Sentiment Analysis on Disneyland Reviews

NLP Final Group Report

Name: Annie Cheng, Junhua Deng

Course Name: DATS6312 Natural Language Processing

Professor: Dr. Ning Rui

Due Date: Dec 5, 2025

1. Introduction

With the increasing volume of feedback from online platforms, extracting insights from the vast amount of reviews has become a crucial topic for companies. One of the most valuable aspects of these reviews is the sentiment behind them. Understanding customer sentiment can help businesses improve satisfaction, tailor services to customer needs, and ultimately increase profitability.

The tourism and hospitality industry is a major part of the US economy, with the amusement and theme park sector projected to generate USD 24.6 billion in revenue in 2025. In this industry, customer decisions are strongly influenced by the experiences shared by others. Platforms like Tripadvisor provide a space where visitors share detailed feedback that future customers rely on.

This research aims to analyze Disneyland customer reviews and build a tool that can extract useful sentiment insights from large collections of text. The insights from this study can potentially impact business strategies and improve profitability.

2. Data Description

The dataset used in this study consists of TripAdvisor reviews from the Disneyland branches in Paris, California, and Hong Kong, obtained from Kaggle (Arush Chillar, 2021). It includes 42,632 entries with six features: a unique identifier, rating, visit date, visitor origin country, review comments, and branch. Ratings are based on a 1 to 5 star scale.

3. Exploratory Data Analysis

To gain an initial understanding of the data, we conducted an exploratory analysis. Since sentiment is our target variable, we 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 illustrates the distribution of 3 branches: Paris, California, and Hong Kong. Among the three locations, Disneyland California received the highest number of reviews, approaching 19000. Disneyland Paris follows with roughly 14000 reviews. Disneyland Hong Kong has the smallest number of reviews, at around 9500.

Fig. 4 shows how average customer ratings evolved over time for the three Disneyland branches. Disneyland California consistently maintains the highest and most stable ratings, generally staying above 4.3. Hong Kong also performs well with ratings mostly between 4.0 and 4.3, showing moderate fluctuations. Paris exhibits the lowest and most variable ratings, especially in the early years, with several dips below 4.0. Overall, California leads in customer satisfaction, Hong Kong remains steady, and Paris shows more variability and lower overall sentiment.

Fig. 5 and Fig. 6 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. 7 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. 9, neutral reviews contain overlapping terms such as “ride”, “good”, “time”, and “queue”.

