TEAM DELTA

SENTIMENT ANALYSIS ON SOCIAL MEDIA DATA

This project focuses on using Natural Language Processing (NLP) and machine learning techniques to analyze sentiment expressed in tweets. By classifying tweets as positive, negative, or neutral, the system helps businesses, politicians, and marketers understand public sentiment in real time. Starting with Twitter, the project aims to expand sentiment analysis to other social media platforms in the future.

PROJECT DESCRIPTION & KEY POINTS

SOCIAL MEDIA INFLUENCE

Social media platforms, such as Twitter, are pivotal in real-time communication and shaping public opinion.

FOCUS ON X (FORMER TWITTER)

X was selected for this project due to its role as a global hub for major announcements, trends, and end-to-end feedback.

GOAL

Building a sentiment analysis system to classify tweets as positive, negative, or neutral, utilizing NLP and machine learning models.

BUSINESS QUESTIONS ADDRESSED

MONITORING FEEDBACK FOR BUSINESSES

How can large corporations monitor millions of tweets for customer feedback and brand reputation efficiently?

POLITICAL IMPACT ANALYSIS

How can politicians understand the real-time impact of their public statements and adjust strategies accordingly?

MARKETING INSIGHTS AND PREDICTIONS

How can marketers use sentiment trends to predict the success of their campaigns and adapt strategies?

OBJECTIVES AND GOALS

DEVELOPING A ROBUST SENTIMENT ANALYSIS SYSTEM

Aiming to accurately classify tweets into sentiment categories using NLP and LSTM-based models

HANDLING LARGE-SCALE DATA

Processing large volumes of Twitter data to generate insights and provide real-time feedback to organizations.

FUTURE EXPANSION

Plan to extend the system to include other platforms like Facebook and Instagram for a comprehensive social media sentiment analysis.

DATA COLLECTION & DATASET OVERVIEW

Dataset Overview:

- Dataset Source: Kaggle

- Training Set: 27,481 tweets
- Test Set: 4,815 tweets

- Dataset Purpose: Classify tweets into positive, neutral, or negative categories

KEY FEATURES

- text: Main content of each tweet.
- sentiment: Label classifying tweet as positive, neutral, or negative.
- selected_text: Highlights the most sentiment-relevant portion of the tweet, identifying the specific part of the tweet that reflects the assigned sentiment.

- **Class Imbalance:** Neutral tweets slightly more frequent, but distribution across sentiments is consistent

- **Tweet Length Distribution:** 10 words on average, requiring padding during preprocessing for uniform input to the model

DATA CLEANING & LANGUAGE FILTERING

HANDLING MISSING VALUES

- Few missing values in the training set
- Test set had ~1,200 missing values, all removed

LANGUAGE FILTERING

- Non-English tweets removed using the “langid” library
- 1,161 entries removed from the training set, 159 from the test set

RESULT AFTER CLEANING

- Training set: 26,319 entries
- Test set: 3,534 entries

INDEXING, TOKENIZATION & NORMALIZATION

NORMALIZATION

- Removed HTML tags, emoticons, symbols, flags, digits, and irrelevant short words
- All text converted to lowercase for consistency
- Stopwords (e.g., "the", "is", "and") removed to focus on essential words

INDEXING

- Created a new index column for standardized text representation

Example of Indexed Tweet:

- Original Tweet: "its at 3 am, im very tired but i can't sleep"
- Final Outcome: "tired sleep try"

TOKENIZATION

- Converted words into numerical tokens for model input
- Considered the top 5,000 most frequent words

Example of Tokenization:

- Word: "day" → Token: 1
- Word: "good" → Token: 2

Vocabulary Size: 23,450 unique words identified across tweets

CONCEPTUAL APPROACH (RNNS AND LSTMS)

SENTIMENT ANALYSIS APPROACH

We use Recurrent Neural Networks (RNNs), which are designed to handle sequential data like text.

