MOVIE REVIEWS SENTIMENT ANALYSIS WEB APP

INFO 6105 Data Science Engineering Methods and Tools

By

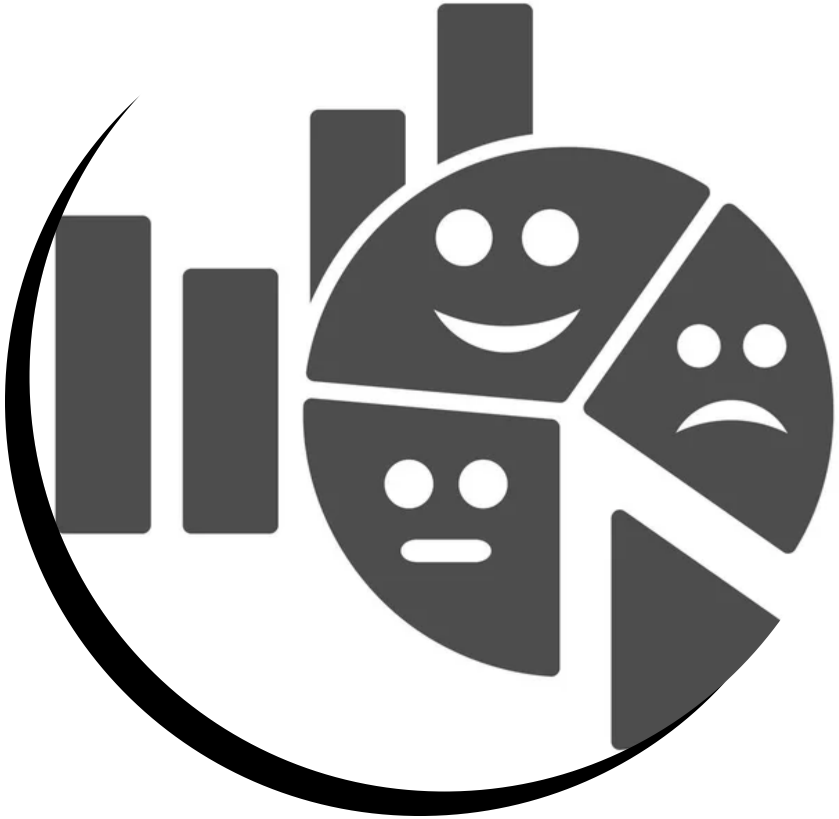
Professor Hong Pan

Presented By

Deepak Kumar

INTRODUCTION

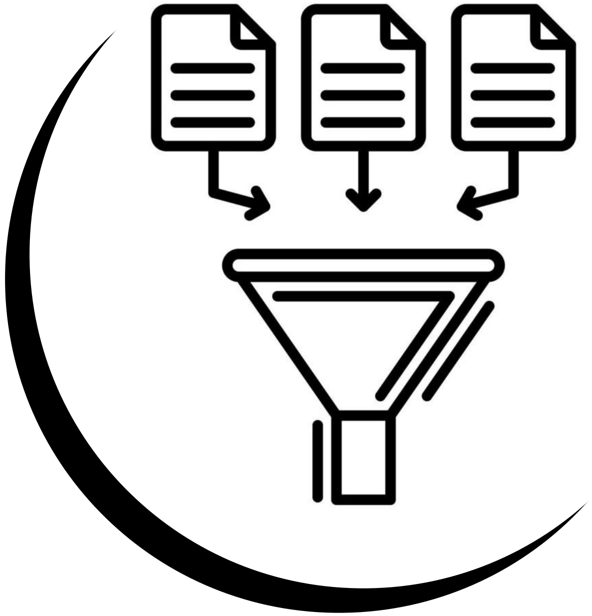
The scope of this project is to create a web tool and deploy on cloud which automatically categorize reviews into positive, or negative sentiment. This tool intends to enhance the movie selection process by providing an aggregated sentiment score, thus offering a nuanced understanding of a film's reception among audiences and critics.



