Joaquin Berriel Martins Report
Twitter Sentiment Analysis Using Classical & Neural Models

1. Problem Statement

Sentiment analysis on social media data is a crucial task for understanding public opinion and user behavior. This project aims to classify tweets from the Kaggle "Twitter Sentiment Dataset" into negative (-1), neutral (0), and positive (+1) sentiments. The challenge lies in handling large-scale textual data efficiently while comparing classical machine learning approaches with a neural network model (CNN–LSTM).

2. Research Questions

1. Which model achieves the best overall and per-class performance for predicting sentiment?
2. How do TF-IDF and Word2Vec features compare when used with classical models?
3. Does a CNN–LSTM hybrid outperform classical ensembles (Voting Classifier) on this dataset?

3. Dataset

- **Source:** Kaggle — Twitter Sentiment Dataset (Saurabh Shahane, 2021)
- **Columns:** clean_text (string), category (int: -1, 0, 1)
- **Size:** 162,980 tweets

4. Methods

1. **Exploratory Data Analysis (EDA) & Preprocessing**
    a. Class distribution, text length analysis, missing data handling
    b. Tokenization and stopword removal
    c. Sampling: due to memory constraints, we experimented with different sample sizes (1,000; 15,000; 20,000) to train models effectively
2. **Feature Engineering**
    a. **TF-IDF Features:** Classical baseline models
    b. **Word2Vec Features:** Average tweet embeddings for classical models
    c. **Neural Network Features:** Keras tokenizer with padding, feeding into CNN–LSTM
3. **Models**
    a. Classical: Decision Tree, KNN, Logistic Regression
    b. Ensemble: Voting Classifier combining best-performing classical models
    c. Neural: CNN–LSTM hybrid (embedding → convolution → LSTM → dense layers)
    d. **Best performing model:** CNN–LSTM
4. **Evaluation**
    a. Metrics: accuracy, precision, recall, macro F1, per-class F1

5. Work Organization

- **Joaquin:** EDA & preprocessing, main pipeline, CNN–LSTM implementation
- **Emin:** TF-IDF feature engineering
- **Will:** Word2Vec feature engineering

Workflow: At the start, the team defined a clear structure and split tasks. Each member worked separately due to different schedules and other project commitments. Weekly meetings during the Data Science lecture were used to check progress, coordinate next steps, and resolve issues.

6. Reflexion

My core role focused on setting the foundation for the analysis through Exploratory Data Analysis (**EDA**) and **data preprocessing**, and the subsequent implementation of the **Neural Network** pipeline (CNN–LSTM).

From a technical perspective, implementing the **CNN–LSTM architecture** was the most challenging component. Early in the process, the dataset's **class imbalance** (specifically the over-representation of Positive tweets) severely skewed initial results, leading to demotivating, **near-random F1-Macro** scores. Addressing this required careful data balancing and sampling which, while eventually successful, introduced a new challenge: training complex models like the CNN–LSTM on small samples led to severe overfitting until we scaled up to 25,000 records.

Beyond the code, team collaboration presented its own complications. It was nearly impossible to organize and meet with the rest of the team due to scheduling and workload differences. Furthermore, a specific technical obstacle arose with the classification output: some downstream data models did not properly accept the standard {-1, 0, 1} sentiment labels, requiring me to implement a crucial correction by incrementing the labels by +1 during the pipeline flow to ensure compatibility.

Despite these challenges, we successfully established a robust and modular pipeline capable of reproducible experiments, culminating in the CNN-LSTM achieving the project's best performance.

7. Conclusion and Future Work

The **CNN-LSTM hybrid model achieved the best overall performance**, demonstrating a robust F1-Macro score of 0.8663 for complex multiclass sentiment classification.
For classical models, **TF-IDF features were substantially superior to Word2Vec features**, confirming TF-IDF's power in isolating discriminative keywords in social media text.
Ultimately, the **deep learning** (CNN-LSTM) **approach outperformed all classical ensembles**, proving that sequence models are necessary for capturing the nuances and context of Twitter data.

This project was a real journey! We started with some seriously frustrating, almost random results, which was a huge motivational dip. Thankfully, realizing that we just needed to throw more data at the problem. In the end, we proved that the cutting-edge stuff really works: our CNN-LSTM hybrid model completely crushed the traditional approaches, hitting a nice F1-Macro score. While the simple Logistic Regression with TF-IDF held its own as the classical champ, it was clear the neural network's ability to capture the subtle nuance in Twitter language was the decisive factor.

It was a great lesson in how much feature engineering matters, and how sometimes you just need a bigger hammer (or, in this case, a deeper network and more data) to get the job done.

To take these results from "great project" to "deployment-ready model," here’s where the focus should shift next:

- **Final Data Rerun**: The most critical step is to execute the entire winning pipeline on the full 162,980 dataset. This is necessary to maximize the model's generalizability and provide the final, most accurate performance metrics for the real world.
- **External Embedding Evaluation**: We need to investigate why our corpus-trained Word2Vec embeddings failed so badly. The next logical move is to integrate and test powerful, pre-trained word vectors like GloVe or fastText. These embeddings have seen billions of words and should offer the semantic quality needed to finally compete with, or even surpass, TF-IDF.
- **Targeted Error Analysis**: Now that we have a strong baseline, we should dive deep into the CNN-LSTM's confusion matrix. We can perform a granular error analysis to identify specific failure modes, particularly for the Neutral class, and target those weaknesses to squeeze out the final few percentage points of performance.