NAME:	VICTOR AGBELEYE
STUDENT ID:	22079387
COURSE NAME:	RESEARCH METHODS
COURSE CODE:	7PAM2015-0901-2025
COLAB LINK:	**https://colab.research.google.com/drive/1MpD3A8pjxHjhJACN3ohofaEbbxt-ctDG?usp=sharing**
GITHUB LINK:	**https://github.com/AgVicCodes/nlp_analysing_imd_review_using_bert_model**

TITLE: ANALYSING IMDB REVIEW USING BERT MODEL

1. INTRODUCTION

PROBLEM STATEMENT

Sentiment analysis is a crucial aspect of understanding people and consumers of a particular product, including those on social media, e-commerce platforms, or other forms of media such as games, TV shows, and movies. This work attempts classification of movie reviews using fine-tuned BERT (Bidirectional Encoder Representations from Transformers). I will then compare the performance to other traditional models to show the advantage of context-based language models.

DATASET (EXPLORATORY DATA ANALYSIS)

The IMDB dataset contains 100,000 movie reviews, with 25% being training and testing, while 50% were classified as unsupervised.

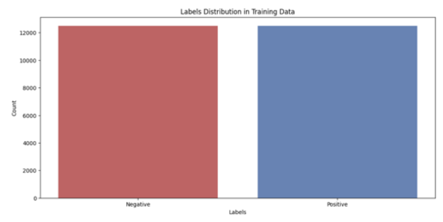


Figure 1 Label Distribution (Negative and Positive) in Train Dataset.

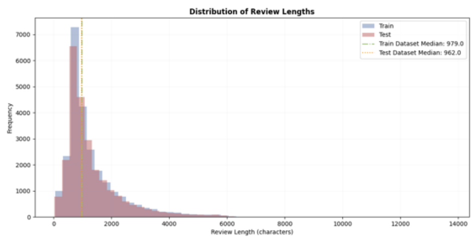


Figure 2 Distribution of lengths of review.

The text is in plain English and has a binary label classification (0 - Negative, 1 - Positive). The dataset is balanced, with 50% positive and 50% negative reviews, eliminating concerns about class imbalance.

MOTIVATION

Although traditional machine learning models, such as Logistic Regression and Naïve Bayes, ignore contextual significance and word order of text, treating it as a mere collection of words. In the case of the BERT model, the transformer architecture enables the capture of context from both directions (bi-directional) (Devlin et al, 2018) because of its self-attention mechanisms, further inferring a higher understanding of language patterns compared to the other models.

2. METHODOLOGY

MODEL SELECTION

The model I selected for this project is the BERT-base-uncased model (Geetha et al, 2021). It has 12 transformer layers, 768 hidden dimensions and 110 million parameters. The “uncased” variant makes all letters lowercase, thereby eliminating duplication. It also processes text in both directions (left to right and vice versa) because of its bidirectional attention mechanism, unlike other language models that process text from left to right by default.

ARCHITECTURE JUSTIFICATION

I chode this model for its balance of performance and computational efficiency. Other variants such as BERT-large (with more parameters – approximately 340 million) (Devlin et al, 2018), RoBERTa (which, although yielding higher accuracy, takes longer training time) (Timoneda et al, 2025) and DistilBERT (which takes less time to train but sacrifices accuracy) (Amandeep et al. 2025) either take longer time to train, are too high in performance or low in accuracy, making BERT-base-uncased the ideal model for this work.

3. PREPROCESSING STEPS

TOKENIZATION

Using the BertTokenizer from transformers, I was able to encode and decode text samples, converting them to tokens (numbers that correspond to different words from the BERT vocabulary). This process is then included in the class for data preparation before model training.

SEQUENCE HANDLING AND BATCHING

BERT can accept input tokens of up to 512 in length. Any reviews exceeding this length are truncated. Data is also batched in sizes of up to 8 per batch for maintaining GPU constraints, as with the use of a higher batch size, I encountered a series of RAM usage issues on Colab. The batches are also shuffled to avoid overfitting.

