December 3, 2025

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Sentiment Analysis on Disneyland Reviews

Team Members:
Annie Cheng
Junhua Deng

Introduction

Online platforms contain a large amount of customer feedback, and companies need to understand the sentiment behind these reviews to improve their services and decisions.

In this project, we aim to analyze Disneyland customer reviews and build a tool that can extract useful sentiment insights from large collections of text.

Project Overview

1 Data Collection

Extract Disneyland reviews data from Kaggle

2 Text Preprocessing

Basic cleaning, tokenization, stop word removal, lemmatization

3 - Model Development

Classical ML models and transformer-based models

4. Evaluate

Evaluate performance metrics and address class imbalance

5. Insights and future work

key findings and future improvment

Dataset Description

The dataset contains 42,000 TripAdvisor reviews from three Disneyland locations: Paris, California, and Hong Kong.

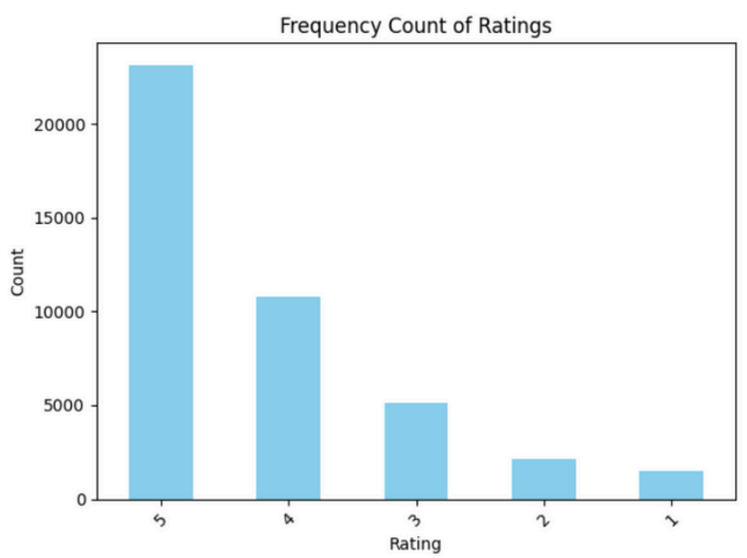
Target Variable:

