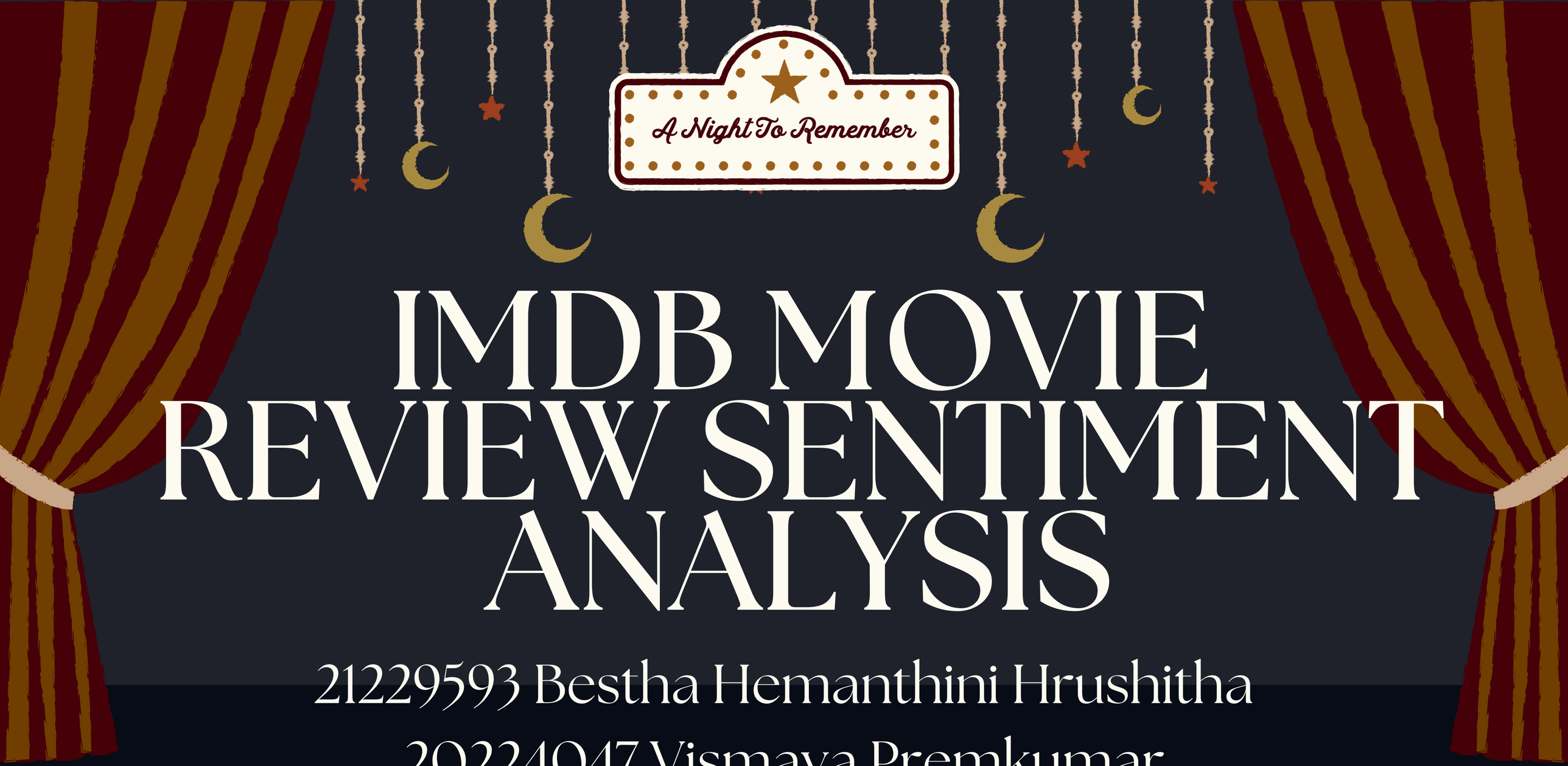


IMDB MOVIE REVIEW SENTIMENT ANALYSIS

21229593 Bestha Hemanthini Hrushitha

20224047 Vismaya Premkumar

WHAT IS HUGGING FACE?

Hugging Face is a platform and community dedicated to NLP, providing resources for building, training, and deploying machine learning models.

IMPORTANCE

1. AI for everyone: It provides readymade models making it easier for data analysis.
2. Performance that sets the bar
3. Save time and resources
4. Transparency and collaboration:
5. An evolving technology

HOW CAN PRE-TRAINED MODELS ACCELERATE DATA SCIENCE AND AI RESEARCH?

