

# Sentiment Analysis with Convolutional Neural Networks: A Final Project Report

Macalalag, Bedis, Ledesma

December 2025

## Abstract

This report details a sentiment analysis project using Convolutional Neural Networks (CNNs) on the IMDB movie review dataset. The implementation achieves high accuracy in binary sentiment classification through text preprocessing, model training, and real-time prediction. The report outlines the methodology, results, and provides access to the complete project resources.

## 1 Introduction

Sentiment analysis determines the emotional tone of text, classifying it as positive or negative. This project applies a CNN to analyze IMDB movie reviews, leveraging convolutional layers to capture local text patterns.

### 1.1 Project Objectives

Develop a CNN model for sentiment classification

Train on IMDB dataset with preprocessing

Create a CLI for predictions

Evaluate performance with standard metrics

### 1.2 Course Information

Course: CS 3A (Computer Science 3A)

Semester: 2025-2026

Authors: Macalalag, Bedis, Ledesma

Repository: GitHub Link

## 2 Literature Review

Convolutional Neural Networks, originally designed for image processing, have been successfully applied to text classification tasks. The use of 1D convolutions allows the model to capture local patterns in text sequences, making it suitable for sentiment analysis.

The IMDB dataset, consisting of 50,000 movie reviews, is a standard benchmark for sentiment analysis tasks. Previous work has shown that CNNs can achieve competitive performance compared to recurrent neural networks like LSTMs.

## 3 Methodology

### 3.1 Dataset

The project uses the IMDB Movie Reviews dataset from Kaggle, containing 50,000 reviews split evenly between positive and negative sentiments. The dataset is preprocessed through tokenization and padding to prepare it for the neural network.

### 3.2 Model Architecture

The CNN architecture consists of the following layers:

**Embedding Layer:** Converts words to dense vectors (vocab\_size=20000, embed\_dim=128)

**Conv1D Layer:** 128 filters with kernel size 3, ReLU activation

**GlobalMaxPooling1D:** Reduces dimensionality

**Dense Layer:** 128 units with ReLU activation

**Dropout:** 0.5 rate for regularization

**Output Layer:** 1 unit with Sigmoid activation for binary classification

### 3.3 Training Configuration

Optimizer: Adam (learning rate = 0.001)

Loss Function: Binary Crossentropy

Batch Size: 128

Maximum Epochs: 10

Callbacks: Early Stopping and ReduceLROnPlateau

## 4 Implementation

### 4.1 Technologies Used

Python 3.8+

TensorFlow/Keras for deep learning

NumPy and Pandas for data processing

Jupyter Notebook for development

Matplotlib for visualization

## 4.2 Code Structure

The project consists of several key files:

`sentiment-analysis.ipynb`: Jupyter notebook for model training and evaluation

`sentiment-analyser.py`: Command-line prediction script

`app.py`: Additional application code

`requirements.txt`: Python dependencies

`sentiment_model.h5`: Pretrained model weights

## 4.3 Installation and Setup

To set up the project environment:

```
1 git clone https://github.com/rjledesma/sentiment-analysis-final-cnn.git
2 cd sentiment-analysis-final-cnn
3 pip install -r requirements.txt
```

Download the IMDB dataset from Kaggle and place `IMDB.Dataset.csv` in the project root.

## 5 Results and Evaluation

The model achieves 85-90% test accuracy. Metrics include precision (0.87), recall (0.88), F1-score (0.87).

Example prediction: "Amazing movie!" Positive (0.92 confidence).

Challenges: Overfitting mitigated by dropout; dataset imbalance handled via balanced splits.

## 6 Discussion

The CNN effectively captures sentiment features. Strengths: Speed, accuracy. Limitations: Context beyond n-grams. Future: Explore transformers or multilingual data.

## 7 Conclusion

This project demonstrates CNN efficacy for sentiment analysis, providing a robust, accessible implementation for further research.

## 8 References

1. Maas et al. (2011). Learning word vectors for sentiment analysis.
2. Kim (2014). Convolutional neural networks for sentence classification.
3. TensorFlow Docs. <https://www.tensorflow.org/>

## A Project Files Access

For complete access to the project files, including the pretrained model and dataset, download the archive from: [Google Drive Link](#)

The archive includes:

- `sentiment_model.h5` (pretrained CNN model)
- `sentiment-analyser.py` (prediction script)
- `sentiment-analysis.ipynb` (training notebook)
- `app.py` (additional code)
- `requirements.txt` (dependencies)
- `README.md` (documentation)
- `IMDB_Dataset.csv` (dataset)