WHY RNNS?

Unlike feedforward neural networks, RNNs maintain a "memory" through a hidden state, making them ideal for tasks like time series and text analysis.

LSTM INTRODUCTION

Long Short-Term Memory (LSTM) networks improve RNNs by handling long-term dependencies through specialized gates (input, output, forget).

PURPOSE

LSTMs can decide which information to retain or forget, helping them manage sequential relationships in data more effectively than traditional RNNs.

LSTM MODEL
KEY FEATURES

Memory cells and gates:

LSTMs use three main gates (input, forget, and output) to control the flow of information.

Activation functions:

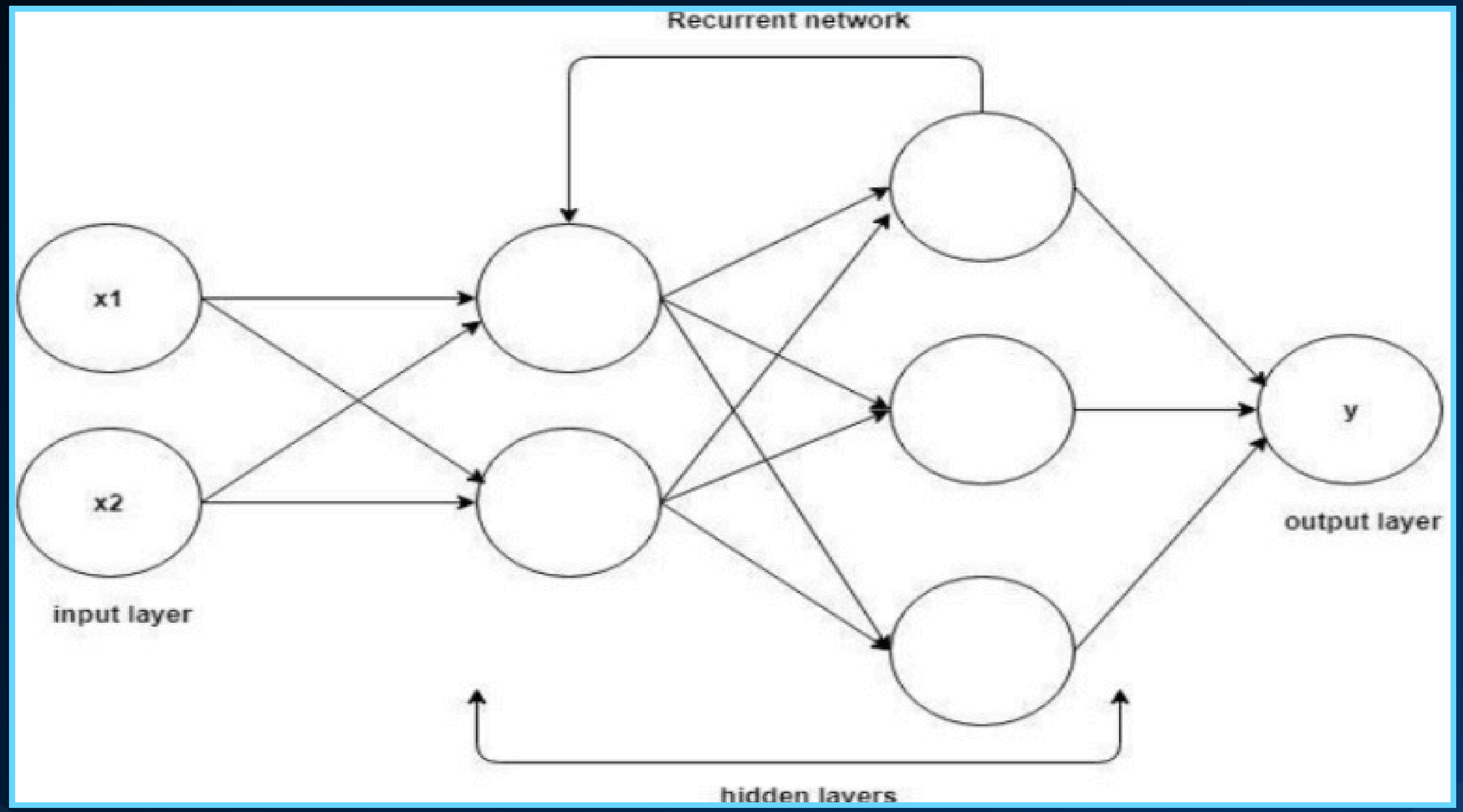
- Tanh: Handles both positive and negative values.
- ReLU (Rectified Linear Unit): Returns input for positive values and sets negative values to zero.
- Softmax: Generates probabilities for class predictions, ensuring outputs sum to 1.

