

P15: Sentiment Classification of IMDb Movie Reviews

CS485 – Project Presentation

Zervos Spiridon Chrisovalantis (csd4878)
Drakakis Rafail (csd5310)

University of Crete

Project Objectives

- ▶ Build a sentiment classification pipeline for IMDb reviews
- ▶ Apply classical and deep learning models
- ▶ Evaluate performance: accuracy, speed, and confusion matrices
- ▶ Analyze model behavior and limitations

Dataset Overview

- ▶ **Name:** IMDb Large Movie Review Dataset
- ▶ **Size:** 25k train / 25k test reviews
- ▶ **Labels:** Binary – Positive (1), Negative (0)
- ▶ **Source:**
<https://ai.stanford.edu/~amaas/data/sentiment/>

Text Preprocessing

- ▶ Lowercasing, tokenization using NLTK
- ▶ Removing punctuation, numbers, stopwords
- ▶ For deep learning:
 - ▶ Vocabulary built (min frequency = 2, max size = 20,000)
 - ▶ Sequences padded/truncated to 200 tokens

Feature Engineering

- ▶ **Classical ML:** TF-IDF vectorization (unigrams, top 5,000 features)
- ▶ **Deep Learning:** Trainable embeddings (dimension = 100)

Classical Model Configurations

- ▶ **Logistic Regression:** L2 regularization, $C = 1.0$
- ▶ **Naïve Bayes:** Multinomial, $\alpha = 1.0$
- ▶ **Linear SVM:** $C = 1.0$

Deep Learning Architectures

LSTM:

- ▶ Single layer, 128 hidden units
- ▶ Uses final hidden state

CNN:

- ▶ Filters: sizes [3, 4, 5], 100 each
- ▶ Max pooling and FC layer

Training Setup

- ▶ Optimizer: Adam, learning rate = $1e^{-3}$
- ▶ Epochs: 5, Batch size: 64
- ▶ Implemented in PyTorch
- ▶ Inference script: `predict.py` (Can be used to predict new reviews)

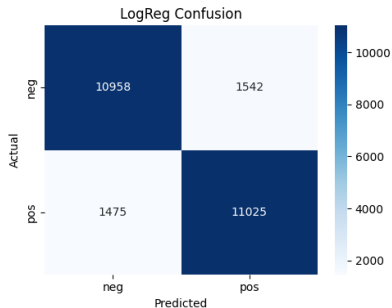
Classical Model Results

Model	Accuracy	Time (ms/sample)
Logistic Regression	0.880	0.47
Naïve Bayes	0.840	0.13
Linear SVM	0.863	0.65

TF-IDF features, evaluated on 25k test samples

Logistic Regression Report

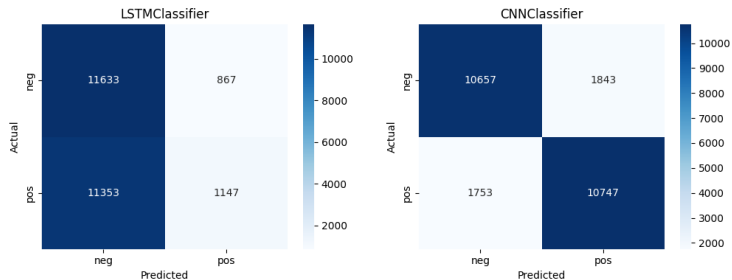
- ▶ Precision, Recall, F1-score: 0.88 for both classes
- ▶ Balanced performance across sentiment labels
- ▶ Fast inference: 0.47 ms/sample



Confusion Matrix: Logistic Regression

Deep Learning Results

Model	Accuracy	Notes
LSTM	0.511	Underfitting, unstable learning
CNN	0.856	Competitive, robust features



Confusion Matrices: LSTM (left), CNN (right)

Model Comparison

- ▶ **Best Accuracy:** Logistic Regression (0.88)
- ▶ **Best DL Model:** CNN (0.856)
- ▶ **Fastest Inference:** Naïve Bayes (0.13 ms/sample)
- ▶ **Underperformer:** LSTM (0.511), due to lack of tuning and pretraining

Error Analysis

- ▶ **LSTM:** Poor generalization without pretrained embeddings
- ▶ **CNN:** Stronger with local patterns (n-grams)
- ▶ **All Models:**
 - ▶ Misclassify short or sarcastic reviews
 - ▶ Struggle with implicit sentiment

Conclusion

- ▶ Classical methods remain strong
- ▶ CNNs show promise with competitive accuracy
- ▶ LSTM requires optimization: embeddings, attention, deeper networks

Questions?