- **Faster Development:** Pre-trained models eliminate training from scratch, allowing for quick testing and iteration.
- **Improved Performance**: Leverage pre-trained models' massive dataset training for better results. Fine-tuning further boosts accuracy.
- **Reproducibility & Collaboration:** Publicly available models and documentation facilitate collaboration and knowledge sharing.
- **Cutting-edge Technology:** Access to the latest NLP and AI advancements through constant platform updates.
- **Democratization of AI:** Open-source platform makes AI accessible to a wider range of researchers.

RESEARCH QUESTION

Can the model detect the percentage of negativity and postivity in the movie review sample input?

DATA MODEL

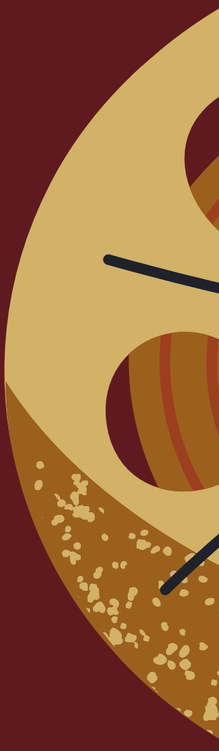
Hugging Face model used here:

JiaqiLee/imdb-finetuned-bert-base-uncased

This model performs sentiment analysis on movie reviews from IMDB to determine if they are positive or negative.

It is a fine-tuned version of the bert-based-uncased model from HuggingFace models.

- Classifying movie reviews automatically
- Analyzing trends in movie reviews
- Personalizing movie recommendations



Problem and Motivation

The model addresses the sentiment analysis for movie reviews.

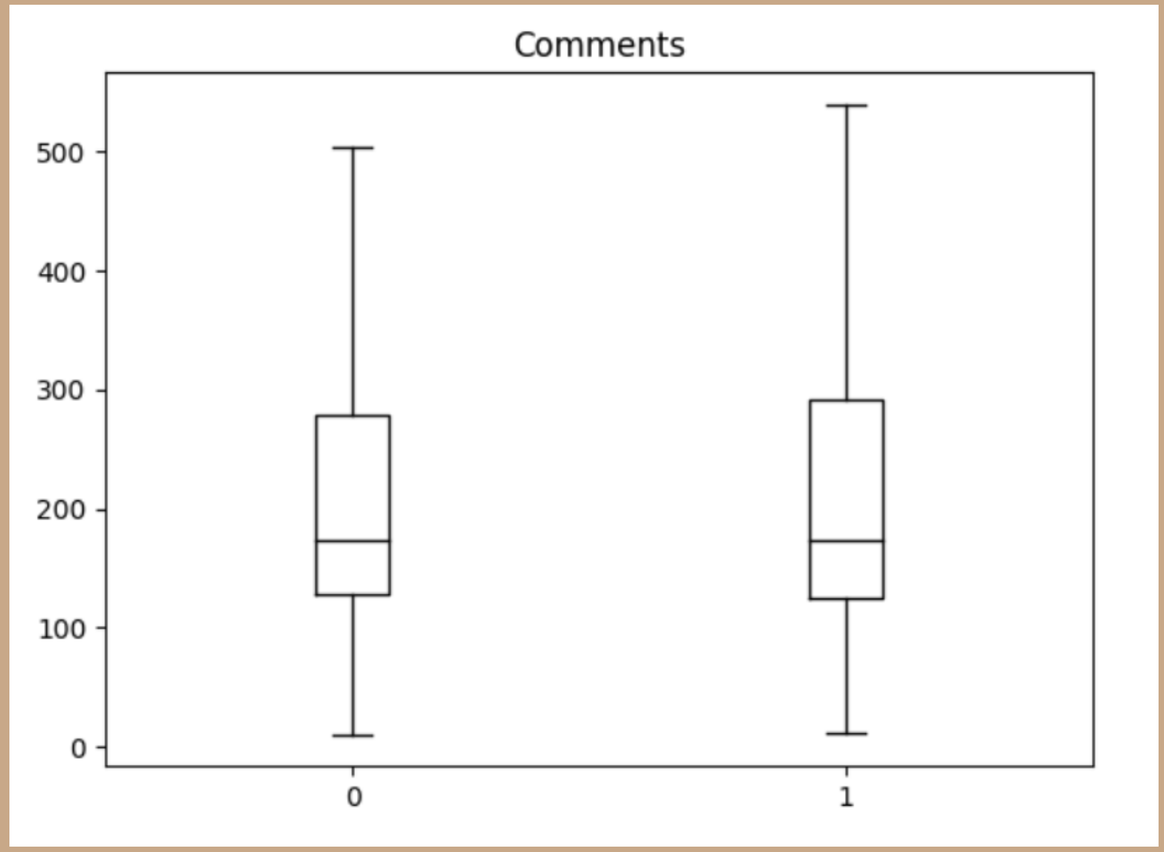
Problems:

- Bias: This model is unable to provide accurate results for customized data.
- Accuracy: The results provided for this dataset is around 88%, which cannot be considered the best accuracy results.