WHY THESE FUNCTIONS?

These functions help the LSTM model handle non-linear patterns and make confident predictions based on tweet content.

MODEL IMPLEMENTATION
LSTM ARCHITECTURE

Tool used: The implementation was carried out using Keras, a high-level API for building neural networks.



SUMMARY OF LSTM MODEL ARCHITECTURE

Layer (Type)	Purpose	Neurons	Activation Function
Embedding Layer	Converts input to dense vectors	50	
LSTM Layer	Captures sequential dependencies	64	Tanh, Sigmoid
GlobalMaxPooling1D	Downsamples data to a fixed-size representation		
Dense Layer 1	Applies transformations to input from LSTM	128	ReLU
Dropout Layer	Randomly drops connections (50%) to prevent overfitting		
Dense Layer 2	Further refines learned features	32	ReLU
Dropout Layer	Randomly drops connections (20%) to prevent overfitting		
Output Layer	Produces final sentiment classification (positive, neutral, negative)	3	Softmax

MODEL HYPERPARAMETERS & OPTIMIZATION

- Batch size: 32 (standard setting to balance performance and training speed).
- Epochs: 5 (number of times the model cycles through the entire training dataset).
- Loss function: Categorical crossentropy, appropriate for multi-class classification tasks.
- Regularization: L2 Regularization applied to dense layers to prevent overfitting.
- Optimizer: Adam (Adaptive Moment Estimation), chosen for its efficiency and adaptability.
- Learning rate: Set at 0.01, which is optimal for the dataset and model structure.

EVALUATION PROTOCOL

- Precision = (True Positives)/(True Positives & False Positives):

Measures how many predicted positive samples are actually positive.

- Recall = (True Positives)/(True Positives & False Negatives):

Measures how many actual positive samples were correctly predicted.

- F1-Score = 2×[(Precision× Recall)/(Precision & Recall)]:

Balances precision and recall, providing a comprehensive performance measure that considers both false positives and false negatives.

ENHANCING MODEL PERFORMANCE: HYPERPARAMETER TUNING

Purpose of hyperparameter tuning:

We employed hyperparameter tuning to optimize the model's performance by adjusting key settings that control the learning process.

Hyperparameters tested:

- Epochs: [5, 10]
- LSTM Neurons: [32, 64, 128]
- Layers: [1, 2, 3]
- Neurons per Layer: [[32], [128, 32], [128, 64, 32]]
- Dropout Rate: [0.2]
- Learning Rate: [0.001]
- Embedding Output Dimensions: [50]
- Batch Size: [32]

TESTING PROCESS

- A total of 54 different model configurations were tested.
- The configuration with the highest accuracy was selected for further development, as described in the earlier LSTM architecture section.

PERFORMANCE METRICS:
PRECISION, RECALL & F1-SCORE

Class	Precision	Recall	F1-Score	Support
Class 0 (Negative)	0.78	0.64	0.71	955
Class 1 (Neutral)	0.67	0.74	0.71	1,371
Class 2 (Positive)	0.76	0.79	0.77	1,049
Accuracy			0.73	3,375
Macro Avg	0.74	0.72	0.73	3,375
Weighted Avg	0.73	0.73	0.73	3,375

Overall performance:

The LSTM model achieved 73% accuracy, with balanced precision, recall, and F1-scores across all classes.

