Sentiment Analysis on Disneyland Reviews

NLP Final Individual Report

Name: Annie Cheng

Course Name: DATS6312 Natural Language Processing

Professor: Dr. Ning Rui

Due Date: Dec 5, 2025

1. Introduction

With the rapid growth of online reviews, extracting sentiment from large volumes of text has become increasingly important for businesses. Understanding customer sentiment helps organizations improve satisfaction, tailor services, and enhance profitability. In the tourism and hospitality industry, customer decisions are heavily influenced by shared experiences, especially on platforms like TripAdvisor where visitors provide detailed feedback.

This study analyzes Disneyland customer reviews (Arush Chillar, 2021)and develops NLP models to predict sentiment from text. These insights can support data-driven business strategies and improve decision-making.

In this project, my teammates and I each trained multiple NLP models using the Disneyland TripAdvisor dataset (Arush Chillar, 2021). For the initial stages, Junhua handled data preprocessing while I conducted the exploratory data analysis. Junhua later developed the Streamlit application, and I created the presentation slides. We collaboratively contributed to the group report and overall model development. Overall, our teamwork was effective and complementary.

2. Individual Contribution

My individual work in this project focused on exploratory data analysis, model development, and experimentation. I performed the EDA, including rating and sentiment distributions, n-gram frequency analysis, and word cloud visualizations.

For model development, I implemented and evaluated Naive Bayes and SVM using TF–IDF features. For transformer-based models, I fine tuned BERT and RoBERTa using HuggingFace, conducted hyperparameter configuration, and implemented multiple class imbalance strategies such as weighted loss functions and WeightedRandomSampler. After identifying RoBERTa as the strongest baseline transformer, I performed additional experiments involving alternative class weighting schemes, learning rate adjustments, increased epochs, and early stopping.

I also developed the sentence-transformer-based approaches, including MPNet with downstream classifiers (logistic regression and LightGBM) and the SetFit framework. This involved fine tuning MPNet with triplet loss, extracting embeddings, training classifiers, and evaluating SetFit's contrastive learning performance relative to fully fine tuned transformers.

3 Exploratory Data Analysis

In the exploratory data analysis, we mapped rating mapped ratings 1 and 2 to negative, 3 to neutral, and 4 and 5 to positive. Fig. 1 and Fig. 2 show that the distribution is heavily skewed toward positive reviews, indicating a class imbalance issue.

Fig. 3 and Fig. 4 highlight the most frequent bigrams and trigrams. "Fast pass" appears repeatedly across both, including phrases such as "get fast pass" and "fast pass system." The trigram "happiest place earth" is also highly frequent, consistent with the predominance of positive sentiment.

The word clouds in Fig. 5 further support these patterns. "Fast pass" appears prominently in positive reviews, while negative reviews frequently mention "ride," "time," "queue," and "line," suggesting concerns about wait times. In Fig. 7, neutral reviews contain overlapping terms such as "ride", "good", "time", and "queue".