Motivation:

- **Understanding audience opinion through data analysis**
- **Research and development**

Data visualization



The box plot you provided shows the distribution of the number of comments for two groups. The first group - Negative has the highest number of comments, with a median of 400. The second group - Positive has the lowest number of comments, with a median of 100.

MODEL SELECTION AND TRAINING

1. **Data preparation:** The IMDB dataset is pre-processed and formatted for training. This may involve cleaning the data, tokenization, and padding.
2. **Data Pipeline:** used to provide a percentage distribution of the customized data. It also was used to find the accuracy of the model

```
preds=classifier(sample_review_two, return_all_scores = True)
preds_df = pd.DataFrame(preds[0])
labels = ["negative", "positive"]
plt.bar(labels, 100*preds_df["score"])
plt.title("positivity and negativity review sample 2")
plt.ylabel("Class probability (%)")
plt.show()
```

DATA TOKENIZTION AND EVALUATION

```
def tokenize(batch):
    return tokenizer(batch["text"], truncation=True)
# How to work tokenizer on our some data:
print(tokenize(data["train"][:2]))
data_encoded = data.map(tokenize, batched=True,
                                  batch_size=None)
```

```
{'input_ids': [[101, 1045, 12524, 1045, 2572, 8025, 1011, 3756, 20
```

TOKENIZATION

```
import evaluate
import numpy as np

accuracy = evaluate.load("accuracy")

def compute_metrics(eval_pred):
    predictions, labels = eval_pred
    predictions = np.argmax(predictions, axis=1)
    return accuracy.compute(predictions=predictions,
                            references = labels)
```

EVALUATION

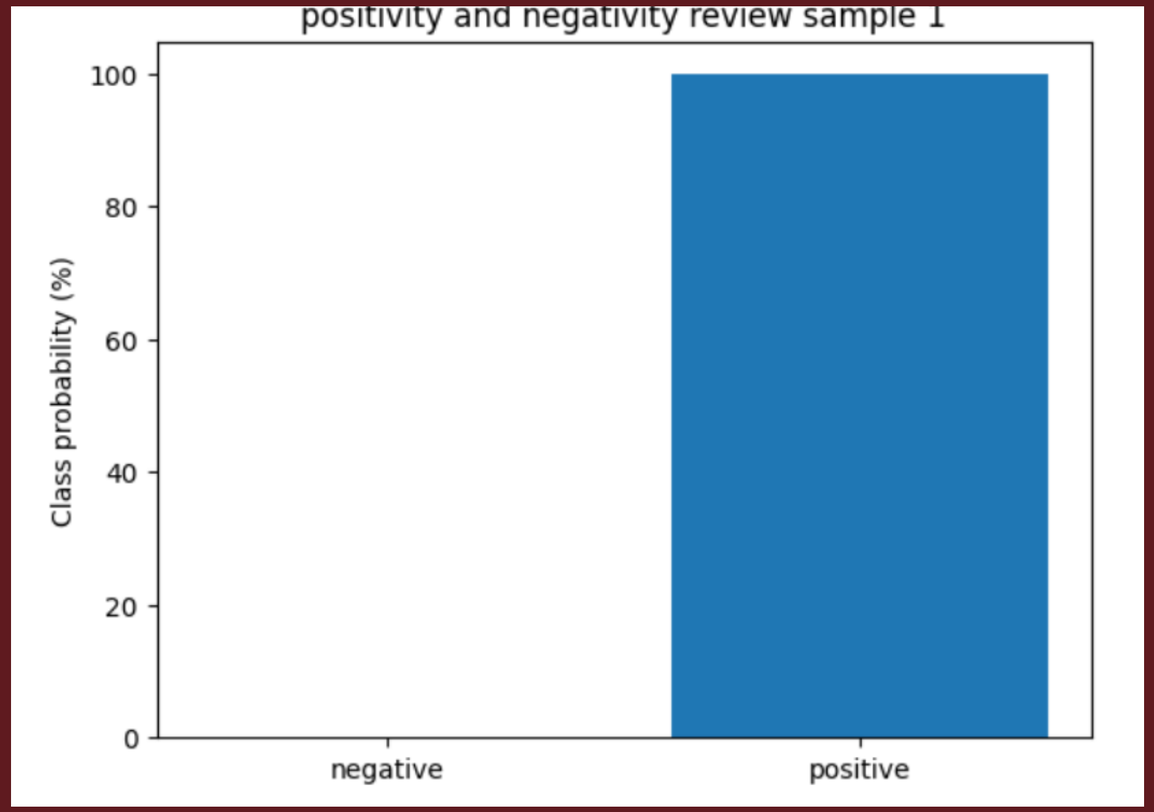
Percentage Prediction

```
from transformers import BertForSequenceClassification, BertTokenizer, TextClassificationPipeline
model_path = "JiaqiLee/imdb-finetuned-bert-base-uncased"
tokenizer = BertTokenizer.from_pretrained(model_path)
model = BertForSequenceClassification.from_pretrained(model_path, num_labels=2)
pipeline = TextClassificationPipeline(model=model, tokenizer=tokenizer)
print(pipeline(custom_text))
```

```
[{'label': 'positive', 'score': 0.8255593776702881}]
```