CONFIGURATION

The Hyperparameters used are:

- Learning rate: Initially 0.001 (1e-3), standard for deep learning.
- Batch size: 8 (To avoid GPU constraints)
- Epoch: 3 (Standard epoch size as 1- 2 might be too small, not enough training and anything above 3 might lead to overfitting)
- Optimizer: AdamW (With a weight decay rate of 0.01)

4. MODELLING

BASELINE MODELS

Logistic regression and Naïve Bayes, with both models sharing the same train/test split as the BERT model and training in under 5 minutes each.

EVALUATION METRICS

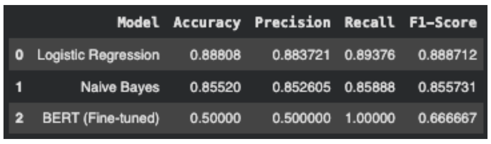
Performance was assessed using 4 metrics:

- Accuracy,
- Precision
- Recall
- F1-Score

5. RESULTS

MODEL PERFORMANCE

The table below presents the results across all models



	Model	Accuracy	Precision	Recall	F1-Score
0	Logistic Regression	0.88808	0.883721	0.89376	0.888712
1	Naive Bayes	0.85520	0.852605	0.85888	0.855731
2	BERT (Fine-tuned)	0.50000	0.500000	1.00000	0.666667

Figure 3 Model Comparison Table.

Initially, BERT achieved a 50% accuracy, underperforming both Logistic Regression and Naïve Bayes by -43.7% and -41.53%, respectively. This was a result of the learning rate used (0.001), which was too low for proper model training. The model loss during training kept fluctuating around 1.0, which was too high and indicated the model wasn't learning.

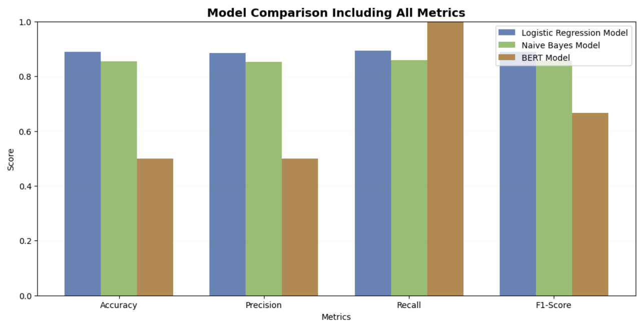


Figure 4 Model comparison before hyperparameter tuning.

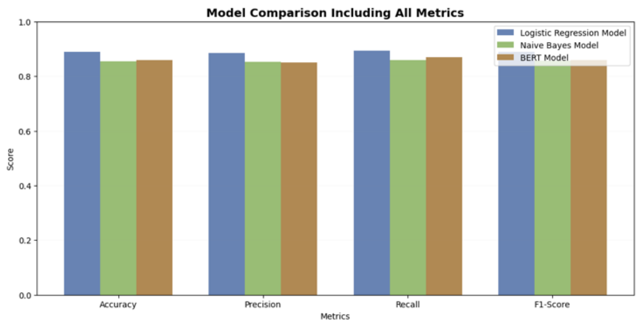


Figure 5 Model comparison bar plot after hyperparameter tuning.

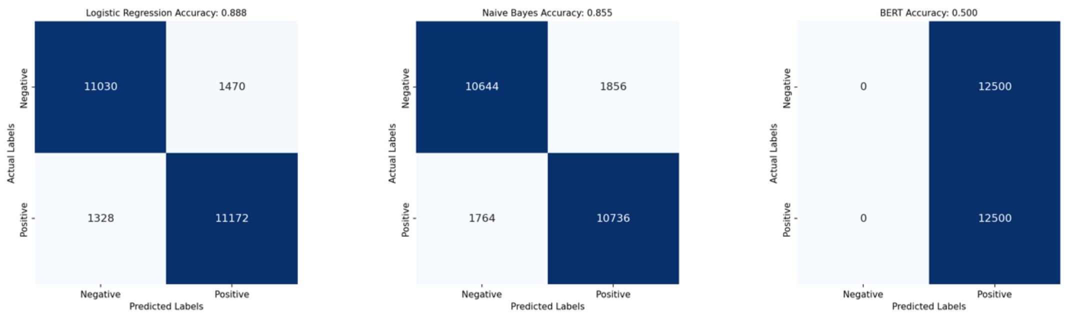


Figure 6 Confusion matrix before hyperparameter tuning.

