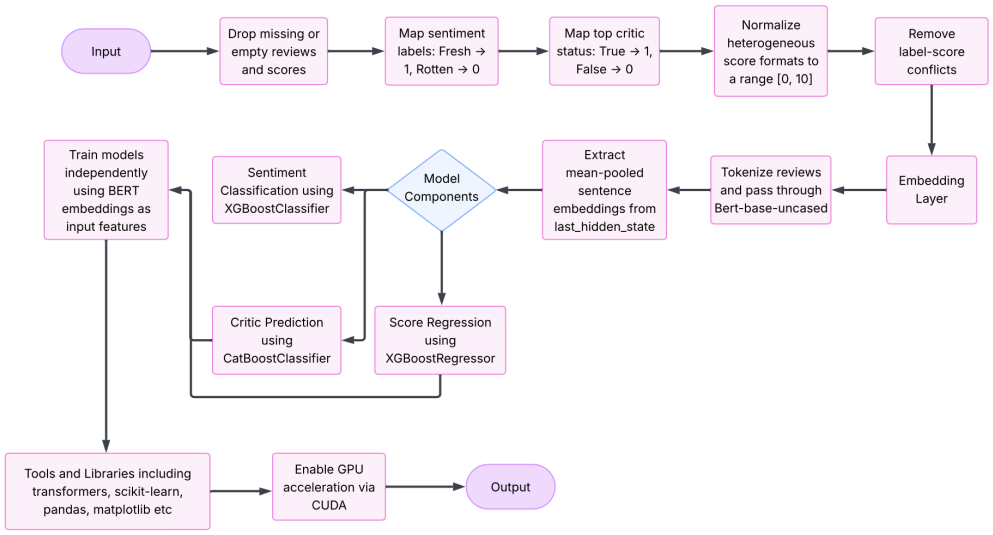

Figure 1: Project Data Flow

to the BERT-based model described in 4.2. For the `XGBoostClassifier` and `XGBoostRegressor`, we experimented `n_estimators` in the range of 100 to 300, `learning_rate` between 0.05 and 0.3, and `max_depth` between 3 and 7. The objective function was set to `"binary:logistic"` for sentiment classification and `"reg:squarederror"` for score prediction, depending on the task. For the `CatBoostClassifier`, we adjusted `iterations` from 100 to 300 and `depth` from 4 to 6. Since the dataset contains more regular reviewers than top critics, we also applied `class_weights` to encourage better recognition in minority class.

During training , we observed that increasing the number of trees and slightly lowering the learning rate improved stability in both classification and regression tasks. However, the improvements becomes marginal beyond 100 estimators.

Our final models were trained with `n_estimators=100` for `XGBoostClassifier` and `XGBoostRegressor`, and `iterations=100` for `CatBoostClassifier`. These settings resulted in 85% accuracy and an F1-score of 0.95 for positive reviews in Sentiment Classification achieved. MSE of 2.57 for Score Prediction. 84% accuracy and F1-score of 0.89 in Critic Identification. However, the performance on identifying top critics is low with a F1-score of 0.30.

5 Numerical Experiments

After training the models, we evaluated their performance on the test set to measure how well they generalized to unseen data.

In the sentiment classification task, the XGBoost model achieved an overall accuracy of 80%. The F1-score for the "fresh" class was 0.85, while the "rotten" class reached 0.70. The confusion matrix shown as Figure 2 left graph, illustrates that the model tended to over-predict positive sentiment, which is likely due to mild class imbalance. Compared to the 72% accuracy achieved by the baseline logistic regression model trained on TF-IDF features, the BERT-based classifier had a significant improvement in capturing context-aware sentiment.