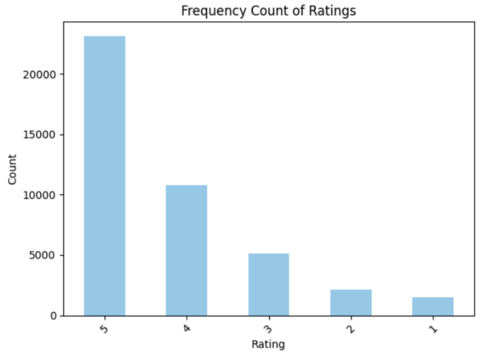


Figure 1: Distribution of Ratings

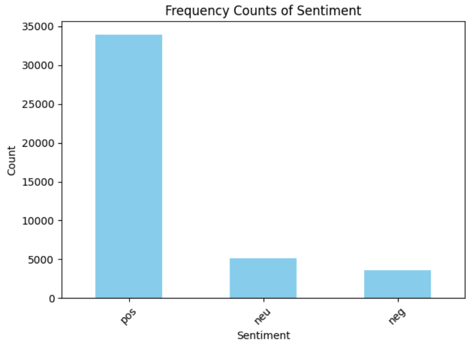


Figure 2: Distribution of Sentiments

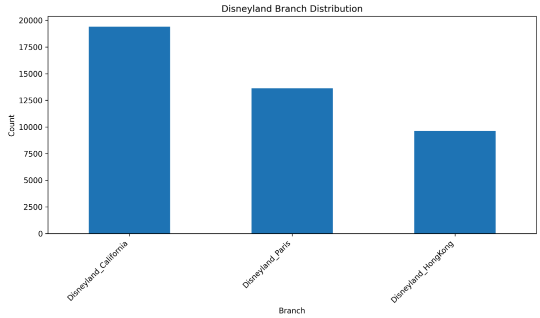


Figure 3: Distribution of Branch

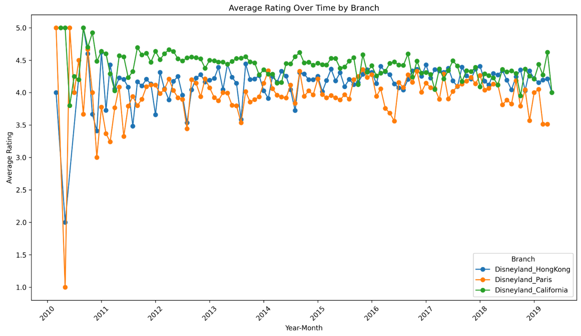


Figure 4: Average Rating Over Time by Branch

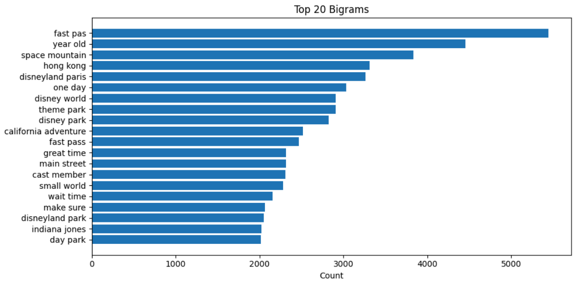


Figure 5: Top 20 Bigrams

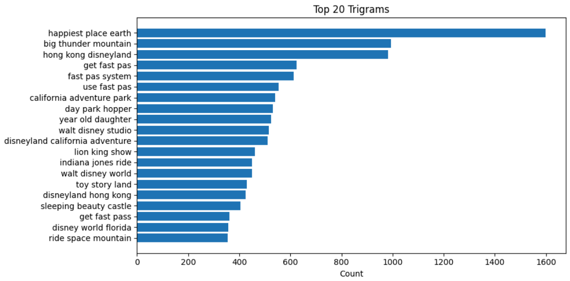
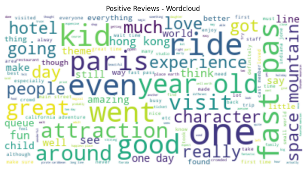
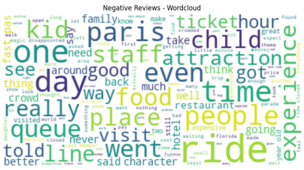


Figure 6: Top 20 Trigrams





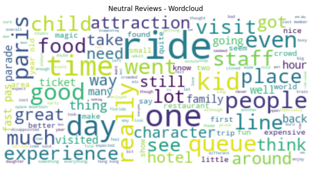


Figure 7: Word Cloud (Positive) Figure 8: Word Cloud (Negative) Figure 9: Word Cloud (Neutral)

4. Model Selection

4.1. Classical Machine Learning Models

Logistic Regression

Logistic regression serves as a simple and strong baseline for classification models. It is a linear classification model that estimates the probability of an input belonging to each class. In the multiclass setting, which applies to our three sentiment categories, logistic regression uses the softmax function to convert linear scores into class probabilities. Given an input vector x, the predicted probability for class k is:

$$P(y = k \mid x) = \frac{\exp(w_k^\top x + b_k)}{\sum_{j=1}^{K} \exp(w_j^\top x + b_j)} \tag{1}$$