TOOLS AND TECHNOLOGIES

- Python
- PyTorch
- Amazon SageMaker
- AWS Lambda
- AWS API Gateway
- HTML/CSS

DATA COLLECTION



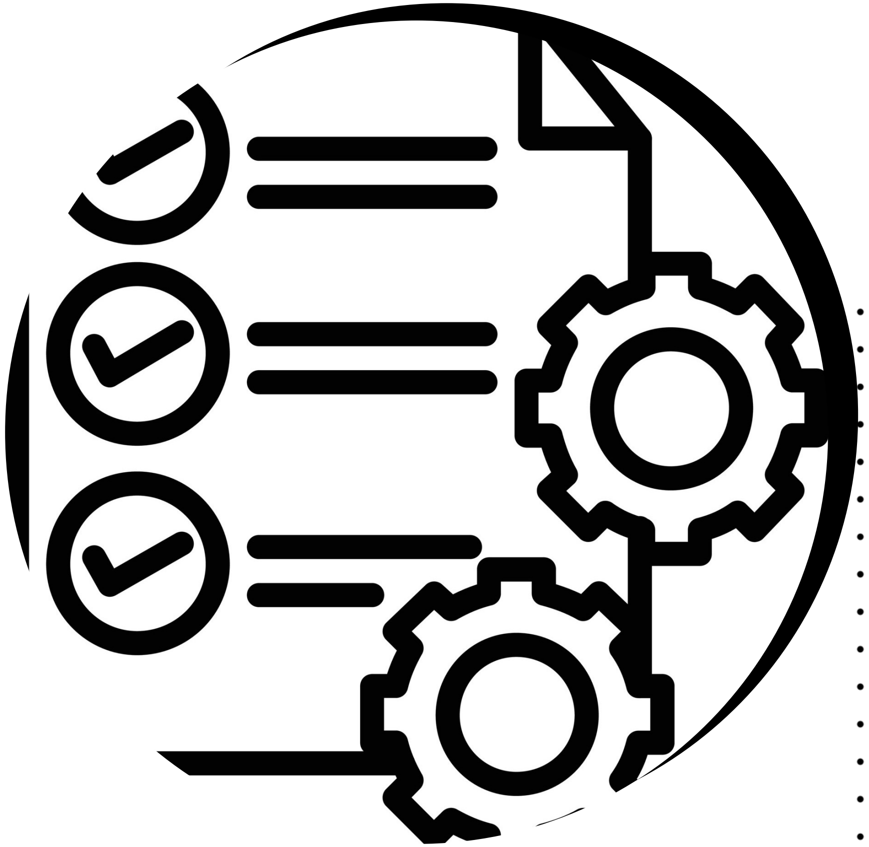
This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. We provide a set of 25,000 highly polar movie reviews for training, and 25,000 for testing.

Publications Using the Dataset

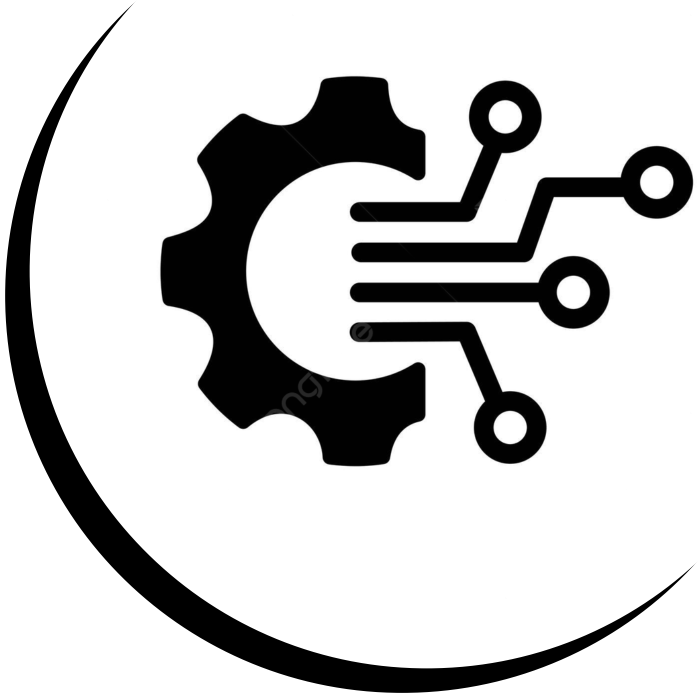
Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).