For the task of scoring regression, the BERT-based model significantly outperformed the baseline model. To be more specific, the baseline model has shown high R-squared values in the training set, but its test predictions were widely scattered and did not align well with the actual scores. This indicates that the model had overfit the training data and failed to generalize. In the end, the baseline model produced a 4.14 MSE in the test set. However, the BERT-based approach had better results with a MSE of 2.82 and a RMSE of around 1.68. The predicted scores maintained a high degree of linear relationship with the ground truth as shown in Figure 2 middle graph, while only deviating

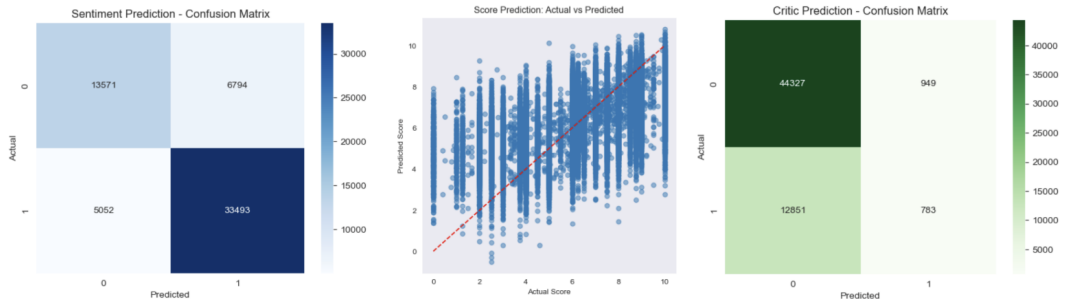

Figure 2: Visualization of Testing Result

slightly at the lower and upper ends. This shows that BERT embeddings work better than TF-IDF for predicting review scores because they reduce overfitting and improve generalization.

The top critic classification task was particularly difficult because there was a severe imbalance between the classes. The baseline model achieved 77% test accuracy, but this was misleading. Because it simply predicted most reviews as non-critics. Though it achieved an F1-score of 0.87 for regular reviewers but it completely failed to detect top critics with a 0 in F1-score. In contrast, the BERT based model achieved the same overall accuracy but handled the minority class better. It improved the F1-score for top critics from 0.00 to 0.10, and precision increased from 0.25 to 0.45. This shows that BERT embeddings helped the model recognize critic reviews more effectively. However, recall for top critics remained low with only 0.06. Also, as shown in Figure 2 right graph, 12851 critic reviews as been classified as normal reviews. This means while BERT improved representation quality, class imbalance remained a major limitation in this task.

Overall, our results indicate that BERT embeddings provide strong semantic representations for review data. The accuracy, F1-score and MSE for the three tasks all achieved our identified metrics threshold in section 3.4. However, challenges remain in handling imbalanced tasks such as critic classification.

6 Conclusions

In this project, we developed a machine learning solution that analyzes movie reviews using BERT embeddings and tree based models XGBoost and CatBoost. The system performed well across three key tasks: sentiment classification, score prediction, and critical identification, with 80% accuracy, mean square error 2.82 and 77% accuracy respectively.

During the implementation, we discovered that data quality is important, especially when dealing with human sentiment, which is often tricky. So we carefully cleaned and aligned the dataset to improve consistency. However, as human emotion is complex, there can be meaning behind seemingly contradictory reviews like low mark with "positive" comment. Therefore cleaning dataset is not the best practice, a more advanced solution is needed for analyzing the reviews. Moreover, the performance improvement from a linear regression to a BERT-based model was not very significant. This suggests that both types of models may share the same limitations and highlights the need for more powerful approaches.

To overcome this, the next step is moving forward to Large Language Models (LLMs). LLMs have a deeper understanding of the context and nuance. For example, ChatGPT and Google PaLM have already shown strong performance in tasks like sentiment detection and content summarization. Therefore, leveraging these models and exploring fine-tuning or prompt-based learning will be the path toward more accurate and robust online review analysis.

References

[1] S. Leone, "Rotten Tomatoes Movies and Critic Reviews Dataset," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/stefanoleone992/rotten-tomatoes-movies-and-critic-reviews-dataset.

[2] "What Is a Good Accuracy Score in Machine Learning?" Deepchecks. [Online]. Available: https://www.deepchecks.com/question/what-is-a-good-accuracy-score-in-machine-learning/.

[3] N. Buhl, "F1 Score in Machine Learning Explained," Encord, Jul. 18, 2023. [Online]. Available: https://encord.com/blog/f1-score-in-machine-learning/.

[4] J. H. Cabot and E. G. Ross, "Evaluating Prediction Model Performance," Surgery, vol. 174, no. 3, pp. 723–726, Sep. 2023. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC10529246/.