The model assigns the label with the highest predicted probability:

$$\hat{y} = \arg\max_k \, P(y = k \mid x) \tag{2}$$

Naive Bayes

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:

$$\hat{y} = \arg\max_{y_j} \; P(Y = y_j) \prod_i P(X_i \mid Y = y_j) \tag{3}$$

Support Vector Machine

Support Vector Machines (SVMs) are widely used in text classification tasks due to their strong performance in high-dimensional and sparse feature spaces characteristic of natural language data. When combined with representations such as TF–IDF or bag-of-words, SVMs learn a discriminative hyperplane that maximizes the margin between document classes, enabling effective generalization even when the feature dimension far exceeds the number of training samples. Linear and kernel-based variants allow SVMs to model both linearly and nonlinearly separable patterns, and empirical results show that linear SVMs often serve as competitive baselines for sentiment analysis, topic categorization, and other NLP tasks, particularly when data availability is limited. The objective of a linear SVM is to learn a maximum-margin classifier by solving the following optimization problem:

$$[\min_{\mathbf{w}, b, \boldsymbol{\xi}} \; \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i \quad \text{subject to} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \;\; \xi_i \geq 0.] \tag{4}$$

4.2. Transformer-Based Models

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.

DistilBERT

DistilBERT is a compact version of the original BERT model, created through a technique called knowledge distillation. By training a smaller network to mimic the behavior of BERT-base, it achieves a strong balance between efficiency and performance. Despite having 40% fewer parameters, it retains around 97% of BERT’s accuracy on major natural language understanding benchmarks.

Because of its lighter design, DistilBERT is much faster during inference and requires significantly less memory, making it ideal for real-time or resource-constrained applications. It is widely used for tasks such as sentiment analysis, text classification, and question answering, offering a practical alternative when full-size transformer models are too costly to deploy.

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.

DeBERTa

DeBERTa (Decoding-enhanced BERT with Disentangled Attention) is an advanced transformer architecture designed to improve contextual representation learning beyond BERT and RoBERTa. Its primary innovation lies in disentangled attention, which independently encodes content and positional information, allowing the model to compute attention distributions with greater precision. Additionally, DeBERTa incorporates an enhanced mask decoder, which refines absolute position modeling during pretraining and leads to improved generalization across downstream NLP tasks. DeBERTa is a widely adopted architecture in research and industry applications where high accuracy and fine-grained language understanding are essential.*(He et al., 2021)*

4.3 Sentence-Embedding Models

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.

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. Experimental Setup

5.1 Classical Machine Learning Setup

For the classical models (Logistic Regression, Naive Bayes, and SVM), we first applied standard text preprocessing: lowercasing, removing URLs and HTML tags, normalizing spaces, tokenizing, removing stopwords, and lemmatizing. The cleaned text was then converted into numerical features using TF-IDF. We trained the models using an 80/20 train-validation split, stratified by sentiment. Since the dataset is imbalanced, we applied class-weighted loss to mitigate bias toward the majority class. Macro F1 was used as the primary evaluation metric, supplemented with accuracy and per class metrics.

5.2 Transformer-Based Fine-Tuning Setup

For DistilBERT, BERT, RoBERTa, and DeBERTa, we 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 we implemented are “bert-based-uncased”, “distilbert-base-uncased”, “roberta-base”, and “deberta-v3-base”.

For all models, we 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. We first trained each model using the baseline hyperparameter configuration recommended in the original BERT paper (Devlin et al., 2019). We used macro F1 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	
DistilBERT	2e-5	16	3	Balanced class weighted loss
RoBERTa	3e-5	16	3	Inverse Frequency weighted loss
DeBERTa	2e-5	12	3	Balanced RandomSampler

After evaluating all models with the baseline configuration using macro F1, we narrowed our focus to RoBERTa, the best-performing model. We then conducted further experiments on this model to address class imbalance and improve performance. Specifically, we evaluated two weighting schemes, normalized inverse frequency and normalized square-root inverse frequency, and applied these to the loss function and through a WeightedRandomSampler. After selecting the best-performing weighting strategy, we performed additional hyperparameter trials by lowering the learning rate to 2e-5 and increasing the number of epochs to 5. We also added early stopping with a patience of 2 rounds.

5.3 Sentence Transformer + Classifier Setup

For the sentence embedding-based models, we 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, we 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, we used logistic regression and LightGBM as the downstream classifiers.