Class-specific performance:

- Class 0 (Negative): High precision (0.78) but lower recall (0.64), indicating missed instances of negative sentiment.
- Class 1 (Neutral): Good recall (0.74) but slightly lower precision (0.67), indicating more false positives.
- Class 2 (Positive): Strong precision (0.76) and recall (0.79), showing the model's ability to correctly detect positive sentiment.

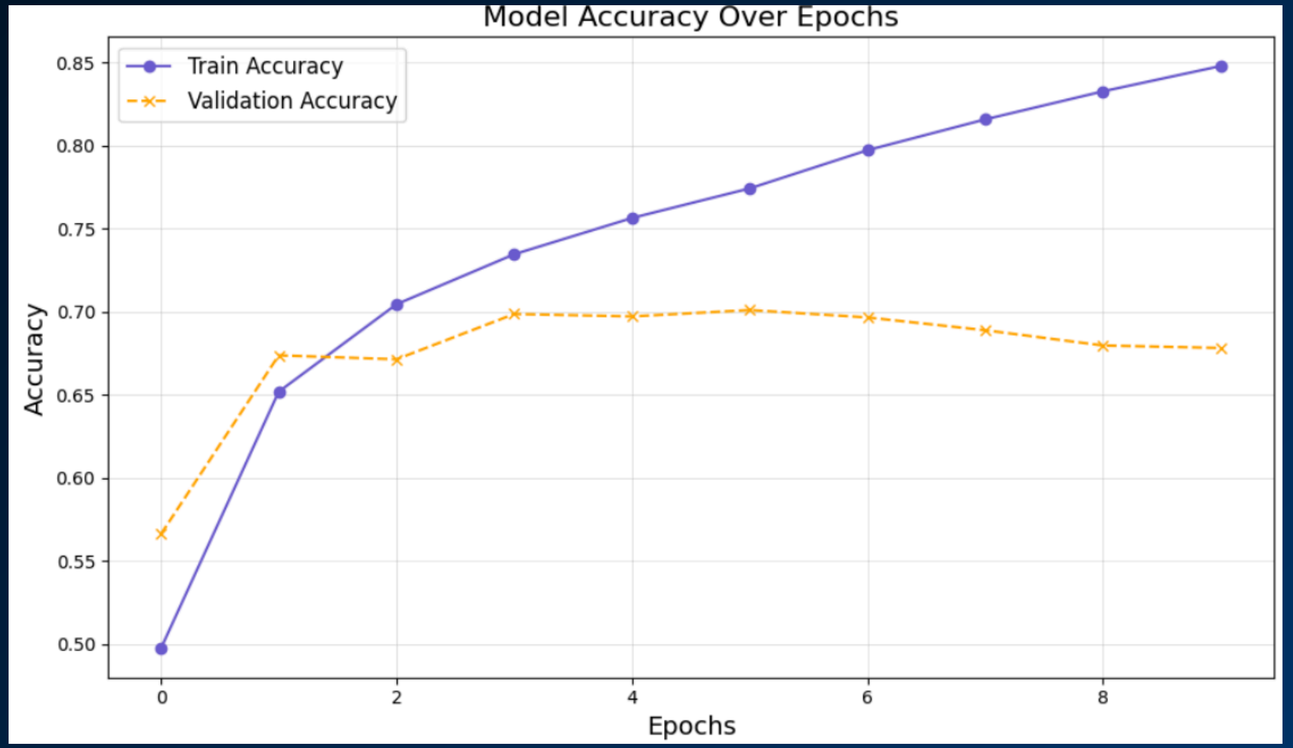
LEARNING CURVE ANALYSIS: ACCURACY

- **Training vs Validation Accuracy:**

The learning curve shows that training accuracy improves steadily, reaching 85%, while validation accuracy peaks at 70% before leveling off.