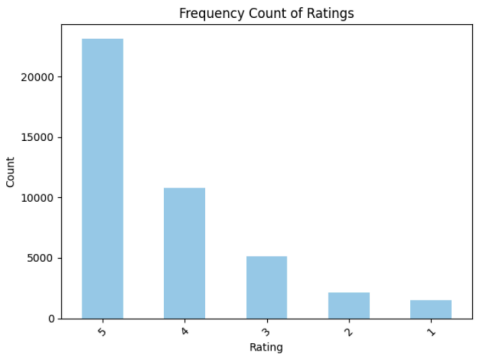

Figure 1: Distribution of Ratings

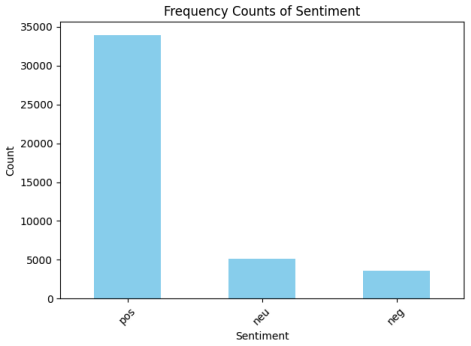

Figure 2: Distribution of Sentiments

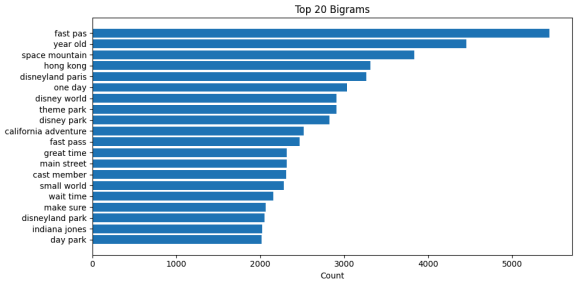

Figure 3: Top 20 Bigrams

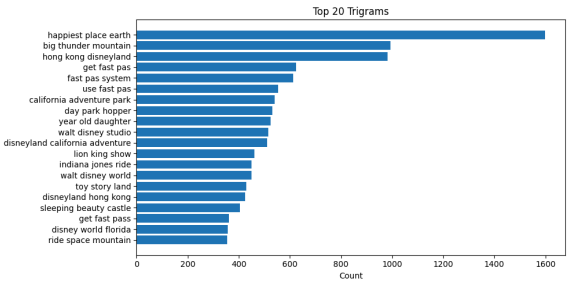

Figure 4: Top 20 Trigrams

Figure 5: Word Cloud (Positive)

Figure 6: Word Cloud (Negative)

Figure 7: Word Cloud (Neutral)

4. Model

4.1. Baseline Models: Naive Bayes, Support Vector Machine

Naive Bayes is a classification algorithm that requires only a small amount of training data to estimate the necessary parameters, making it relatively fast compared to more sophisticated methods. It is based on Bayes' rule and a set of conditional independence assumptions. Specifically, Naive Bayes assumes that all features are independent of one another given the target label. Under this conditional independence assumption and by applying Bayes' theorem, the classification rule becomes:

$$\hat{y} = \arg\max_{y_j} \; P(Y = y_j) \prod_i P(X_i \mid Y = y_j)$$

Support Vector Machine (SVM) constructs a hyperplane that maximizes the margin between classes. By maximizing the distance between the hyperplane and the nearest support vectors, SVM improves generalization

4.2. Transformer-Based Models

4.2.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) is designed to pretrain deep bidirectional language representations by jointly conditioning on both left and right context in every layer (Devlin et al., 2019). This bidirectional conditioning enables the model to capture richer contextual information compared to traditional left to right or right to left architectures. BERT employs attention mechanisms to capture dependencies between words in both directions within a sequence (Vaswani et al., 2017). This enables parallel computation and attention to the entire context of a word. The self- attention mechanism operation is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

BERT is pretrained on two unsupervised objectives. The first is masked language modeling, where a portion of tokens in the input sequence is randomly masked and the model learns to predict the original tokens from context. The second task, next sentence prediction, trains the model to determine whether two sentences appear consecutively in the original corpus. Together, these objectives allow BERT to learn sentence-level and token-level dependencies, enabling strong performance on downstream tasks.

The original paper reports two model sizes: BERTBASE with 12 transformer layers, hidden size 768, and 12 attention heads (110 million parameters), and BERTLARGE with 24 layers, hidden size 1024, and 16 attention heads (340 million parameters). Despite being pretrained on unlabeled text, BERT achieves state-of-the-art results across many NLP benchmarks and substantially improves performance even on low-resource tasks.

4.2.3 RoBERTa

RoBERTa is an optimized version of BERT. It introduces several key training improvements, including dynamic masking, removal of the next sentence prediction objective, training on longer input sequences, larger mini-batches, and a byte-level BPE tokenizer (Liu et al., 2019). These modifications allow the model to leverage more data and more diverse masking patterns, leading to substantial improvement. RoBERTa demonstrates that careful tuning of training procedures can significantly enhance the effectiveness of the original BERT architecture.

4.2.5 MPNet

MPNet combines ideas from BERT and XLNet by using a permuted language modeling objective together with auxiliary position information (Song, Tan, Qin, Lu, & Liu, 2020). This design reduces the position discrepancy introduced in XLNet and enables the model to learn both bidirectional and dependency aware contextual representations. As a result, MPNet produces high quality sentence embeddings and has been shown to perform strongly on semantic similarity and sentence-level tasks.

4.2.6 SetFit

SetFit is a few shot learning framework built on top of sentence transformers (Tunstall et al., 2022). It requires only a small number of labeled examples per class. For fine tuning, SetFit uses contrastive learning by constructing pairs of sentences. A pair is labeled as positive if both sentences share the same sentiment, and negative if their sentiments differ. During training, the embedding model learns to pull the embeddings of positive pairs closer together and push the embeddings of negative pairs farther apart. Because contrastive learning operates on pairs rather than individual samples, SetFit can effectively fine tune a sentence transformer even with very limited labeled data.

After the embedding model is fine tuned, SetFit trains a classifier on top of the generated embeddings. In contrast to the first stage, this step uses labeled samples directly rather than sentence pairs. All training sentences are passed through the fine-tuned embedding model to obtain sentence embeddings, which are then used to fit a logistic regression classifier, the default classifier in the SetFit framework.