For SetFit, we 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

6. Results and Discussion

Among the classical machine learning models, SVM achieved the strongest performance with a macro F1 of about 0.62. From the classification report, we observed that all classical models struggled with the neutral class. This aligns with our observation about the imbalanced nature of the dataset.

Among the fully fine-tuned transformer models with baseline configuration, RoBERTa outperformed the others. After running additional experiments with different class weighting strategies, we found that the square root inverse frequency weighted loss gives us the best performance. However, the model still struggles to achieve a balanced performance.

Although class imbalance remained a challenge, we also examined whether the baseline hyperparameters were near optimal. We lowered the learning rate to 2e-5, increased the number of epochs to 5, and added early stopping with a patience of 2 rounds to reduce overfitting. This configuration did not surpass the baseline setting. We also observed that with learning rate 2e-5 and five epochs, the best performance was reached at epoch 2. In later epochs, the performance fluctuated slightly, rising again at epoch 4 before declining, but these oscillations did not lead to any meaningful improvement. This suggests that the model was not gaining new generalizable signals after the early epochs, especially given the difficulty posed by the imbalanced and 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. SetFit reached a macro F1 that was very close to MPNet with LightGBM, at about 0.6644 and 0.7043 respectively. With fewer than one hundred training samples, SetFit achieved performance close to that of a fully fine tuned sentence transformer combined with a downstream classifier. This result demonstrates that SetFit is an effective and efficient approach, especially when only limited labeled data are available.

Table 3: Model Comparison

	Macro F1	Accuracy
Logistic Regression	0.5563	0.7130
Naive Bayes	0.5979	0.7571
SVM	0.6188	0.8265

DistilBERT	0.7071	0.8531
BERT	0.7194	0.8531
RoBERTa	**0.7321**	**0.8830**
DeBERTa	0.7277	0.8818
MPNet + LightGBM	0.7043	0.8665
SetFit	0.6644	0.7940

Table 4: 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

7. Summary

Overall, all model types performed well on the positive class but struggled with neutral and negative reviews, with neutral recall being particularly challenging. Among the three model families, pretrained transformer models achieved the strongest and most consistent results, outperforming sentence-embedding approaches. Classical machine learning models showed the weakest performance, indicating their limited ability to capture nuanced sentiment cues in this dataset.

The majority of online feedback datasets are inherently imbalanced, and this strongly affects model performance. In our project, even after applying multiple strategies to address class imbalance, achieving balanced performance remained challenging. We suspect that the neutral class consistently underperforms not only because it has fewer examples, but also because its semantic boundaries are more ambiguous.

Another notable finding is the strong capability of the few-shot learning method SetFit. Despite using only a small number of labeled samples, it achieved competitive results compared to a sentence transformer fine-tuned on the full training dataset. In addition, SetFit is a more efficient and computationally less intensive alternative, making it particularly useful with limited resources or limited labeled data.

Future work can focus on expanding the dataset by incorporating reviews from additional platforms to improve domain coverage and reduce sampling bias. Increasing the number of neutral and underrepresented branches, along with adding metadata such as visit time or seasonal events, would support more robust and context-aware analysis.

On the modeling side, future research may explore stronger imbalance-handling methods such as focal loss, dynamic re-weighting, or contextual data augmentation. More advanced transformer architectures and domain-adaptive pretraining also represent promising directions, and multi-task models could jointly learn sentiment and aspect information for richer insights.

8. References (APA format)

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

He, P., Liu, X., Gao, J., & Chen, W. (2020). DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *International Conference on Learning Representations*. https://doi.org/10.48550/arxiv.2006.03654

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). *DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter*. arXiv:1910.01108. https://doi.org/10.48550/arXiv.1910.01108

Wang, Y., Weissweiler, L., Schütze, H., & Plank, B. (2023). *How to distill your BERT: An empirical study on the impact of weight initialization and distillation objectives*. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 1800–1813). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-short.157

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