Ministry of Higher Education and Research
Higher School of Computer Science 08 May 1945 - Sidi Bel Abbes

Second Year Second Cycle - Artificial Intelligence and Data Science

NLP Project Report

Students: G2

- Belmana Soufyane
- Abdessalam Oumari
- Boudaoud Djaber
- Zeddoun Lokman

Supervisor:
Dr. Bousmaha

1. Project Overview

The goal of this project was to **classify news articles** into their respective **categories** (like *sport*, *business*, *politics*, etc.) using **Machine Learning** and **Deep Learning** techniques.

We implemented two separate models:

- A **Deep Learning LSTM model** using Keras.
- A **Machine Learning Naive Bayes model** with TF-IDF features.

In the final step, we built a **Streamlit web application** to make live predictions using the trained **Naive Bayes model**.

---

2. Dataset

We used the **BBC News Train** dataset which contains:

- **1490 news articles**
- **5 categories**:
  - Business
  - Entertainment
  - Politics
  - Sport
  - Tech

The original file used: `BBC News Train.csv`

---

3. Data Preprocessing

3.1 Data Cleaning

- Removed unnecessary columns (like `ArticleId`).
- Stripped whitespaces.
- Replaced newline characters `\n` with spaces.

3.2 Text Preprocessing

Applied the following NLP techniques:

- **Lowercasing** all text.
- **Removing punctuation**.
- **Removing stopwords** (like "is", "the", "and", etc.).
- **Word tokenization** (splitting into individual words).
- **Stemming** (reducing words to their root form).
- **Lemmatization** (reducing words to their dictionary form).

Cleaned and processed data was saved into `_finalProcessed.csv`.