5.1 Experimental Setup

5.3.1 Classical Machine Learning Setup

For the classical models (Naive Bayes and SVM), my teammate handled the text preprocessing steps, which included lowercasing, removing URLs and HTML tags, normalizing spaces, tokenizing, removing stopwords, and lemmatizing. After preprocessing, I used the cleaned text to implement TF–IDF feature extraction and trained Naive Bayes and SVM models using an 80/20 train–validation split stratified by sentiment. Because the dataset is imbalanced, I applied class-weighted loss to reduce bias toward the majority class. Macro F1 was used as the primary evaluation metric, supplemented with accuracy and per class metrics.

3.3.2 Transformer-Based Fine-Tuning Setup

For BERT and RoBERTa, I used a standard transformer fine-tuning pipeline. In data preparation, we concatenated the branch name with the review text and used an 80/20 train-validation split. The pretrained transformers I implemented are “bert-based-uncased”, and “roberta-base”.

For both models, I tokenized the input text with a maximum sequence length of 256 and used the AdamW optimizer with weight decay of 0.01 and a warmup ratio of 0.1. I first trained each model using the baseline hyperparameter configuration recommended in the original BERT paper (Devlin et al., 2019). Macro F1 is used as the primary evaluation metric because it gives equal importance to each sentiment class and is appropriate for imbalanced data. These baseline settings are summarized in Table 1.

Table 1: Baseline Configuration

	Learning Rate	Batch Size	Epochs	Class Weight Strategies
BERT	3e-5	16	3	
RoBERTa	3e-5	16	3	Inverse Frequency weighted loss

After comparing all models with

After evaluating all transformer models under the baseline configuration using macro F1, RoBERTa emerged as the best-performing model. My teammates and I used these baseline results to determine which model to focus on for further experimentation. Building on this shared baseline, I conducted additional fine-tuning experiments on RoBERTa to address class imbalance and improve performance. Specifically, I evaluated two weighting schemes—normalized inverse frequency and normalized square-root inverse frequency—applied through both the loss function and a WeightedRandomSampler. After identifying the best-performing weighting strategy, I ran further hyperparameter trials by lowering the learning rate to 2e-5, increasing the number of epochs to 5, and adding early stopping with a patience of 2 rounds to prevent overfitting.

3.3.2 Sentence Transformer + Classifier Setup

For the sentence embedding-based models, I used two approaches: SetFit and a sentence transformer with a downstream classifier. Both approaches used the same pretrained model, which was "all mpnet base v2". For the sentence transformer with classifier approach, we implemented a two-stage pipeline. In the first stage, I fine tuned the transformer using the batch all triplet loss function (Schroff, Kalenichenko, & Philbin, 2015). A triplet consists of three elements: an anchor text, a positive text that shares the same sentiment as the anchor, and a negative text with a different sentiment. The loss encourages the model to pull the anchor and positive embeddings closer together while pushing the anchor and negative embeddings farther apart. Once the transformer was fine-tuned, we extracted embeddings from the model and trained downstream classifiers on these embeddings. In our case, I used logistic regression and LightGBM as the downstream classifiers.

For SetFit, I selected 32 samples per class and fine tuned the sentence transformer using sentence pairs. SetFit then automatically trained a logistic regression classifier on the embeddings produced by the fine-tuned transformer.

Table 2: Sentence Transformer Hyperparameter

	Batch Size	Epochs	Number of Iterations	Learning Rate	Loss Function
MPNet	8	3	Not Applicable	3e-5	Triple Loss
SetFit	16	3	20	Not Applicable	Not Applicable

4. Results and Discussion

Among Naive Bayes and SVM, SVM achieved the stronger performance with a macro F1 of about 0.62 (Table 3). From the classification reports (Tables 4–5), both classical models struggled with the neutral class, which aligns with the observed class imbalance in the dataset.

Based on the baseline transformer configuration, RoBERTa outperformed BERT, which aligns with expectations given RoBERTa's improved training procedure. After experimenting with different class weighting strategies, the normalized square root inverse frequency loss yielded the best performance for RoBERTa, although the model still did not achieve balanced performance across classes.

I then applied a tuning experiment by lowering the learning rate to 2e5, increasing the number of epochs to 5, and adding early stopping. This configuration did not surpass the baseline. With this setup, the best macro F1 score occurred at epoch 2, while the later epochs showed minor fluctuations before declining. This suggests that the model was no longer learning additional generalizable signal after the early epochs. We suspect it is likely due to the challenges posed by the imbalanced and semantically ambiguous neutral class. Zhang et al. (2020) reported similar challenges in a Yelp review study, where the dataset was initially highly imbalanced. They addressed this by resampling to create 250000 samples per rating level, which resulted in more balanced model performance. These findings indicate that future work could focus on collecting more data, especially neutral reviews, so that balanced sampling strategies such as downsampling can be applied without significant loss of information.

