**SUMMER TRAINING/INTERNSHIP**

**PROJECT REPORT**

(Term June-July 2025)

## *SENTIMENT ANALYSIS USING NEURAL NETWORKS*

Submitted by

**Aman Rai**

**Registration Number : 12307571**

**Course Code: CSE393**

Under the Guidance of

**Mahipal Singh Papola**

# School of Computer Science and Engineering

**CERTIFICATE**

This is to certify that Aman Rai, bearing Registration no 12307571 has completed [**CSE393**] project titled, **"** **SENTIMENT ANALYSIS USING NEURAL NETWORKS"** under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

Name of the Supervisor: Mahipal Singh Papola

School of Computer Science Engineering Lovely Professional University Phagwara, Punjab.

Date: 12th July, 2025

**Acknowledgement**

I would like to express my sincere gratitude to Mahipal Singh Papola, faculty at LPU, for providing invaluable guidance, continuous support, and encouragement throughout the duration of this project.

I am also grateful to LPU, Head of the Department, for providing me with the opportunity to undertake this project and for the resources made available to me.

I would like to extend my thanks to the faculty members of ‘School of Computer Science and Engineering’ for their academic support and the knowledge they have imparted during my course of study, which has been instrumental in the successful completion of this project.

I am thankful to LPU, for providing me with the internship opportunity and the practical exposure to real-world challenges in the healthcare technology domain.

I would also like to acknowledge the support of my peers and friends who have assisted me with their valuable suggestions and encouragement throughout this project.

Finally, I express my heartfelt gratitude to my parents and family members for their unwavering support, understanding, and encouragement throughout my academic journey.

Aman Rai

Reg no – 12307571

**1. ABSTRACT**

**2. INTRODUCTION**  
2.1 Background  
2.2 Problem Statement  
2.3 Objective of the Study  
2.4 Scope of the Project

**3. DATA DESCRIPTION**  
3.1 Source of Dataset  
3.2 Initial Dataset Features  
3.3 Feature Selection and Cleaning

**4. DATA PREPROCESSING**  
4.1 Data Cleaning  
    4.1.1 Text Normalization (lowercasing, punctuation removal, etc.)  
4.2 Sentiment Labelling using VADER  
    4.2.1 Compound Score Thresholding  
4.3 Final Dataset Overview

**5. BASELINE SENTIMENT ANALYSIS MODELS**  
5.1 VADER (Valence Aware Dictionary for sEntiment Reasoning)  
5.2 RoBERTa (Robustly Optimized BERT Pretraining Approach)  
5.3 5.3 Comparison of VADER and RoBERTa

**6. NEURAL NETWORK MODELS**  
6.1 Model 1: Dense + Flatten  
6.2 Model 2: Convolutional Neural Network (CNN)  
6.3 Model 3: Long Short-Term Memory (LSTM)  
6.4 Hyperparameter Settings

**7. MODEL COMPARISON AND RESULTS**  
7.1 Performance Comparison  
7.2 Analysis and Interpretation

**8. DEPLOYMENT USING STREAMLIT**  
8.1 Application Overview  
8.2 Technical Architecture  
8.3 Streamlit User Interface

8.4 Code Snippet

8.5 Hosting and Accessibility

**9. SCREENSHOTS OF STREAMLIT APP**  
9.1 9.1 App Landing Page  
9.2 9.2 Input and Output Example: Positive Review  
9.3 9.3 Input and Output Example: Negative Review

**10. CONCLUSION AND FUTURE WORK**  
10.1 Summary of Findings  
10.2 Possible Improvements  
10.3 Potential Applications

**11. REFERENCES**  
List of all research papers, documentation, tools, and libraries used.

**1. ABSTRACT**

Customer reviews are now an essential source of data for determining user satisfaction and enhancing goods and services. This project uses both rule-based and machine learning models to analyze the sentiment of Amazon book reviews and categorize them as either positive or negative. Essential data preprocessing is the first step in the workflow. Next, VADER, a rule-based sentiment analysis tool, is used for initial sentiment labeling. Finally, a transformer-based model called RoBERTa is compared.

Following the labeling of the reviews, the best model for binary sentiment classification is determined by training and evaluating three distinct neural network architectures: Dense + Flatten, Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM). The CNN model outperformed the others in terms of accuracy and generalization.