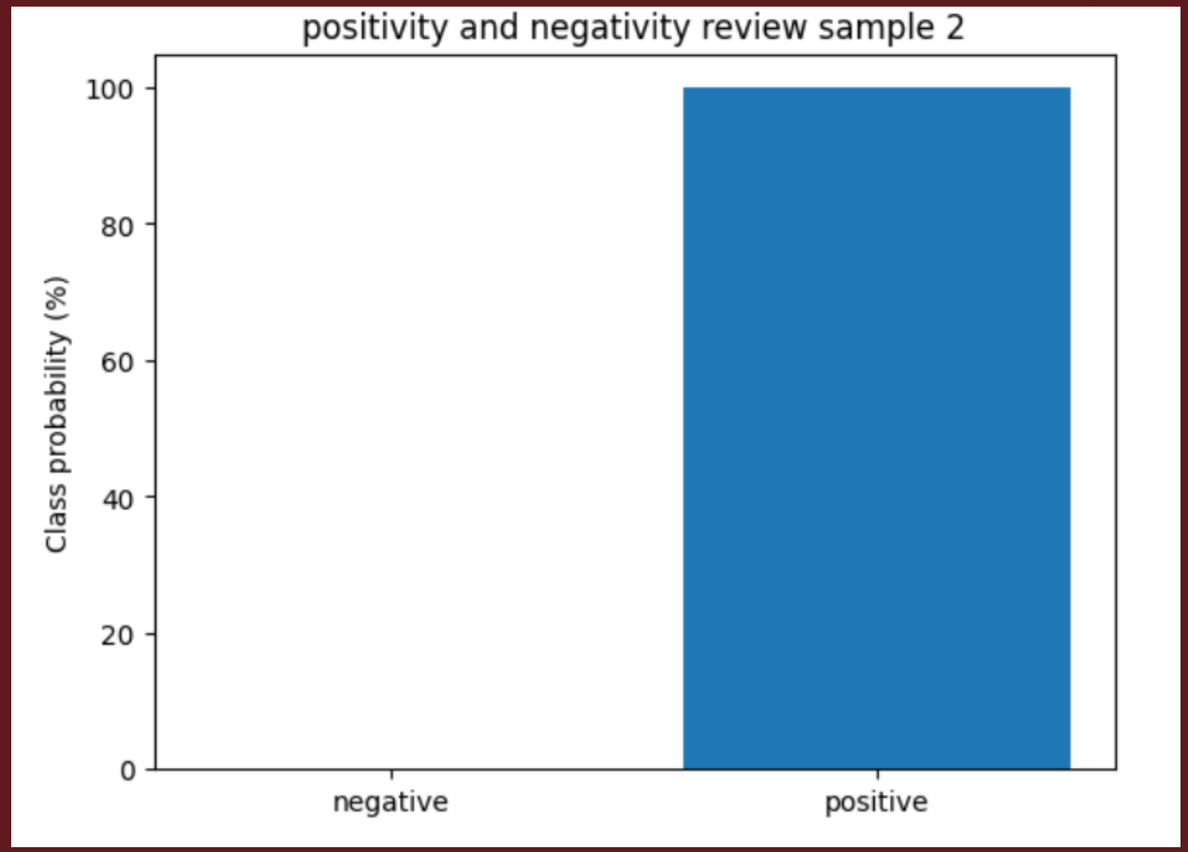
RESULTS & DISCUSSIONS

SAMPLE 1: I liked a movie I watched recently. It was really good however I felt it could have been a better screenplay to show more emotions in the movie