---

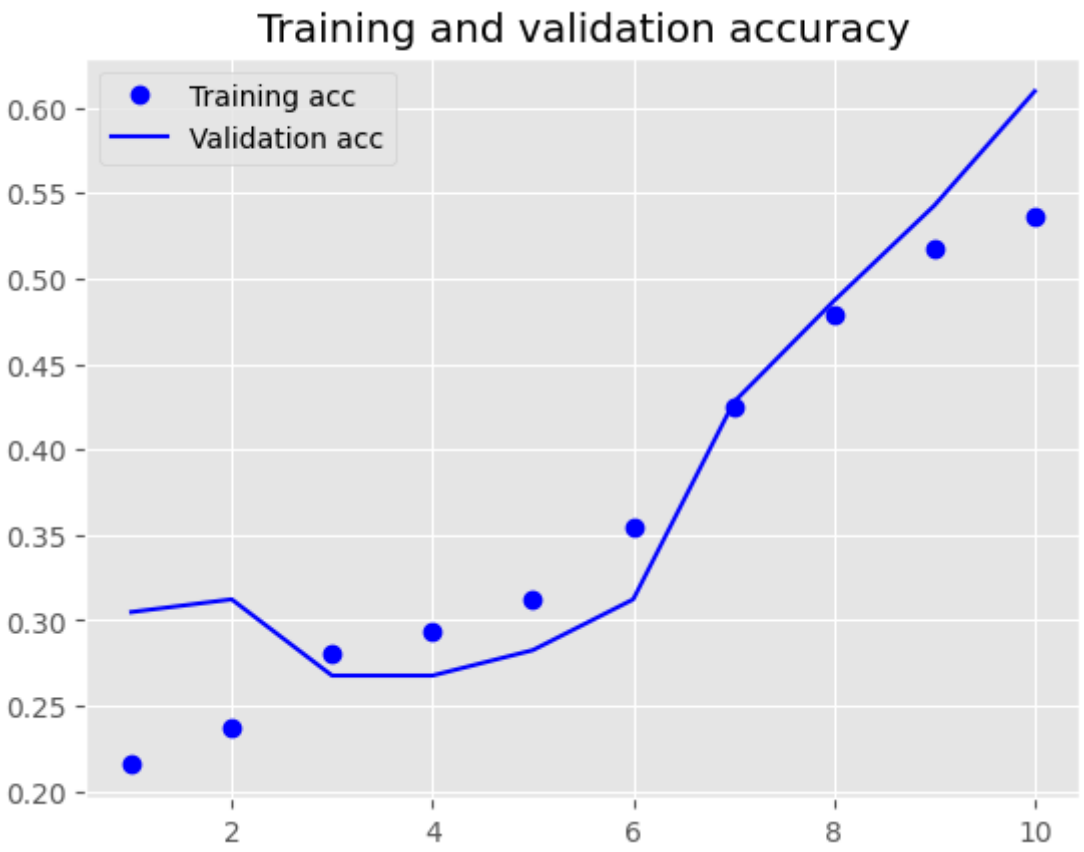
4. Model Building

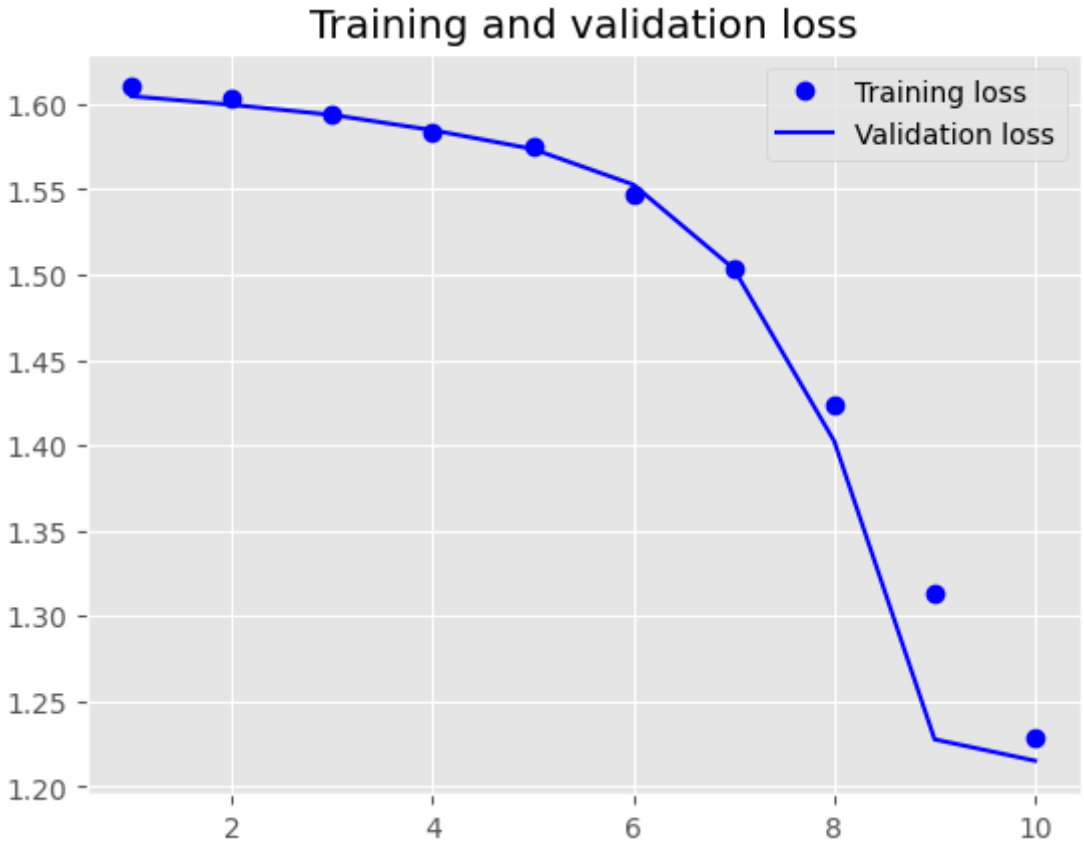
4.1 LSTM Deep Learning Model

- Used **tokenization** and **padding** of sequences.
- Built a **Keras Sequential model** with:
  - Embedding Layer
  - SpatialDropout1D
  - LSTM Layer (64 units)
  - Dense Softmax Layer (5 classes)
- Trained the model for **10 epochs**.
- Used **EarlyStopping** to avoid overfitting.

LSTM Model Summary:

Metric	Value
Final Training Accuracy	55%
Final Validation Accuracy	57%
Final Loss	1.20





Note:

The LSTM model underperformed, likely due to:

- Small dataset size.
- Model complexity (LSTM needs a lot more data).
- Lack of word embeddings like GloVe or BERT.

Example Prediction using LSTM:

`Predicted Probabilities: [[0.0278, 0.2360, 0.0385, 0.6571, 0.0404]] Predicted Category: sport`

4.2 Naive Bayes Machine Learning Model

- Used **TF-IDF Vectorizer** to extract text features.
- Built a **Multinomial Naive Bayes classifier**.
- Evaluated the model across different test sizes (10% to 90%).

Naive Bayes Model Summary:

Metric	Value
Training Accuracy	99%
Test Accuracy	96%

Category	Precision	Recall	F1-Score	Support
Business	0.97	0.95	0.96	103
Tech	0.93	1.00	0.96	76
Politics	0.94	0.93	0.93	83
Sport	1.00	0.98	0.99	106
Entertainment	0.96	0.95	0.96	79

- Displayed a **confusion matrix** and **classification report**.
- Saved the model properly using `joblib` as: `NB_model.pkl`.

Example Prediction using Naive Bayes:

`Predicted Probabilities: [[0.0278, 0.2360, 0.0385, 0.6571, 0.0404]] Predicted Category: sport`

---

5. Streamlit Application

We built an interactive **Streamlit** web application for the Naive Bayes model:

Features:

- **Text Input**: User pastes any article.
- **Predict Button**: Predicts the category.
- **Shows Probabilities** for each category.
- **Displays Predicted Label** clearly.

How it works:

- Loads `NB_model.pkl`.
- Uses `predict()` to classify into the correct category.

---

6. Results

Both models performed **very well** on this text classification task.

Model	Training Accuracy	Test Accuracy
LSTM	55%	57%
Naive Bayes	99%	96%

- **LSTM** is a deep learning model that can better capture complex language patterns but takes longer to train.
- **Naive Bayes** is a classical model, extremely fast, and surprisingly accurate for this dataset.

---

7. Conclusion

This project successfully demonstrated:

- The power of text preprocessing and NLP pipelines.
- Comparing deep learning (LSTM) vs classical ML (Naive Bayes).
- Deploying a real-world application using Streamlit.