- **Overfitting:**

The gap between training and validation accuracy suggests that the model may start to overfit after a few epochs, performing well on training data but less so on unseen data.



LEARNING CURVE ANALYSIS:
LOSS

- **Training Loss:**

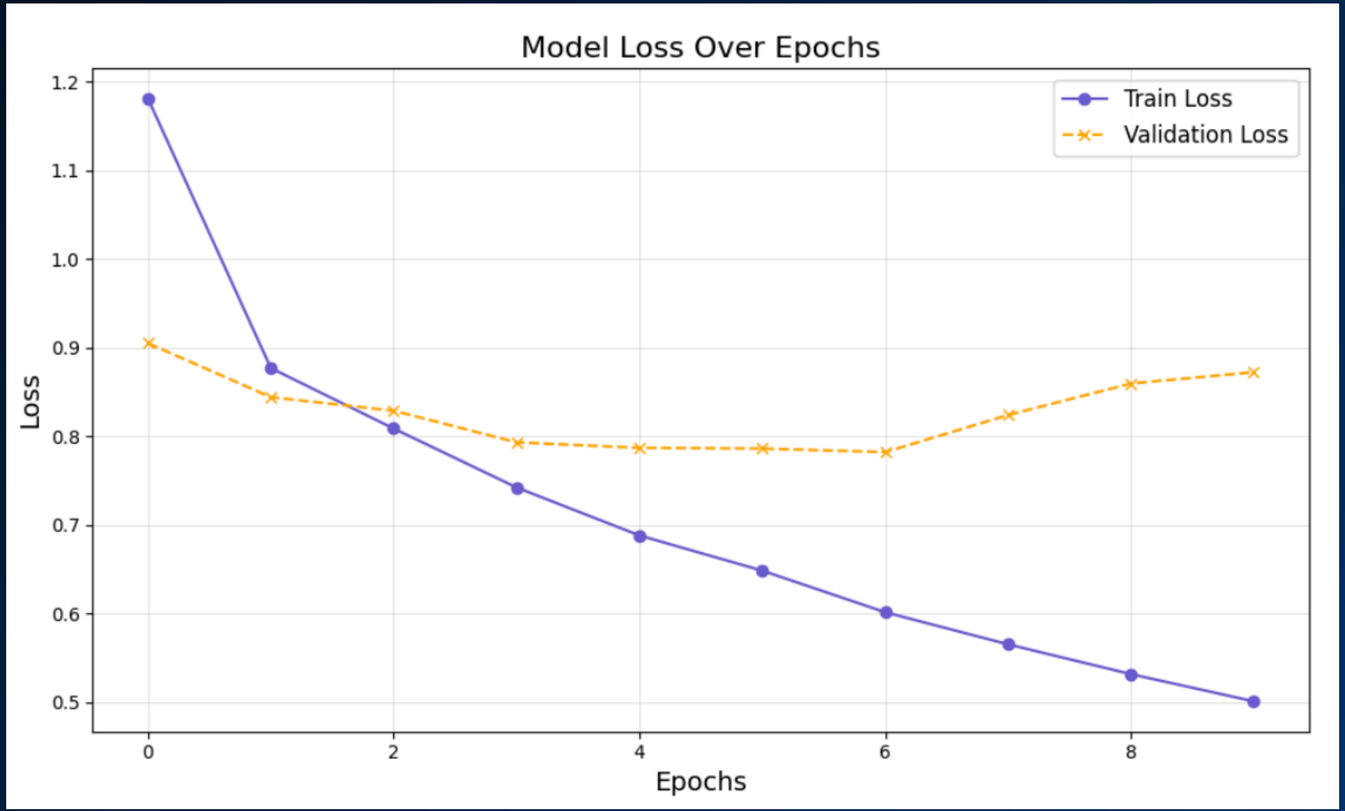
The training loss steadily decreases as the model becomes better at minimizing errors on the training data.