DATA PROCESSING

- Remove HTML tags : BeautifulSoup library
- Text Normalization : re.sub()
- Tokenization : Python's split()
- Stopword Removal : NLTK's stopword corpus
- Stemming : NLTK's PorterStemmer



DATA TRANSFORMATION



Vocabulary Construction : Build a dictionary mapping the most frequently used words in the dataset to unique integers, reserving special indices for 'no word' and 'infrequent words'

Data Transformation : Convert text data into numerical form by replacing each word in a sentence with its corresponding integer from the vocabulary dictionary.

Sentence Padding : Standardize the length of all sentences to a fixed number using padding

Batch Preparation : Transform entire datasets by converting and padding each sentence, capturing both the transformed data and their original lengths for model training.

MODEL BUILDING AND TRAINING

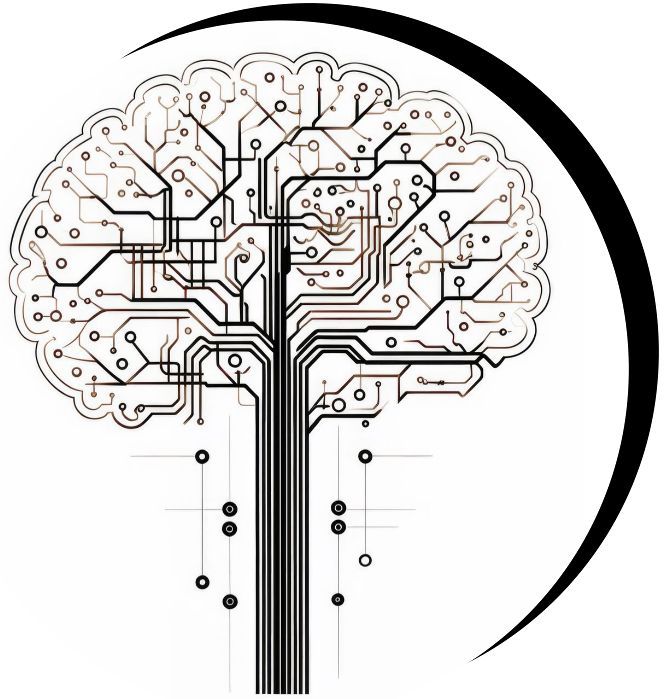
LSTM Architecture: Combines an embedding layer, LSTM layer, dense layer, and sigmoid activation for effective sentiment analysis.

Loss Function: Uses Binary Cross-Entropy Loss to align predicted probabilities with actual binary outcomes.

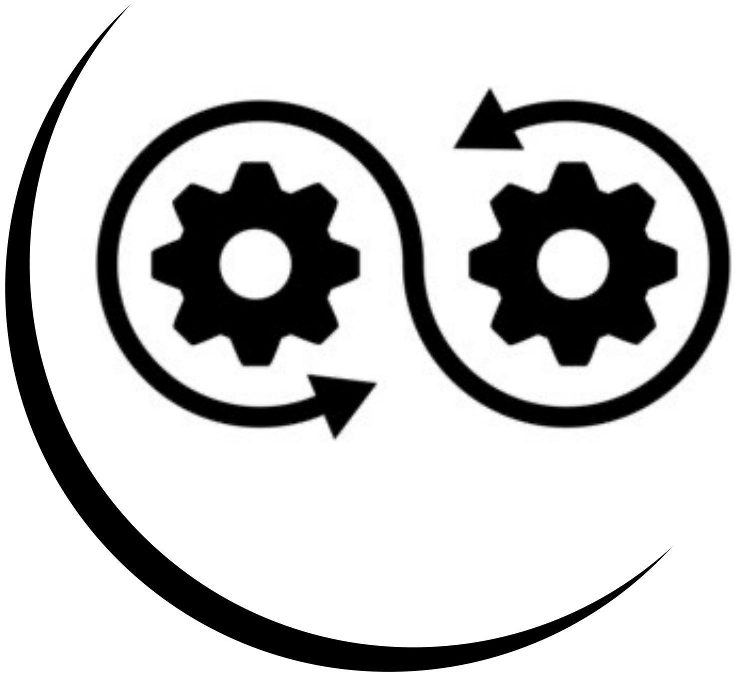
Optimizer: Employs Adam for efficient gradient handling and adaptive learning rate adjustments.

