CSC508
Sentiment Classification of Starfield player reviews
Alexander Kaler

Task Definition

For my project I have developed a machine learning system that can classify Steam game reviews as either positive or negative. Understanding user sentiment is important for game developers and publishers to provide actionable insights for improving game quality, tailoring updates and refining marketing strategies. My model aims to automate the process of large scale data review which would otherwise be difficult and time consuming to process manually.

The input to the system is the text of a Steam game review and the output is a binary classification. 1 for a positive sentiment and 0 for a negative sentiment. To evaluate the model I used several metrics including accuracy, precision, recall, F1-score and ROC-AUC each offering a nuanced view of the system's performance. Accuracy provides a general measure of correctness, while precision and recall highlight the system's handling of false positives and false negatives. The F1-score balances these two, and the ROC-AUC measures the model's ability to differentiate between positive and negative classes.

Infrastructure

The dataset for this project consists of game reviews scraped directly from Steam using the requests library. The game I chose to scrape the reviews for is Starfield, because it was a big deal on release and took a great deal of negative feedback. I knew that the game would have several reviews to utilize for this project. While I think that positive reviews are important, I wanted to choose a game with a large amount of negative feedback, as I think this model would best be utilized in identifying negative feedback for actionable insight. I collected the reviews in three different dataset sizes, 1,000, 5,000 and 10,000 reviews. This was to test the models performance across varying dataset sizes.

Each review is labeled as either positive or negative based on its "thumbs up" or "thumbs down" classification on Steam. The datasets include the following fields, review is the textual content of the review, voted up is a boolean indicating true for positive review and false for negative review, as well as the labels where 1 is a positive review and 0 is a negative review.

I performed some exploratory data analysis on the three datasets. I looked at the distribution of review lengths, which showed that the majority of reviews fell within the 2000-3000 characters in length, several of them being significant outliers. I also found that the negative class was quite dominant in the data. I decided to apply SMOTE, synthetic minority over-sampling technique, to balance the classes in the training set. This helped to avoid bias in the predictions toward the majority class. Word clouds were also provided for the most frequent words found in the positive and negative class of the 10k size dataset. This gave us some insight into words that may be closely associated with the positive and negative class. It looked like "bethesda", "game" and "like" were all dominant in both classes.

Data preprocessing steps ensured the data was clean and ready for the machine learning models. Tokenization had to be used for the Naive Bayes models; reviews were converted into numerical feature vectors using TF-IDF.

For the DistilBERT and BERT model the tokenizer from the Hugging Face transformers library was used to transform raw text into token embeddings suitable for the transformer-based learning. Each subset was split into 80% training and 20% testing datasets.

The following tools and libraries were used to implement the project.

- Requests: to scrape reviews for Starfield from Steam
- Scikit-learn: For preprocessing and implementing the Naive Bayes baseline model
- Transformers: for fine-tuning the DistilBERT and BERT models on the textual data
- Pandas: For efficient data manipulation and organization
- Matplotlib and seaborn: For visualizing the performance metrics of the models
- Ntlk: For n-gram analysis to extract and analyze the most indicative keywords and phrases contributing to the model predictions

Approach

The baseline model in this project was the Naive Bayes classifier. I chose this as the baseline model due to its simplicity and effectiveness in text classification tasks. Reviews were transformed into numerical feature vectors using TF-IDF, capturing the importance of words across the data. The model assumes feature independence, which while limiting makes it computationally efficient.

The advanced model in this project is the DistilBERT model, a lightweight transformer-based architecture, which was fine-tuned on the review dataset. As well as the standard BERT model, another transformer-based model. This helped in demonstrating the trade off of efficiency and accuracy, as the DistilBERT model is much more efficient but less accurate. These models capture contextual relationships between words, enabling them to understand nuances that a simple bag-of-words approach cannot. Key hyperparameters included were a learning rate of 1e-5, batch size of 8 and 3 epochs.

The Naive Bayes model provided a baseline for performance while the DistilBERT and BERT offered state-of-the-art results by leveraging their deep learning capabilities. The comparison highlights the tradeoffs between computational efficiency and classification accuracy.

Literature review

I found an experiment on Kaggle that did something very similar to what I did for my project, with a couple of key differences. Betlsazar's work on sentiment analysis of steam reviews provides a strong foundation for understanding how traditional machine learning models, such as Naive Bayes and Support Vector Machines, can be applied to classify user sentiment. His approach emphasizes broad applicability by analyzing reviews across various games, leveraging preprocessing techniques like tokenization and lemmatization to prepare test data for modeling. In contrast, my project focuses specifically on sentiment classification for reviews of

the game Starfield, combining a baseline Naive Bayes model with a transformer-based approach using DistilBERT and BERT. This distinction allows my analysis to delve deeper into game-specific language patterns and extract meaningful insights through keyword and phrase analysis, which are not emphasized in Beltazar's work. While his analysis provides generalized findings, my project explores more advanced NLP techniques and a more targeted dataset can uncover nuanced sentiment trends, particularly relevant in individual games with distinct player communities.