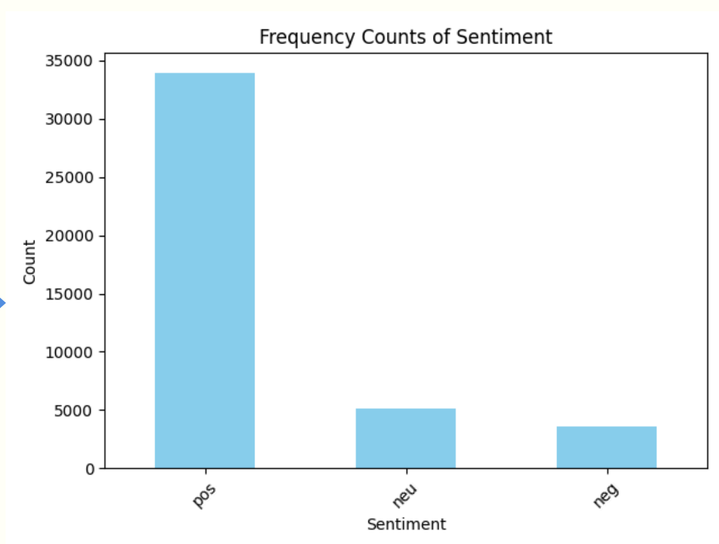
5 star ratings mapped to sentiment

- 1 star, 2 stars → negative
- 3 stars → neutral
- 4 stars, 5 stars → positive

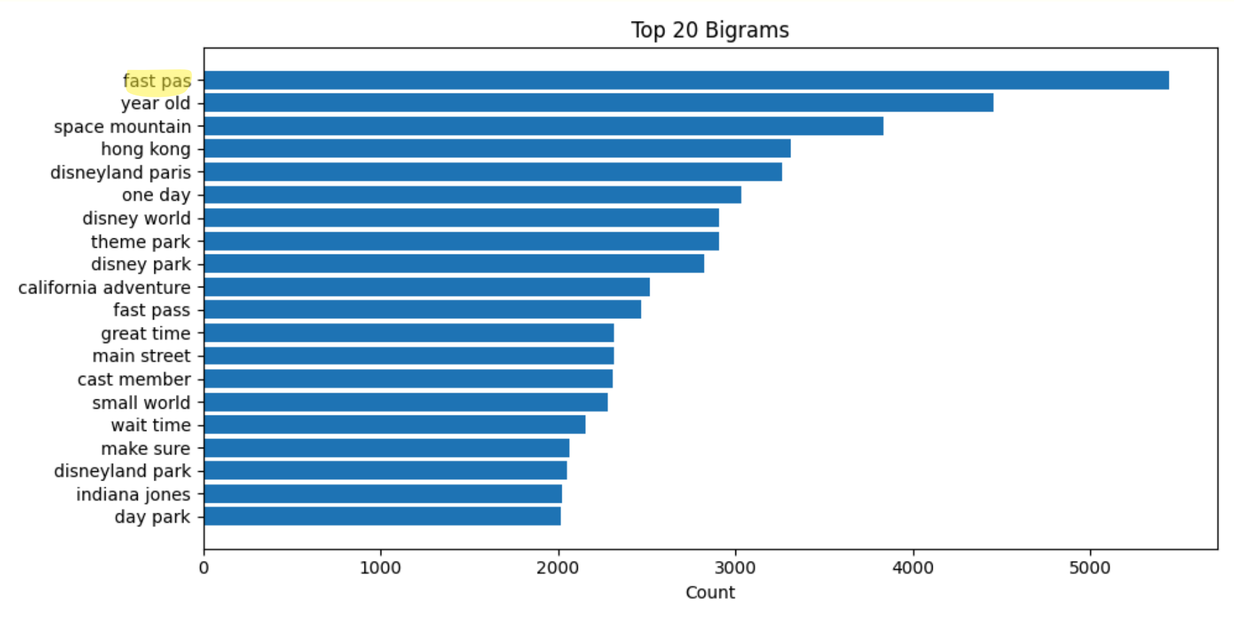
Columns used in this project:

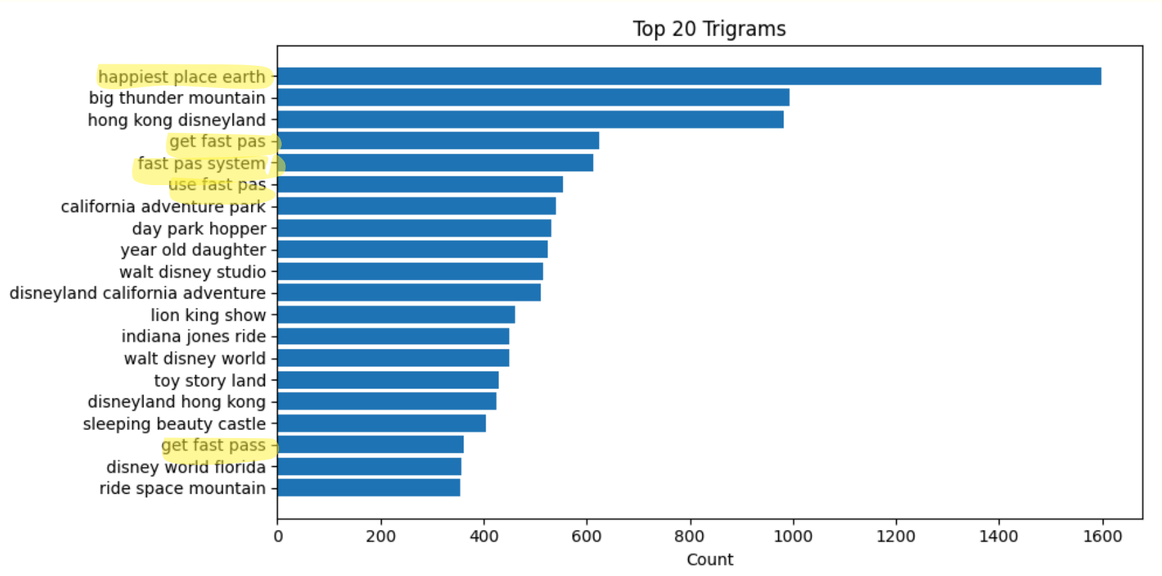
1. Rating — 1 to 5
2. Review_Text — written review
3. Disneyland_Branch — park location





Top 20 Bigrams and Trigrams





Word Clouds by Sentiment

Positive

Negative

Neutral

Classical ML Model

Text Preprocessing

Basic Cleaning

Lowercase text, remove URLs and HTML tags, keep alphabetic characters only, and normalize whitespace.