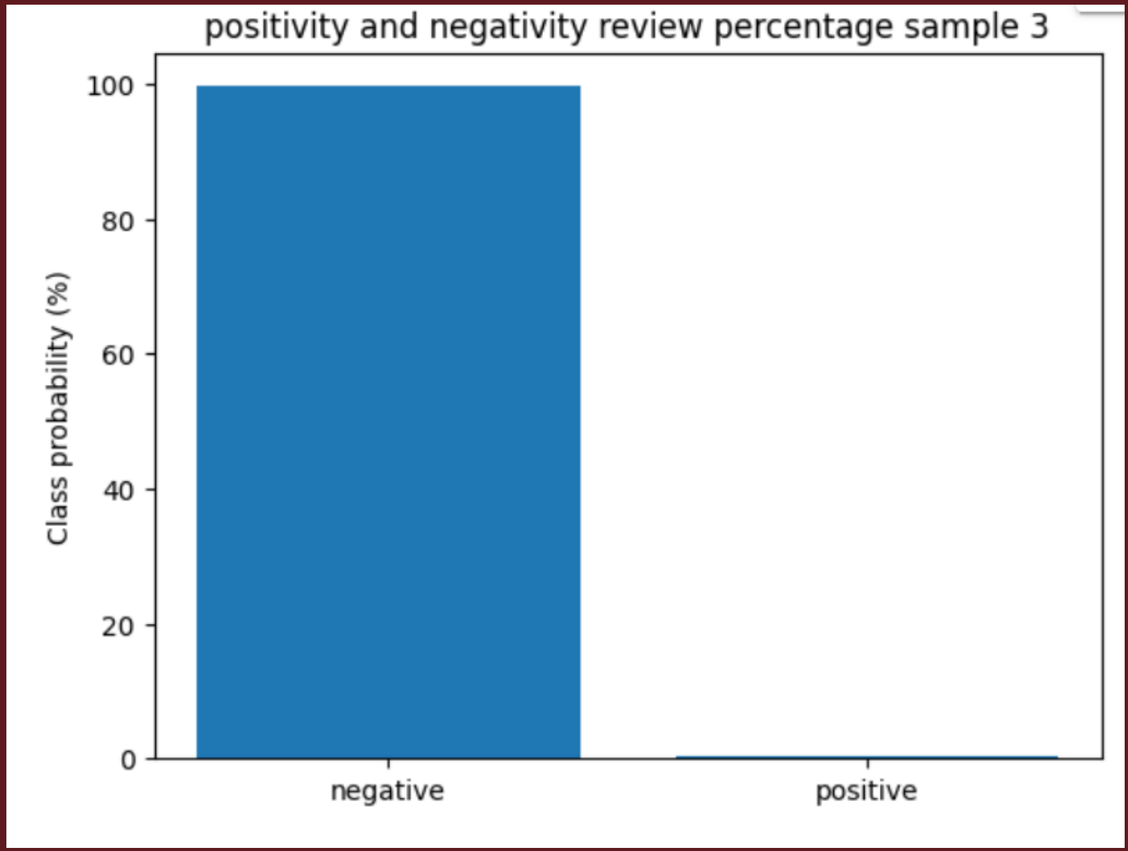
RESULTS & DISCUSSIONS

SAMPLE 2: The Menu is a film with mixed emotions. It boasts strong performances and a unique premise, but ultimately fails to reach its full potential due to a somewhat uneven narration and a rushed climax. Nevertheless, it was visually stunning and thought-provoking film that is worth watching for fans of dark comedies and thrillers."



RESULTS & DISCUSSIONS

SAMPLE 3: I had the worst experience watching this movie.however I liked the music in the movie. I liked how they tried to potray the villain as a strong character



CONCLUSION AND FUTURE WORK

The model was useful to show if a comment was either positive or negative for a movie review from the IMDB dataset. However, it could not recognize the accurate percentage of negativity and positivity in the review given. If possible in the future, We would want to improve this model to show the percentage of negativity and positivity in a movie review.

REFERENCES

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). **A pre-trained transformer-based language model for language understanding.** arXiv preprint arXiv:1810.04805. https://arxiv.org/abs/1903.09722

https://huggingface.co/huggingface-course/distilbert-base-uncased-finetuned-imdb

THANK YOU