Users can now input a review and receive an instant sentiment prediction thanks to the final model's user-friendly interface, which was deployed through a Streamlit web application and hosted on Streamlit Cloud. This project demonstrates how real-world textual data can be effectively analyzed using a combination of deep learning and natural language processing techniques.

1. **INTRODUCTION**
   1. **Background**

User-generated content, like online reviews, has become a crucial component of e-commerce platforms in the digital age. Millions of customer reviews are available on websites like Amazon, providing insightful information about user satisfaction, product quality, and consumer preferences. The sheer volume of these reviews makes it impractical to analyse them manually, which is why automated sentiment analysis techniques are becoming increasingly important.  
  
The process of computationally recognizing and classifying viewpoints expressed in text to ascertain the author's attitude toward a specific subject or product is known as sentiment analysis, a branch of natural language processing (NLP). It is essential to industry-wide decision-making processes because it helps companies track brand reputation, comprehend consumer feedback, and improve user experience.

* 1. **Problem Statement**

Many conventional methods fall short in capturing the semantic subtleties and contextual meaning of user reviews, even with the availability of sophisticated sentiment analysis models. Moreover, manual labelling takes time, and not all reviews are labelled. Consequently, an efficient solution is required that can:  
  
- Automatically categorize reviews' sentiments,  
  
- To determine which model is best suited for binary classification (positive or negative), compare several models.  
  
- Provide the solution in a format that is easy for users to access.

* 1. **Objective of the Study**

The primary objectives of this project are:

* To preprocess and clean Amazon book reviews for analysis,
* To assign sentiment labels using a rule-based model (VADER) and compare results with a transformer-based model (RoBERTa),
* To train and evaluate multiple neural network architectures — Dense + Flatten, CNN, and LSTM — for sentiment classification,
* To compare the performance of these models using standard evaluation metrics,
* To deploy the best-performing model using Streamlit for real-time sentiment prediction
  1. **Scope of the Project**

This project is solely concerned with classifying Amazon book reviews as either positive or negative. The study looks at both contemporary deep learning techniques and conventional sentiment scoring. CNN and LSTM are the limits of advanced neural network architectures. The project does not investigate multi-class sentiment classification, aspect-based sentiment analysis, or multilingual processing, but it does include deployment as a proof of concept.

1. **DATA DESCRIPTION**
   1. **Source of Dataset**

The dataset used in this study consists of Amazon book reviews and was sourced from a publicly available repository. It contains user-generated reviews on various books, including textual feedback and associated metadata. The dataset is particularly suitable for sentiment analysis due to its large volume (3 million rows) and diversity in review content.

The link to the data set is provided in the reference.

* 1. **Initial Dataset Features**

Before preprocessing, the dataset consisted of the following columns:

* **Id**: A unique identifier for each review.
* **Title**: The title of the book being reviewed.
* **Price**: The listed price of the book.
* **User\_id**: An anonymized identifier for the user who submitted the review.
* **profileName**: The display name of the user.
* **review/helpfulness**: A metric indicating how many users found the review helpful.
* **review/score**: The rating assigned to the book (typically on a 1 to 5 scale).
* **review/time**: The timestamp when the review was posted.
* **review/summary**: A short headline or summary of the review.
* **review/text**: The main body of the review containing detailed user feedback.
  1. **Feature Selection and Cleaning**

To streamline the analysis and focus on the most relevant information, the following columns were dropped:

Price, User\_id, profileName, review/helpfulness, review/time, review/text

These columns were either irrelevant to sentiment labelling (e.g., Price, User\_id), redundant (e.g., profileName), or too noisy/unstructured for this phase of the project (e.g., review/text).

After cleaning, the retained columns were:

* Id: To maintain a reference for each review.
* Title: To retain context about the subject of the review.
* review/score: To potentially use as a reference for ground truth sentiment or validation.
* review/summary: Used as the primary input for sentiment classification, as it is concise yet informative.

The cleaned dataset forms the foundation for both rule-based and neural network-driven sentiment classification in the later stages of the project.