Tokenize text using word_tokenize from NLTK

Tokenization

Stopword Removal

Remove stopwords from nltk.corpus

Lemmatize tokens using NLTK’s WordNetLemmatizer

Lemmatization

Classical ML

Evaluate

Assess model performance

Address Class Imbalance

Apply class-weighted loss to address imbalance

Classification Algorithm

Train logistic regression, Naive Bayes, and SVM

Word Embedding

Use TF IDF to convert text into numerical feature vectors

	ACCURACY	MACRO PRECISION	MACRO RECALL	MACRO F1
Logistic Regression	0.7130	0.5366	0.6227	0.5563
Naive Bayes	0.7571	0.5671	0.6705	0.5979
SVM	**0.8265**	**0.6206**	**0.6174**	**0.6188**

$$TF\text{-}IDF(t, d, D) = TF(t, d) \times IDF(t, D)$$

Classification Report

Classification Report:

	precision	recall	f1-score	support
0	0.4509	0.7283	0.5570	725
1	0.2925	0.4814	0.3639	1022
2	0.9579	0.8018	0.8729	6785
accuracy			0 7571	8532
macro avg	0.5671	0.6705	0.5979	8532
weighted avg	0.8351	0.7571	0.7851	8532

SVM Classification Report

Classification Report:

	precision	recall	f1-score	support
0	0.5774	0.5917	0.5845	725
1	0.3688	0.3356	0.3514	1022
2	0.9156	0.9256	0.9206	6785
accuracy			0.8265	8532
macro avg	0.6206	0.6176	0.6188	8532
weighted avg	0.8214	0.8265	0.8238	8532

Naive Bayes Classification Report

Transfromer-Based Model

Experimental Setup

01

Data Preparation

- Input Text = Branch + Review Text
- 80% Training 20% validation, stratifyied by sentiment class

02

Model Implementation

- Transformer Encoder Model: BERT, RoBERTa, DistillBERT, DeBERTa
- Sentence-Embedding model: MPNet SetFit

03

Training

- Hugging Face Transformers (fine tuning)
- SentenceTransformer embeddings plus classifier (MPNet, SetFit)

04

Class Imbalance Handling

- Class weights
- WeightedRandomSampler

05

Evaluation Metrics

- Macro f1 as primary metric
- Accuracy, and metrics per class

Tranformer-Based Model

Fully Fine Tuned Transformer Models

- Bert-based-uncased
- Distilbert-base-uncased
- roberta-base
- Deberta-v3-base

Embedding Based Models

- all-mpnet-base-v2 (Encoder + Classifier)
- SetFit (contrastive tuning + Logistic Regression)

Full Fine Tuned Transformer

Embedding Based Model

Hyperparameter Summary

General Settings:

- Max Length: 256, Loss: Cross Entropy, Optimizer: AdamW with 0.01 decay, warm up ratio:0.1

Training Configuration

	LEARNING RATE	BATCH SIZE	EPOCHS	CLASS WEIGHTS	NOTE
BERT	3e-5	16	3	N/A	
DistillBERT	2e-5	16	3	Balanced	
RoBERTa	2e-5, 3e-5	16	3, 5	Multiple	Best with LR = 3e−5, 3 epochs, square root class weights
DeBERTa	2e-5	12	3	N/A	WeightedRandomSampler instead

RoBERTa: Imbalance Handling + Tuning

Handling Class Imbalance

1. **Class-weighted loss strategies:**
   - Inverse frequency
   - Square root inverse frequency
2. **WeightedRandomSampler**
   - Inverse frequency
   - Square root inverse frequency

Inverse Frequency

$$W_i = \frac{1}{f_i} \qquad W_i = \frac{W_i}{\text{mean}(W)}$$

Square Root Inverse Frequency

$$W_i = \frac{1}{\sqrt{f_i}} \qquad W_i = \frac{W_i}{\text{mean}(W)}$$

Hyperparameter Trials

- Baseline: LR = 3e–5, epochs = 3
- Tried: LR = 2e–5, epochs = 5
- No improvement (accuracy went up but macro F1 went down)

Final Model: LR = 3e–5, epochs = 3, square root class weighted loss

RoBERTa Classification Report

	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
Accuracy			0.8830
Macro Average	0.7302	0.7348	**0.7321**