- **Validation Loss:**

The validation loss initially decreases but starts increasing after a few epochs, indicating that the model's performance on unseen data worsens after a certain point.

- **Overfitting Indication:**

This increase in validation loss, combined with a leveling of validation accuracy, suggests the model starts overfitting after a few epochs.



INSIGHTS, CHALLENGES & SOLUTIONS

Key Insights:

- Positive and negative sentiment: The LSTM model was highly effective at identifying positive and negative sentiments.
- Neutral sentiment: Distinguishing neutral sentiment was challenging due to the informal and brief nature of tweets.

Challenges Encountered:

- Overfitting: The model showed better performance on training data than evaluation data, suggesting overfitting.
- Class Imbalance: The dataset was imbalanced, with more neutral tweets, skewing the model's predictions.

Solutions Applied:

- Dataset Adjustment: Discarded the larger, imbalanced dataset to reduce bias.
- Indexing Validation: Tried using "selected text" but found the "index" method performed better.
- Hyperparameter Tuning: Adjusted batch size, dropout rates, and the number of neurons, which improved the model’s performance.

FUTURE WORK & MODEL ENHANCEMENTS

- **Model Refinement:**

Improve F1-scores for Class 0 (negative) and Class 1 (neutral) by addressing class imbalance and refining feature engineering.

- **Introducing Transformers:**

Implement a transformer-based model like BERT to better capture context and subtle nuances, such as sarcasm.

- **Expansion to Other Platforms:**

Expand the sentiment analysis to other social platforms like Facebook and Instagram for broader sentiment understanding across different networks.

TEAM MEMBERS & ROLES

This project was completed by four part-time MSc students in Business Analytics, combining their professional experience with their technical and strategic skills to ensure the success of the project.

GEORGIOS MARKAKIS

Role: Business Strategist, Project Manager, and Data Collection Lead

Background: Growth & Performance Marketing Manager at PeopleCert.

Focus: Ensured that the project aligned with business goals like feedback analysis and social monitoring.

EMMANOUIL MOSCHOGIANIS

Role: Model Developer and NLP Engineer

Background: Public Tender Consultant at TCS Toolbox.

Focus: Data preprocessing (tokenizatio and normalization) and preparing the dataset for sentiment analysis.

CHRISTOS NINOS

Role: Risk and Data Processing Specialist

Background: Underwriter at Allianz European Reliance.

Focus: Evaluated model performance using accuracy and precision metrics, ensuring minimal bias in results.

VASILIKI VAMVAKA

Role: Technical Lead and Model Developer

Background: Junior Developer at Adzuna

Focus: Managed tasks such as stopword removal and text cleaning, working alongside Manos in developing the model..

TIMELINE

The project followed a structured timeline to ensure the timely completion of each key phase, with milestones set for each week.

WEEKS 1–2: DATA COLLECTION AND EXPLORATION

- Gathered relevant datasets from Kaggle.
- Initial exploration of data to assess its suitability for sentiment analysis.

WEEKS 3–4: DATA PREPROCESSING

- Applied data cleaning, tokenization, normalization, and stopword removal.
- Ensured the dataset was fully prepared for model training

WEEKS 5–6: MODEL DEVELOPMENT AND TRAINING

- Developed the LSTM model.
- Ran multiple iterations to optimize accuracy in classifying sentiments.

WEEK 7: MODEL EVALUATION AND REFINEMENT

- Evaluated the model using metrics like accuracy, precision, and recall.
- Refined the model based on the evaluation results to improve its performance.

WEEK 8: FINAL REPORT AND PRESENTATION

- Compiled the final project report.
- Presented the results, and completed all deliverables on time.

TEAM DELTA

THANK YOU

FOR YOUR ATTENTION