Error Analysis

When it came to error analysis I looked at a couple of different things. I wanted to see how the Naive Bayes, DistilBERT and BERT models scaled with higher sample sizes. As well as how the models compared to each other when using the 10k dataset.

The DistilBERT model scaled well with higher sample sizes, leveraging the additional data to improve its understanding of contextual relationships in the text. The model's performance consistently improved across all metrics, demonstrating its ability to generalize better with more data.

The BERT model started off with a high degree of accuracy, 89% on the 1k dataset, about 2% short of the DistilBERT model when using the 10k dataset. However as we increased the sample size its increase in accuracy was marginal. 91% on the 5k dataset and 92% on the 10k dataset. Making it slightly more accurate than the DistilBERT mode. However it had a much longer runtime, of around 50 minutes on the 10k dataset, while the DistilBERT model took roughly 20 minutes.

There are several bar chart comparison provided in the code, but the main comparisons of interest are Naive Bayes model comparison with Varying Dataset Sizes, Naive Bayes vs DistilBERT with Varying Dataset sizes, BERT with Varying Dataset Sizes and Naive Bayes Vs DistilBERT Vs BERT for the 10k dataset. Note that for the Naive Bayes vs DistilBERT with Varying Dataset Sizes, we are just interested in how the DistilBERT model scales with varying dataset sizes.

The Naive Bayes model shows notable improvement in performance as the dataset size increases. For smaller datasets, precision is relatively higher for the negative class compared to the positive class, indicating that the model is better at identifying true negatives. However as the dataset size increases, precision improves for both classes, particularly for the positive class. Recall starts to drop in the negative class as the dataset size increases, but the positive class consistently increases in recall. These trends are reflected in the F1-scores, where both classes show improvement as the dataset size grows, with the positive class achieving consistently higher performance. Accuracy also improves steadily with more data, underscoring that Naive Bayes benefits significantly from larger datasets, though its performance still falls short of transformer-based models.

The DistilBERT varying data sizes shows how the DistilBERT model scales as the dataset increases. Precision remains consistently high for both the positive and negative classes, showing that the model is effective at minimizing false positives regardless of dataset size. Recall trends differently for the two classes, recall for the positive class increases but decreases for the negative class. Suggesting a potential trade-off in the model's sensitivity to identifying negatives as it learns more data. The F1-score for the negative class remains relatively stable, balancing the model's precision and recall performance. These trends highlight that DistilBERT is robust with smaller datasets, but scales effectively, especially for the positive class, while showing some challenges in recall for the negative class with larger datasets.

The BERT model Comparison for Varying Dataset Sizes shows how BERTs performance evolves as the dataset size increases. Precision remains consistently high for both the negative and positive classes across all dataset sizes, indicating BERT's strong ability to minimize false positives even with smaller datasets. Recall, however, demonstrates a decrease in the negative class and an increase in the positive class. Suggesting that BERTs ability to identify true negatives diminishes with more training data. But the recall for the positive class suggests that BERTs ability to identify true positives increases. Accuracy increases overall as the dataset size increases.

Finally the Naive Bayes Vs DistilBERT Vs BERT demonstrates how these three models compare to each other with the 10k dataset size. BERT consistently performs the best by all metrics. It has a higher precision, recall, F1-score and accuracy compared to the other two models. DistilBERT consistently comes in second, and Naive Bayes last. This demonstrates the transformer based models ability to effectively balance precision and recall, adapt to complex patterns in the data, and generalize well across both classes, even when handling imbalanced or challenging datasets. However this also demonstrates the tradeoff of accuracy and efficiency. The BERT model took the longest to train, but performed better overall. The Naive Bayes model was the fastest, but was not able to keep up with the transformer based models.

The ROC curve comparison highlights the performance of Naive Bayes, DistilBERT and BERT in distinguishing classes. Naive Bayes demonstrates a strong classification ability, with an AUC of .94, but is outperformed by the transformer based models. DistilBERT achieves an AUC of .96, reflecting its improved ability to capture complex patterns and maintain a higher true positive rate at lower false positive rates compared to Naive Bayes. BERT had the highest AUC at .97, exhibiting the best overall performance.

Using keyword analysis we were able to identify the most indicative bigrams for positive and negative reviews in the DistilBERT model. The most indicative bigrams in the positive reviews include “Side quest”, “open world” and “great game”. While “great game” doesn't provide much in terms of actionable feedback, “side quest” and “open world” gives us insight into what aspects of the game people seemed to enjoy. Negative reviews included “game breaking”, “loading screens” and “paid mod”. These suggest that the game had some game breaking bugs, long loading times and monetization that players typically frown upon.

Works Cited

Beltsazar, Daniel. *Steam Games Reviews Analysis (Sentiment Analysis)*. Kaggle, https://www.kaggle.com/code/danielbeltsazar/steam-games-reviews-analysis-sentiment-analysis . Accessed 12/4/2024