Final Report: Sentiment Analysis on IMDb Movie Reviews

# 1. Objective

The primary objective of this project was to build a machine learning classification model to predict the sentiment (positive/negative) of IMDb movie reviews using natural language processing techniques.

# 2. Dataset Overview

- Dataset Size: 50,000 movie reviews  
- Attributes:  
 - review: Text of the movie review  
 - sentiment: Sentiment label (positive / negative)

# 3. Exploratory Data Analysis (EDA)

- Missing Values: None  
- Class Distribution:  
 - Positive: ~25,000 reviews  
 - Negative: ~25,000 reviews (Balanced dataset)  
- Review Length Analysis:  
 - Varying review lengths (most between 100–500 words)  
 - Outliers in both very short and very long reviews

# 4. Data Cleaning & Preprocessing

Performed using NLTK:  
- Converted text to lowercase  
- Removed HTML tags, special characters, numbers, and punctuation  
- Tokenized text into words  
- Removed stopwords  
- Applied lemmatization followed by stemming  
  
A new column 'cleaned\_review' was added containing the preprocessed text.

# 5. Feature Engineering

- Vectorization Techniques:  
 - Bag-of-Words (BoW)  
 - TF-IDF (used for traditional ML models)  
- Textual Features:  
 - Word count  
 - Character count  
 - Average word length

# 6. Model Development

Traditional Machine Learning Models:

| Model | Accuracy (Approx.) |  
|------------------------|--------------------|  
| Logistic Regression | 87% |  
| Support Vector Machine | 88% |

- Evaluation Metrics: Precision, Recall, F1-score, and Confusion Matrix  
- Observations: Both models performed well on TF-IDF features, with SVM slightly outperforming Logistic Regression.

Deep Learning Model: LSTM (Keras):

- Text Tokenization: Done using Keras Tokenizer (max vocab: 10,000)  
- Input Shape: Padded sequences (length = 200)  
- Model Architecture:  
 - Embedding Layer  
 - LSTM Layer (64 units)  
 - Dropout Layer  
 - Dense Layer with ReLU  
 - Final Sigmoid Output Layer  
- Training:  
 - Epochs: 5  
 - Batch size: 128  
 - Early stopping used for regularization  
- LSTM Accuracy: ~89%

# 7. Model Evaluation

- Traditional models:  
 - Quick to train  
 - Effective for simple text classification  
- LSTM:  
 - Outperformed traditional models slightly  
 - Better at capturing sequence patterns in reviews

# 8. Conclusion

- All models performed well with balanced and cleaned data.  
- TF-IDF + SVM is efficient and fast for small-to-medium projects.  
- LSTM is ideal for more complex patterns and longer texts.  
- Future work could include BERT or fine-tuned transformers for improved accuracy and contextual understanding.

# 9. Recommendations

- Use LSTM or BERT for production-level applications with sufficient resources.  
- For lightweight tasks, Logistic Regression or SVM with TF-IDF is recommended.  
- Perform hyperparameter tuning and cross-validation to further improve model performance.

**10-Video Link-**

<https://drive.google.com/file/d/1NtfAHfJIp5YtOiCvMiMbqwLRw1J0o1-t/view?usp=sharing>