1. **DATA PREPROCESSING**

Textual data must be thoroughly pre-processed to guarantee that it is clean, standardized, and appropriate for model input for sentiment analysis to be effective. This phase is essential for lowering noise and enhancing machine learning models' overall functionality.

* 1. **Data Cleaning**

The initial cleaning phase involved filtering and transforming the dataset to retain only relevant information for sentiment classification.

* Null and Empty Values: Reviews with missing or empty review/summary entries were removed, as they do not provide any textual input for sentiment analysis.
* Duplicates: Duplicate entries, if any, were dropped to avoid data leakage or bias in model training.
  + 1. **Text Normalization**
* The review summaries were processed through a standard normalization pipeline:
* **Lowercasing**: All characters were converted to lowercase to reduce vocabulary size and eliminate case sensitivity.
* **Punctuation Removal**: Common punctuation marks were removed to simplify the text structure.
* **Whitespace Normalization**: Extra spaces and newlines were stripped to maintain consistency.

No lemmatization or stemming was applied in this phase, as the review summaries were already relatively short and structured.

* 1. **Sentiment Labelling using VADER**

To facilitate supervised learning, the reviews were labelled as either **positive** or **negative** based on the **VADER (Valence Aware Dictionary for Sentiment Reasoning)** sentiment analyser. VADER is a rule-based model specifically tuned for sentiments expressed in social media and short texts, making it suitable for review summaries.

* + 1. **Compound Score Thresholding**:
  + Reviews with a **compound score ≥ 0.05** were labelled as **positive**.
  + Reviews with a **compound score ≤ -0.05** were labelled as **negative**.
  + Neutral-range reviews (between -0.05 and 0.05) were dropped to ensure binary classification clarity.
  1. **Final Dataset Overview**

After cleaning and labelling:

* The dataset was reduced to entries with clearly positive or negative sentiments.
* Each entry consisted of the review ID, book title, review score, normalized summary, and sentiment label.
* This processed dataset was then used as input for training and evaluating the deep learning models.

1. **BASELINE SENTIMENT ANALYSIS MODELS**

Before training neural network models, it is essential to establish baseline performance using existing sentiment analysis techniques. In this project, two well-known models were employed for this purpose: a rule-based approach using VADER, and a transformer-based deep learning model, RoBERTa.

* 1. **VADER (Valence Aware Dictionary for Sentiment Reasoning)**

VADER is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in short, informal texts such as social media posts and product reviews. It outputs four sentiment scores for a given input: positive, negative, neutral, and compound.

* The compound score, a normalized value between -1 and 1, reflects the overall sentiment polarity of the input.
* In this project, the compound score was used to classify reviews:
* Compound > 0 → Positive
* Compound ≤ 0 → Negative

VADER is computationally efficient and provides quick results, making it a suitable tool for initial labeling and exploration. However, it is limited in its ability to understand contextual and domain-specific nuances**.**

* 1. **RoBERTa (Robustly Optimized BERT Pretraining Approach)**

RoBERTa is a transformer-based language model developed by Facebook AI, built on top of BERT (Bidirectional Encoder Representations from Transformers). It is trained on a significantly larger corpus and uses dynamic masking, leading to improved performance in a variety of NLP tasks, including sentiment classification.

For this project, RoBERTa was used in a zero-shot or fine-tuned inference setup (depending on the variant used), where review summaries were passed through the model to obtain sentiment probabilities for positive and negative classes.

* RoBERTa was evaluated for its ability to detect sentiment with greater semantic understanding than VADER.
* Due to its pretrained nature and larger model size, RoBERTa was more computationally expensive and slower compared to VADER.
  1. **Comparison of VADER and RoBERTa**

A graph of blue squares

AI-generated content may be incorrect.

A graph of a positive score

AI-generated content may be incorrect.

**A screenshot of a graph

AI-generated content may be incorrect.**

In practice, VADER was used for the final sentiment labelling because:

* It provided reasonably accurate binary labels for short review summaries.
* It was significantly faster and simpler to implement for a large dataset.
* It enabled consistent and reproducible preprocessing for training deep learning models.

1. **NEURAL NETWORK MODELS**

To improve upon baseline sentiment classification, this project implemented and evaluated three neural network architectures using the VADER-labeled dataset. Each model was designed to classify short Amazon book review summaries as either positive or negative, based on the textual content.