Prevention of Overfitting: Suggests potential regularization methods like dropout to enhance model generalization.

Efficient Training: Utilizes mini-batch training and epochs, with possible cross-validation for optimal parameter tuning.



MODEL DEPLOYMENT FOR TESTING



Deployment Configuration: Deployed the LSTM model on Amazon SageMaker using a single 'ml.m4.xlarge' instance for initial testing.

Data Preparation: Prepared test data by combining sentence lengths with numerical features into a DataFrame, formatted for model prediction.

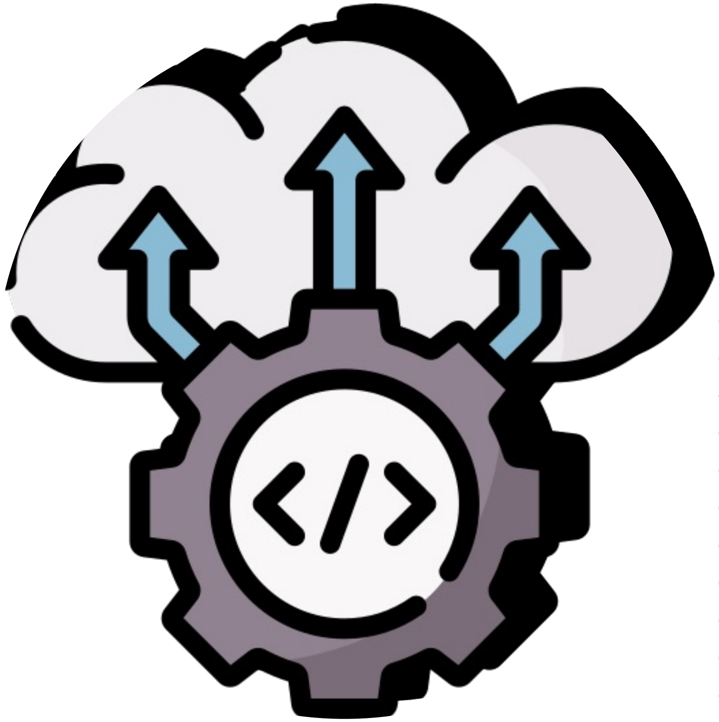
Batch Prediction: Implemented a function to divide test data into batches, enabling efficient prediction without overloading system memory.

Generate Predictions: Executed predictions on batches, rounding output probabilities to binary values (0 or 1) to classify sentiment.