For the sentence embedding models, SetFit delivered surprisingly strong performance. Its macro F1 of 0.6644 was close to that of MPNet with LightGBM (0.7043), despite using only close to 100 labeled samples per class. This demonstrates that SetFit is an efficient and effective alternative when only limited labeled data are available.

Table 3: Model Comparison

	Macro F1	Accuracy
Naive Bayes	0.5979	0.7571
SVM	0.6188	0.8265
BERT	0.7194	0.8531
RoBERTa	**0.7321**	**0.8830**
MPNet + LightGBM	0.7043	0.8665
SetFit	0.6644	0.7940

Table 4: Per Class Performance of Naive Bayes Model

	Precision	Recall	F1 Score
Negative	0.5774	0.5917	0.5845
Neutral	0.3688	0.3356	0.3514
Positive	0.9156	0.9256	0.9206
Macro Avg	0.8214	0.6176	0.6188

Table 5: Per Class Performance of SVM Model

	Precision	Recall	F1 Score
Negative	0.4509	0.7283	0.5570
Neutral	0.2925	0.4814	0.3639
Positive	0.9579	0.8018	0.8729
Macro Avg	0.9579	0.6705	0.5979

Table 6: Per Class Performance of the Final RoBERTa Model

	Precision	Recall	F1 Score
Negative	0.6951	0.7421	0.7178
Neutral	0.8822	0.5078	0.525

Positive	0.9521	0.9546	0.9533
Macro Avg	0.7302	0.7348	0.7321

5. Summary

Through my experiments, I found that class imbalance had a substantial impact on model performance, particularly for the neutral class. Even after applying multiple imbalance handling strategies, achieving balanced performance remained challenging. The neutral class consistently underperformed, likely due to both its smaller sample size and its more ambiguous semantic characteristics.

A noteworthy observation from my work is the strong performance of SetFit. Despite using only a small number of labeled samples, SetFit achieved results that were close to those of a fully fine tuned sentence transformer with a downstream classifier. Its computational efficiency also makes it a practical choice when resources or labeled data are limited.

For future work, collecting additional neutral reviews would allow for more balanced training and more reliable evaluation. Further improvements may be possible through advanced imbalance handling techniques such as focal loss, resampling strategies, or domain-adaptive pretraining. These directions could help address the persistent difficulty of modeling the neutral sentiment class.

Code found on Internet: Approximately 40-50% of my code was found on the internet.

7. Reference

Agni Siddhanta, & Bhagat, A. K. (2025). Sentiment Showdown - Sentence Transformers stand their ground against Language Models: Case of Sentiment Classification using Sentence Embeddings. *Procedia Computer Science*, *257*, 1205–1212. https://doi.org/10.1016/j.procs.2025.03.161

Arush Chillar. (2021). Disneyland Reviews. Retrieved December 4, 2025, from Kaggle.com website: https://www.kaggle.com/datasets/arushchillar/disneyland-reviews/data

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Retrieved from https://arxiv.org/pdf/1810.04805

huggingface. (2025, August 5). GitHub - huggingface/setfit: Efficient few-shot learning with Sentence Transformers. Retrieved December 5, 2025, from GitHub website: https://github.com/huggingface/setfit/

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … Allen, P. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. Retrieved from https://arxiv.org/pdf/1907.11692

Liu, Z. (2020). Yelp Review Rating Prediction: Machine Learning and Deep Learning Models. https://doi.org/10.48550/arXiv.2012.06690

Pretrained Models — Sentence Transformers documentation. (n.d.). Retrieved from sbert.net website: https://sbert.net/docs/sentence_transformer/pretrained_models.html

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2015.7298682

Song, K., Tan, X., Qin, T., Lu, J., & Liu, T.-Y. (2020). MPNet: Masked and Permuted Pre-training for Language Understanding. https://doi.org/10.48550/arXiv.2004.09297

Theme Park Market Size And Share | Industry Report, 2033. (2024). Retrieved from Grandviewresearch.com website: https://www.grandviewresearch.com/industry-analysis/theme-park-market-report

Tunstall, L., Reimers, N., Seo, U., Bates, L., Korat, D., Wasserblat, M., … Ai, C. (2022). *Efficient Few-Shot Learning Without Prompts*. Retrieved from https://arxiv.org/pdf/2209.11055

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017, June 12). Attention Is All You Need. Retrieved from Cornell University website: https://arxiv.org/abs/1706.03762