* 1. **Model 1: Dense + Flatten**

The first architecture was a simple **feedforward neural network** using a flattening layer followed by fully connected dense layers:

* **Text vectorization** was done using tokenization followed by sequence padding.
* The input was passed through an **embedding layer** to generate word representations.
* A **Flatten layer** converted the output into a 1D vector, followed by one or more **Dense layers** with ReLU activation.
* The final layer used a **sigmoid activation** to output a binary sentiment prediction.

A graph of a model loss

AI-generated content may be incorrect.A graph of a graph

AI-generated content may be incorrect.This model served as a baseline for evaluating performance of deeper and more specialized architectures.

* 1. **Model 2: Convolutional Neural Network (CNN)**

The second model was a **Convolutional Neural Network**, adapted for text classification:

* After tokenization and embedding, 1D **convolutional layers** were used to capture local patterns in word sequences.
* The convolutional layers were followed by **max pooling** layers to reduce dimensionality and retain important features.
* The output was flattened and passed through dense layers, ending in a sigmoid output unit.

CNNs are effective in extracting position-invariant n-gram features, making them suitable for short and moderately structured text such as review summaries.

A graph with blue and orange lines

AI-generated content may be incorrect.

A graph with blue line and orange line

AI-generated content may be incorrect.

* 1. **Model 3: Long Short-Term Memory (LSTM)**

The third model utilized an **LSTM network**, a type of recurrent neural network capable of capturing sequential dependencies in text:

* After embedding, the review summaries were passed through one or more **LSTM layers**.
* LSTMs processed input sequentially, maintaining a memory of previous tokens, which allows the model to consider word order and long-range dependencies.
* A dense layer followed the LSTM output, with sigmoid activation for final classification.
* A graph of a graph

  AI-generated content may be incorrect.This model was designed to explore the impact of sequential understanding in sentiment classification.

A graph of a model loss

AI-generated content may be incorrect.

* 1. **Hyperparameter Settings**

Common hyperparameters across all models included:

* Embedding dimension: 100
* Max sequence length: Based on the average review summary length
* Batch size: 32
* Epochs: 10–15 (with early stopping based on validation loss)
* Optimizer: Adam
* Loss Function: Binary Crossentropy
* Evaluation Metric: Accuracy

1. **MODEL COMPARISON AND RESULTS**
   1. **Performance Comparison**

All three models were trained on the same dataset under similar conditions. Their performance was evaluated on a held-out test set using accuracy, loss, and confusion matrix analysis. The following table summarizes the key results:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Training Accuracy** | **Test Accuracy** | **Test Loss** | **Remark** |
| **Dense + Flatten** | **90.46%** | **90.08%** | **0.253** | **Prone to overfitting; fast to train** |
| **CNN** | **97.25%** | **94.01%** | **0.161** | **Best generalization and performance** |
| **LSTN** | **88.06%** | **88.05%** | **0.354** | **Struggled with short text context** |

* 1. **Analysis and Interpretation**
* **Dense + Flatten**:
  + Performed reasonably well on both training and test datasets, indicating minimal overfitting.
  + However, its performance plateaued compared to deeper architectures, as it lacks the capacity to extract complex patterns from text.
* **CNN**:
  + Achieved the highest training and test accuracy, along with the lowest test loss.
  + Demonstrated strong generalization, suggesting that convolutional layers were effective in capturing relevant local features in the review summaries.
  + The performance gap between training and test accuracy was minimal, indicating robust model stability.
* **LSTM**:
  + Achieved the lowest accuracy and highest test loss among the three models.
  + Despite being designed for sequential data, its performance may have been limited by the short and context-light nature of review summaries.
  + The model may have underfit due to insufficient sequence complexity or suboptimal tuning.
  1. **Conclusion**

Among the three models, the **Convolutional Neural Network (CNN)** provided the best performance across all evaluation metrics. Its ability to capture position-invariant local features from short textual inputs made it the most suitable model for this sentiment classification task. Consequently, the CNN was selected for deployment in the final web application.

1. **Streamlit User Interface**

To make the sentiment analysis model accessible and user-friendly, the best-performing neural network (CNN) was deployed as a web application using **Streamlit**, an open-source Python library that allows for the rapid development of interactive machine learning tools.