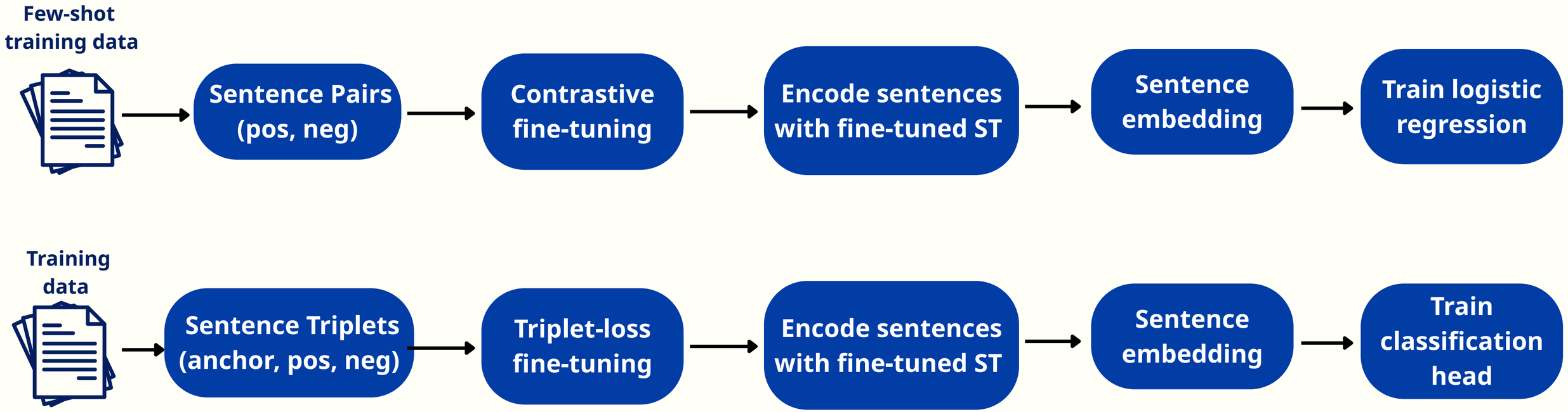
Sentence Transformer + Classifier

SetFit (Contrastive Fine-Tuning)

- Fine tuned MPNet with 32 samples/class
- Iterations: 20, Epochs: 3, Batch size: 16
- Classifier: Logistic Regression

SBERT Fine-Tuning + Classifiers

- Fine tuned MPNet with triplet loss
- Batch size: 8, LR = 3e-5, epochs: 3
- Classifier: Logistic Regression / LightGBM



Classification Report

	PRECISION	RECALL	F1 SCORE
Negative	0.6441	0.6961	0.6693
Neutral	**0.3258**	**0.6243**	**0.4282**
Positive	0.9725	0.8299	0.8956
Accuracy			0.7940
Macro Average	0.6475	0.7169	**0.6644**

SetFit Classification Report

	PRECISION	RECALL	F1 SCORE
Negative	0.6591	0.6800	0.6694
Neutral	**0.4758**	**0.5196**	**0.4967**
Positive	0.9552	0.9387	0.9460
Accuracy			0.8665
Macro Average	0.6967	0.7128	0.7043

SBERT + LightGBM Classification Report

Model Comparison

	MACRO F1	ACCURACY
Logistic Regression (Baseline)	**0.5563**	**0.7130**
Naive Bayes	0.5979	0.7571
SVM	0.6188	0.8265
DistillBERT	0.7071	0.8531
BERT	0.7194	0.8773
RoBERTa	**0.7321**	**0.8830**
DeBERTa	0.7277	0.8818
SBERT+LightGBM	0.7043	0.8665
SetFit	0.6644	0.7940

Streamlit Demo

https://disneylandreview-2q25dhcuvyjqripkydashx.streamlit.app

Test Comments

Positive Review
Hong Kong Disneyland was an amazing experience! The park is clean, beautifully designed, and full of Disney magic. The parades and nighttime show were spectacular, and the cast members were incredibly friendly and helpful. Even though the park is smaller than some others, it still offers plenty of attractions and a charming atmosphere. I would absolutely visit again.

Neutral Review
Hong Kong Disneyland was decent overall. The rides and shows were enjoyable, but the park felt a bit small compared to other Disney locations. The wait times varied, with some attractions moving quickly while others were slow. It’s a nice place for a half-day or full-day visit, especially for families, but it may not feel as impressive for frequent Disney travelers.

Negative Review
My experience at Hong Kong Disneyland was a bit disappointing. The park felt overcrowded, and many popular rides had long wait times. Several food options were expensive but not particularly good, and the overall selection was limited. While the atmosphere was nice, the experience didn’t feel worth the ticket price.

Key Takeaways

Online feedback datasets are inherently imbalanced, and this strongly affects model performance. Even after applying multiple correction strategies, achieving balanced performance remained difficult. We suspect that the neutral class underperforms not only because of imbalance, but also because its semantic boundary is ambiguous.

SetFit underperforms full transformer model; however, given the small samples it achieve a competitive performance.

Future work could focus on expanding the dataset, especially neutral reviews, and exploring more advanced imbalance-handling methods such as focal loss or domain-adaptive pretraining to improve model performance.

Thank You