جامعة الأخوين
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵏ ⵓⵅⵡⴰⵢⵏ
AL AKHAWAYN UNIVERSITY

CSC3348

Assignment 1

Sentiment Analysis using Logistic Regression

Under the supervision of:

- Dr. Asmae MOURHIR
- Ms. Aya Lyousfi

By:

- Khawla CHRIFI ALAOUI < 165466 >
- Rihab ZOUITNI < 100552 >
- Salma SAMINE < 85492 >

~ Spring 2025 ~

Table of Contents

List of Figures

Introduction

In this first assignment, we had the opportunity to build a logistic regression model from scratch using Python, and we have worked on a sentiment analysis dataset that we have split into training and validation. Lastly, we did an evaluation on the testing set to get the accuracy of our predictive model.

Preprocessing steps

- ✓ Dropping unnecessary columns: 'desc_id', 'source_name', 'review_text', and 'is_accessibility'.
- ✓ Dropping rows where the 'vader sentiment' column is 'neutral'.
- ✓ Oversampling to get balanced data.
- ✓ Applying Vader Lexicon.
- ✓ Converting each review into a numerical vector.
- ✓ Converting sentiment to binary (positive -> 1, negative -> 0).

Results with balanced data

During our preprocessing, we have tried to balance our data using the oversampling method “RandomOverSampler” that exists in the “imblearn.over_sampling” library, and we got the following results:

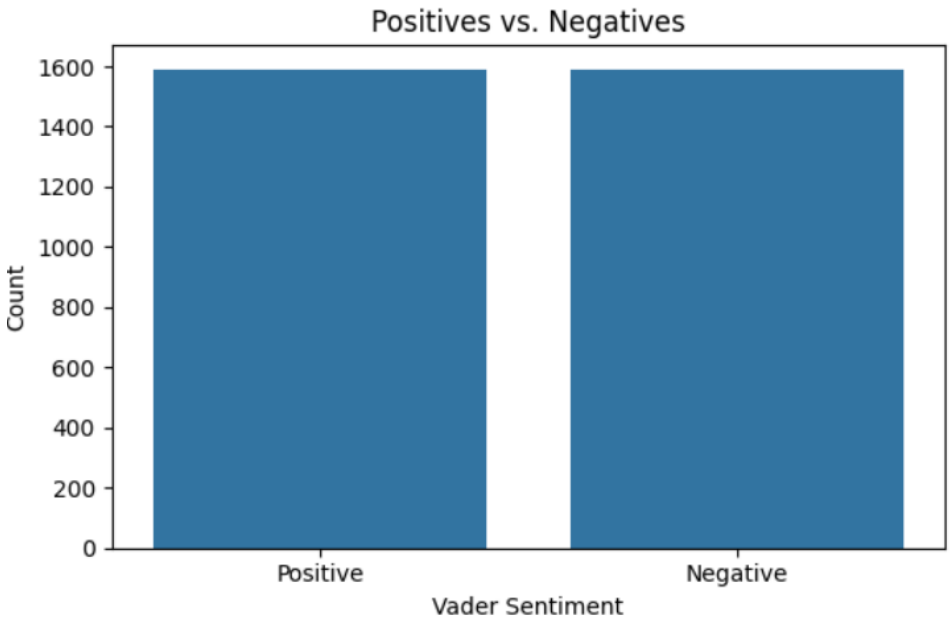

Figure 1: Visualization of the Balanced Data

The best results we could reach were with **8500 epochs** and a **learning rate of 0.7**:

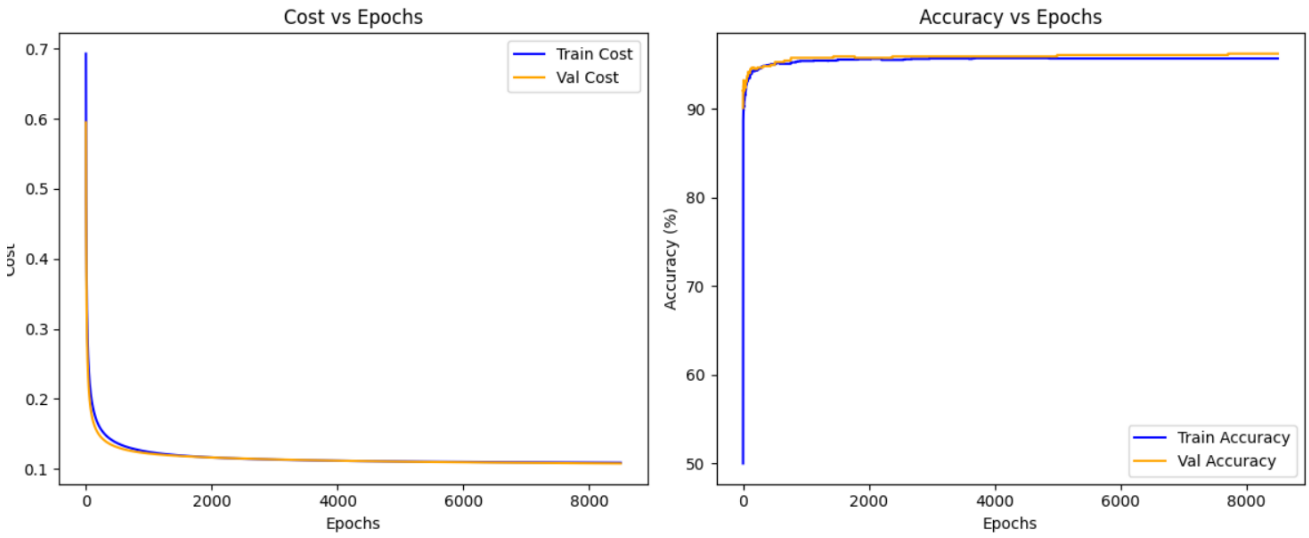

Figure 2: Cost and Accuracy Diagrams with Balanced Data

```
print(f"Final Training Cost: {cost_history_train[-1]:.4f}")
print(f"Final Validation Cost: {cost_history_val[-1]:.4f}")
print(f"Final Training Accuracy: {accuracy_history_train[-1]:.4f}%")
print(f"Final Validation Accuracy: {accuracy_history_val[-1]:.4f}%")
```

```
Final Training Cost: 0.1086
Final Validation Cost: 0.1075
Final Training Accuracy: 95.6761%
Final Validation Accuracy: 96.2264%
```

Figure 3: Final Global Results with Balanced Data

After saving our model and evaluating it, our submission on Kaggle gave us an accuracy of **0.95959.**

Results with unbalanced data

Link:

https://alakhawayn365-my.sharepoint.com/:f:/g/personal/r_zouitni_aui_ma/EvrcvOJ1lEpIt5I23uRElFkBNTeoJRkFtj1phoQ5RBv3UQ?e=m20Mfx

We wanted to try working on the model in another notebook to see what results we might get with unbalanced data by removing the oversampling part (, and we got the following results:

For **1700 epochs** and a **learning rate of 0.07**, we had the best result as showed in the figure below:

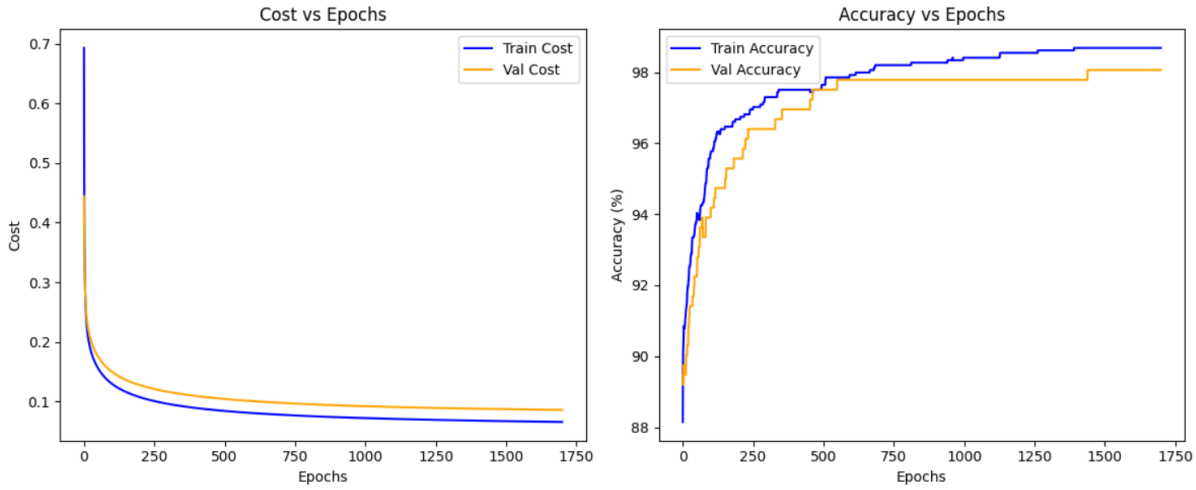

Figure 4: Cost and Accuracy Diagrams with Unbalanced Data

```
print(f"Final Training Cost: {cost_history_train[-1]:.4f}")
print(f"Final Validation Cost: {cost_history_val[-1]:.4f}")
print(f"Final Training Accuracy: {accuracy_history_train[-1]:.4f}%")
print(f"Final Validation Accuracy: {accuracy_history_val[-1]:.4f}%")
```

```
Final Training Cost: 0.0765
Final Validation Cost: 0.0965
Final Training Accuracy: 98.1982%
Final Validation Accuracy: 97.7839%
```

Figure 5: Final Global Results with Unbalanced Data

After saving our model and evaluating it, our submission on Kaggle gave us an accuracy of **0.97306.**

Conclusion:

This sentiment analysis assignment has allowed us to build a model using logistic regression and trying out two different ways of predicting: one with complete preprocessing steps including balanced data, and another one with unbalanced data.

Results has shown that the unbalanced data model gave us a higher accuracy than with balanced data. Maybe our oversampling did not generate a lot of accurate training examples, and this what can explain the difference in the results we got.

N.B:

We have also tried to use these techniques to get better results:

- L2 regularization.
- L2 regularization with early stopping.
- Adam optimization technique.
- Adam optimization technique with learning rate decay.

However, we did not notice high improvements, so we have decided to keep our logistic regression simple.