* 1. **Application Overview**

The web application enables users to input a short book review and instantly receive a sentiment prediction based on the trained CNN model. The application is hosted on **Streamlit Cloud**, allowing for public access without the need for local installations or backend infrastructure.

* 1. **Technical Architecture**

The deployed application consists of the following core components:

* **Model Loading**: The trained CNN model is loaded using TensorFlow’s load\_model method.
* **Tokenizer Restoration**: The tokenizer used during training is deserialized from a tokenizer.json file to ensure consistency in input text preprocessing.
* **Preprocessing Pipeline**:
  + User input is first converted into a sequence of tokens using the loaded tokenizer.
  + The sequence is padded to a maximum length of 100 tokens, matching the training configuration.
* **Prediction and Output**:
  + The processed input is passed through the model to generate a sentiment score (a float between 0 and 1).
  + A threshold of 0.5 is applied to determine the final sentiment label:
  + **≥ 0.5** → *Positive 😊*
  + **< 0.5** → *Negative 😞*
  1. **Streamlit User Interface**

The user interface was designed to be minimalistic and intuitive:

* The application displays a title and a short description.
* A text area is provided for users to enter a review.
* A "Predict Sentiment" button triggers model inference.
* The predicted sentiment and associated confidence score are displayed dynamically.
  1. **Code Snippet**

Below is a simplified version of the Streamlit app logic:

A screenshot of a computer program

AI-generated content may be incorrect.

* 1. **Hosting and Accessibility**

The app was deployed via **Streamlit Cloud**, which provides:

* Free hosting with integrated version control via GitHub,
* Seamless deployment pipelines,
* Scalability and browser-based access across platforms.

1. **SCREENSHOTS OF RESULTS**
   1. **App Landing Page**

A screenshot of a computer

AI-generated content may be incorrect.

* 1. **Input and Output Example: Positive Review**

A screenshot of a computer

AI-generated content may be incorrect.

* 1. **Input and Output Example: Negative Review**

A screenshot of a computer

AI-generated content may be incorrect.

**10. CONCLUSION AND FUTURE WORK**

**10.1 Conclusion**

This project used both contemporary deep learning models and conventional sentiment analysis techniques to categorize sentiments in Amazon book reviews. The pipeline started with preprocessing and data cleaning, then used the VADER lexicon-based model to label sentiment. A transformer-based model called RoBERTa was also assessed for comparison.  
  
The labelled data was then used to implement and train three neural network architectures: Dense + Flatten, Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM). After a thorough evaluation, the CNN model was found to be the best-performing architecture, with the lowest loss and a test accuracy of 94.01%.  
  
Streamlit was used to deploy the final CNN model to improve accessibility and usability. This allowed users to submit reviews and receive real-time sentiment predictions through a clear and interactive web interface.

The project shows how well deep learning works for sentiment analysis of brief review texts and emphasizes how useful it is to use lightweight web applications to implement such models.

**10.2 Future Work**

While the results of this project are promising, several improvements and extensions are possible:

* **Multiclass Sentiment Classification**: Expanding beyond binary classification to include neutral or fine-grained emotion categories such as joy, anger, and sadness.
* **Incorporating Full Review Texts**: Including the review/text column instead of just the summary to provide richer input for deeper context modeling.
* **Aspect-Based Sentiment Analysis (ABSA)**: Identifying sentiments toward specific aspects of a book, such as plot, characters, or writing style.
* **Transfer Learning with Fine-Tuned Transformers**: Applying and fine-tuning models like RoBERTa or BERT on the review dataset may further improve accuracy and context understanding.
* **User Feedback Loop**: Incorporating user feedback into the model’s predictions to refine and personalize sentiment analysis over time.
* **Mobile-Friendly Deployment**: Optimizing and deploying the application on mobile platforms to reach a broader audience.

Through these enhancements, the model could evolve into a more comprehensive and intelligent sentiment analysis system suitable for real-world applications in e-commerce and customer experience management.

1. **REFERENCES**
2. Amazon Product Review Dataset: <https://www.kaggle.com/datasets/mohamedbakhet/amazon-books-reviews>