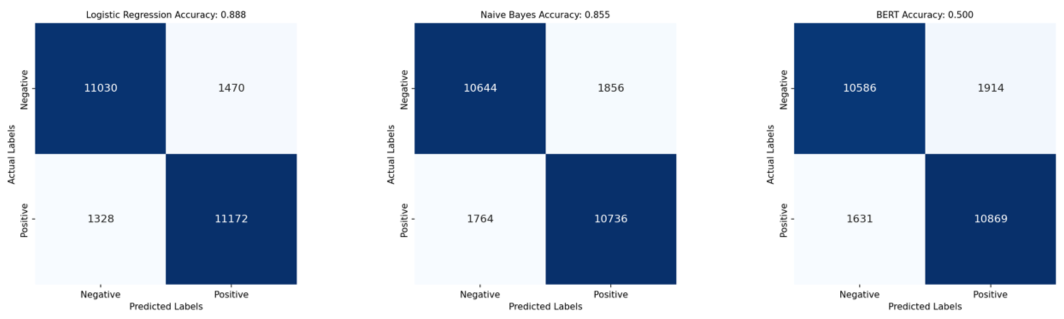


Figure 7 Confusion matrix after hyperparameter tuning.

Logistics Regression, on the other hand, with a 89% accuracy, performed a lot better, correctly classifying 22202 reviews. Although there were error rates on sarcastic reviews and complex sentiments, displaying the limitations of multiple complex datasets of sentiments.

Naïve Bayes had an accuracy of 85.8%, correctly classifying 21380 reviews, also underperforming Logistic Regression.

HYPERPARAMETER TUNING

After adjusting the learning rate to 2e-5 (0.00002 to help improve training), the BERT model had an improved accuracy score of 86%. Although better than Naïve Bayes, the model still underperformed Logistics Regression, which, compared to the standard performance of about 92% accuracy, isn't very good. This happened because I didn't train the model from scratch. After some time training the baseline model, I terminated the process and started the finetuned version, basically overwriting the previous training weights and parameters.

LIMITATIONS

- Time: While the other models took less than 5 minutes, the BERT model took about 2 hours and 45 minutes of training, which meant I couldn't train the BERT model more than once per Colab session. This was one of the factors affecting the accuracy.
- Sarcasm and Complex Sentences: BERT fails on sarcastic reviews because of a lack of tone detection, which is a weakness in most transformer models.
- Pre-training Domain Compatibility: BERT being pretrained on Wikipedia might not align with the colloquial informal movie review language and tone. A version of BERT being pre-trained on the movie review domain might be better.

6. CONCLUSION

This project successfully fine-tuned BERT for binary-based movie review classification, achieving a 86% accuracy, which is 3.36% worse than Logistic Regression but 0.35% better than Naïve Bayes. A few findings from this work include BERT's performance competitiveness and the challenges facing most models, including tone detection and sarcasm. Future work can include domain-based pretraining (on a movie review dataset), as this should significantly improve the model's performance.

7. REFERENCE

- Amandeep and Suresh, S. (2025) 'Transforming Fake News Detection: Leveraging DistilBERT Models for Enhanced Accuracy', Procedia computer science, 260, pp. 283–290.
- Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018) 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'.
- Geetha, M.P. and Karthika Renuka, D. (2021) 'Improving the performance of aspect based sentiment analysis using fine-tuned Bert Base Uncased model', International journal of intelligent networks, 2, pp. 64–69.
- Timoneda, J.C. and Vera, S.V. (2025) 'BERT, RoBERTa, or DeBERTa? Comparing Performance Across Transformers Models in Political Science Text', The Journal of politics, 87(1), pp. 347–364.