# IMDB Movie Review Sentiment Analysis using Simple RNN

This project focuses on sentiment analysis of IMDB movie reviews using a \*\*Simple Recurrent Neural Network (RNN)\*\*. The goal is to classify movie reviews as either \*\*positive\*\* or \*\*negative\*\* based on their text content. The model is trained on the IMDB dataset, which contains 50,000 labeled reviews, and is deployed as a web application using \*\*Streamlit\*\*.

## Project Overview

The project involves building and training a \*\*Simple RNN\*\* model to classify IMDB movie reviews into two categories:

- \*\*Positive (1)\*\*: Indicates a favorable review.

- \*\*Negative (0)\*\*: Indicates an unfavorable review.

The model is trained on the IMDB dataset, preprocessed to handle text data, and deployed as a web application where users can input a movie review and get the sentiment prediction.

## Dataset

The \*\*IMDB dataset\*\* is a collection of 50,000 movie reviews, labeled as either positive or negative. The dataset is preprocessed to:

1. Limit the vocabulary size to the top 10,000 most frequent words (`max\_features=10000`).

2. Pad or truncate reviews to a fixed length of 500 words (`max\_len=500`) for uniformity.

### Data Preprocessing

- \*\*Word Index Mapping\*\*: Each word in the dataset is mapped to a unique integer index.

- \*\*Padding Sequences\*\*: Reviews shorter than 500 words are padded with zeros, while longer reviews are truncated.

- \*\*Decoding Reviews\*\*: A helper function is provided to decode integer sequences back into readable text for debugging and understanding.

## Model Architecture

The model is built using \*\*TensorFlow\*\* and \*\*Keras\*\* with the following layers:

1. \*\*Embedding Layer\*\*: Converts integer-encoded words into dense vectors of fixed size (128 dimensions).

2. \*\*Simple RNN Layer\*\*: A recurrent layer with 128 units and ReLU activation to capture sequential dependencies in the text.

3. \*\*Dense Layer\*\*: A single neuron with a sigmoid activation function to output the probability of the review being positive.

The model is compiled using:

- \*\*Optimizer\*\*: Adam

- \*\*Loss Function\*\*: Binary Crossentropy (since it's a binary classification problem)

- \*\*Metric\*\*: Accuracy

## Training the Model

The model is trained for \*\*10 epochs\*\* with:

- \*\*Batch Size\*\*: 32

- \*\*Validation Split\*\*: 20% of the training data is used for validation.

- \*\*Early Stopping\*\*: To prevent overfitting, training stops early if the validation loss does not improve for 5 consecutive epochs.

The trained model is saved as `simple\_rnn\_imdb.h5` for later use.

## Prediction

The model can predict the sentiment of new reviews. The prediction process involves:

1. \*\*Preprocessing\*\*: The input text is converted to lowercase, split into words, and mapped to their corresponding integer indices. The sequence is then padded to a length of 500.

2. \*\*Prediction\*\*: The preprocessed input is passed through the model to get a probability score.

3. \*\*Classification\*\*: If the probability score is greater than 0.5, the review is classified as \*\*positive\*\*; otherwise, it is classified as \*\*negative\*\*.

## Deployment

The model is deployed as a \*\*Streamlit web application\*\*. Users can input a movie review, and the application will:

1. Preprocess the input text.

2. Predict the sentiment using the trained model.

3. Display the sentiment (positive or negative) and the prediction score.

## How to Use

1. \*\*Clone the Repository\*\*:

```bash

git clone <repository-url>

```

2. \*\*Install Dependencies\*\*:

```bash

pip install tensorflow streamlit numpy

```

3. \*\*Run the Streamlit App\*\*:

```bash

streamlit run app.py

```

4. \*\*Input a Review\*\*: Enter a movie review in the text box and click "Classify" to see the sentiment prediction.

---

## Dependencies

- Python 3.x, TensorFlow, Streamlit, NumPy

## Conclusion

This project demonstrates the use of a \*\*Simple RNN\*\* for sentiment analysis on the IMDB movie review dataset. By preprocessing the text data and training a neural network, the model achieves accurate sentiment classification. The deployment of the model using \*\*Streamlit\*\* makes it accessible and easy to use for end-users. This project can be extended to handle more complex models (e.g., LSTM, GRU) or applied to other text classification tasks.