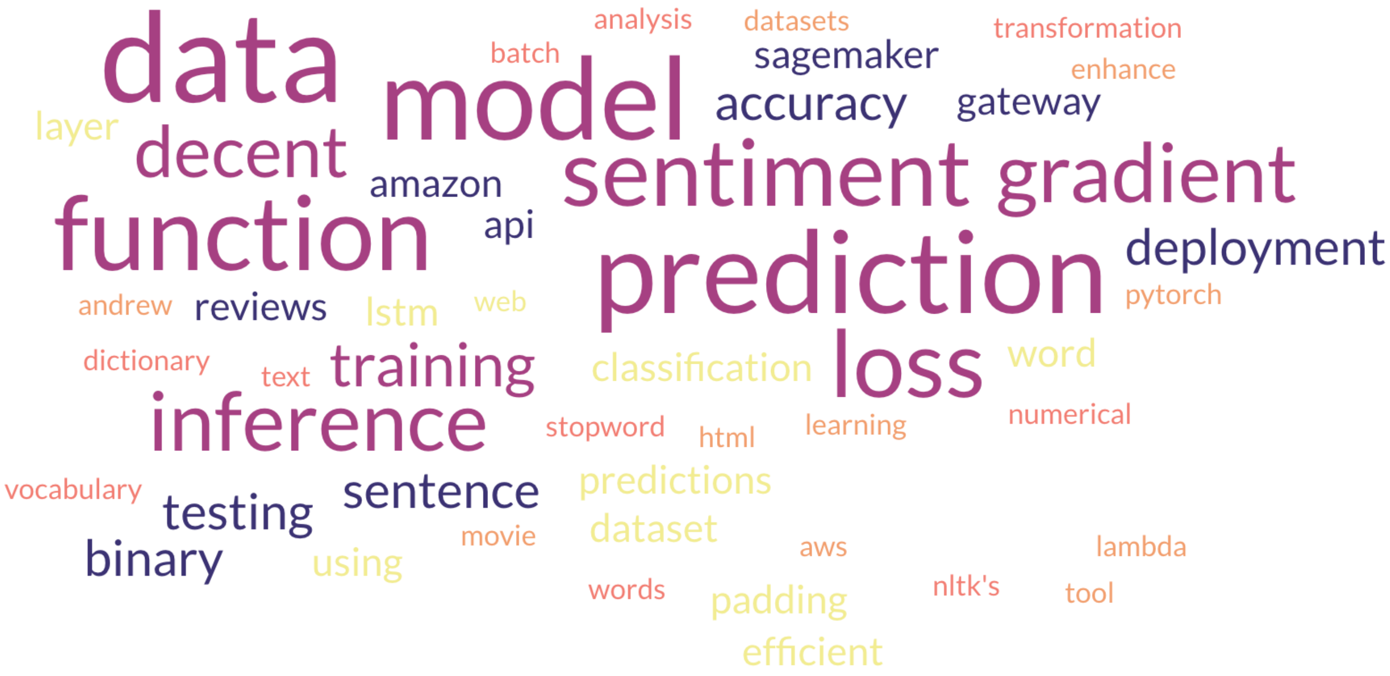
Accuracy Highlight: Achieved a accuracy of 85.264%, demonstrating the model's effectiveness in sentiment classification on unseen data.

MODEL DEPLOYMENT FOR TESTING FOR WEB APP

- Deployment Strategy
  - ✓ Deployed on Amazon SageMaker.
  - ✓ Packages PyTorch model.
  - ✓ Includes predict.py endpoint.
  - ✓ Serves real-time predictions.
- Lambda Function
  - ✓ Handles prediction requests.
  - ✓ Serverless compute service.
  - ✓ Executes model inference.
  - ✓ Connects to API Gateway.
- API Gateway
  - ✓ Routes user requests.
  - ✓ Front door for communication.
  - ✓ Ensures secure interactions.
  - ✓ Supports scaling, efficiency.
- Model Accuracy
  - ✓ Evaluated on reviews.
  - ✓ Achieves 84% accuracy.
  - ✓ Reliable sentiment classification.
  - ✓ Ensures actionable user insights.



LEARNING



data
model
sentiment
gradient
prediction
function
decent
loss
inference
training
accuracy
sagemaker
gateway
deployment
amazon
api
reviews
sentence
testing
binary
classification
word
predictions
dataset
lstm
layer
padding
efficient
using
web
analysis
datasets
transformation
batch
enhance
pytorch
andrew
dictionary
text
stopword
html
learning
numerical
vocabulary
movie
aws
lambda
words
nltk's
tool

REFERENCES

- https://ai.stanford.edu/~amaas/data/sentiment/
- https://pytorch.org/docs/stable/index.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/whatis.html
- https://docs.aws.amazon.com/lambda/
- https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html
- https://github.com/udacity/sagemaker-deployment/tree/master/Project
- https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html

THANK YOU!