MULTILINGUAL SENTIMENT ANALYSIS SYSTEM (ARABIC & ENGLISH) USING XLM-ROBERTA

Problem

- This project builds a **multilingual sentiment analysis system** that works with **Arabic and English** text.
- The model classifies text into **Negative, Neutral, or Positive sentiment.**
- In real life, many people do not write in one language only. They often write in Arabic and mix English words or sentences in the same text. Most traditional sentiment models cannot handle this **mixed-language text** well.
- To solve this problem, a multilingual transformer model **(XLM-RoBERTa)** was used. The model was **trained** on both **Arabic and English** datasets. To improve Arabic performance, **Back Translation** was applied as a **data augmentation technique**, which helped **increase data diversity and improve generalization.**
- The final model achieved **about 80% accuracy** and showed strong performance, especially for positive and negative sentiment.
- The system was deployed using **FastAPI and Docker**, making it ready for real-world use as an API.

Example:

هذا المنتج ممتاز جدًا , i really like it

- This is called code-switching (mixing languages in one sentence).
- Many sentiment models fail to understand this type of text.
- Arabic is already complex, and mixing it with English makes the problem harder.
- We need a model that can understand:
  - Arabic text
  - English text
  - Arabic and English together in the same sentence

Why This Problem Is Important

- People use mixed languages on:
- Social media
- Product reviews
- Comments and chats
- Real users do not follow language rules.
- A real sentiment analysis system must work with mixed-language text.
- This project solves this problem using a multilingual transformer model.

Data Size and Data Sources

This project was trained on large datasets in both Arabic and English to build a strong multilingual sentiment analysis model.

Arabic Dataset

- Dataset name: Arabic 100K Reviews
- Number of samples: 100,000 reviews
- Language: Arabic
- Source (Kaggle):
- https://www.kaggle.com/datasets/abedkhooli/arabic-100k-reviews/data

English Dataset

- Dataset name: Amazon Fine Food Reviews
- Number of samples used: ~500,000 reviews
- Language: English
- Source (Kaggle):
- https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews

Arabic Dataset Augmentation Using Back Translation

Back Translation was used to augment the **Arabic dataset** by translating text to English and back to Arabic, preserving meaning while varying wording. This increased data size, reduced class imbalance, maintained clean and natural data, and improved model generalization.

Back Translation Strategy

- From the Arabic dataset:
  - 30,000 short sentences (15 words or less) were selected
  - Short sentences are easier to translate and produce less noise
- 50% of these sentences were randomly selected:
  - 15,000 sentences
- Each sentence generated one new augmented sentence

Final Arabic Dataset Size

- Original Arabic samples: 100,000
- Back-translated samples: +15,000
- **Final Arabic samples: 115,000**

English Data Labeling and Balanced Sampling

The original English dataset contains about **500,000 reviews.**
Using all English data would make the model biased toward English.
To avoid this:

- **Review ratings** were converted into **sentiment labels (Negative, Neutral, Positive).**
- A random sample of **150,000 reviews** was selected.
- The sample was balanced across sentiment labels.

This approach helps the model to:

- Learn Arabic and English more fairly
- Avoid dominance of English data
- Improve multilingual performance
- Ensure stable training across all sentiment classes

Dataset After English Sampling and Arabic Back Translatio

LanguageNumber of Samples

- Arabic (Augmented)
- 115,000
- English (Selected)
- 150,000

Text Cleaning Strategy

- Only basic text cleaning was applied.
- I did not remove stop words.
- I did not remove punctuation.

Why Stop Words and Punctuation Were Kept

- XLM-RoBERTa is a transformer-based model.
- It uses subword tokenization and contextual embeddings.
- Stop words and punctuation:
- Help the model understand context
- Can change the meaning and sentiment of a sentence
- Are important for sentence structure

Example

- “This product is not good”
- Removing not would change the sentiment.
- Punctuation can also affect meaning and emotion.

Cleaning Steps Applied

- Text normalization
- Removing extra spaces
- Removing unnecessary special characters
- Keeping stop words and punctuation
- Tokenization handled by XLM-RoBERTa tokenizer

Model Evaluation & Performance

The model achieved an **accuracy close to 80%**, showing strong performance for a multilingual sentiment classification task.
A classification report was used to evaluate **precision, recall, and F1-score** for each sentiment class.
The results show **balanced performance,** with slightly lower scores for the Neutral class.
A confusion matrix was used to visually analyze prediction errors and class confusion.
Together, these metrics provide a reliable and complete evaluation of the model.

Class	Precision	Recall	F1-score
Negative	0.82	0.79	0.8
Neutral	0.64	0.65	0.64
Positive	0.87	0.88	0.88

UI Overview

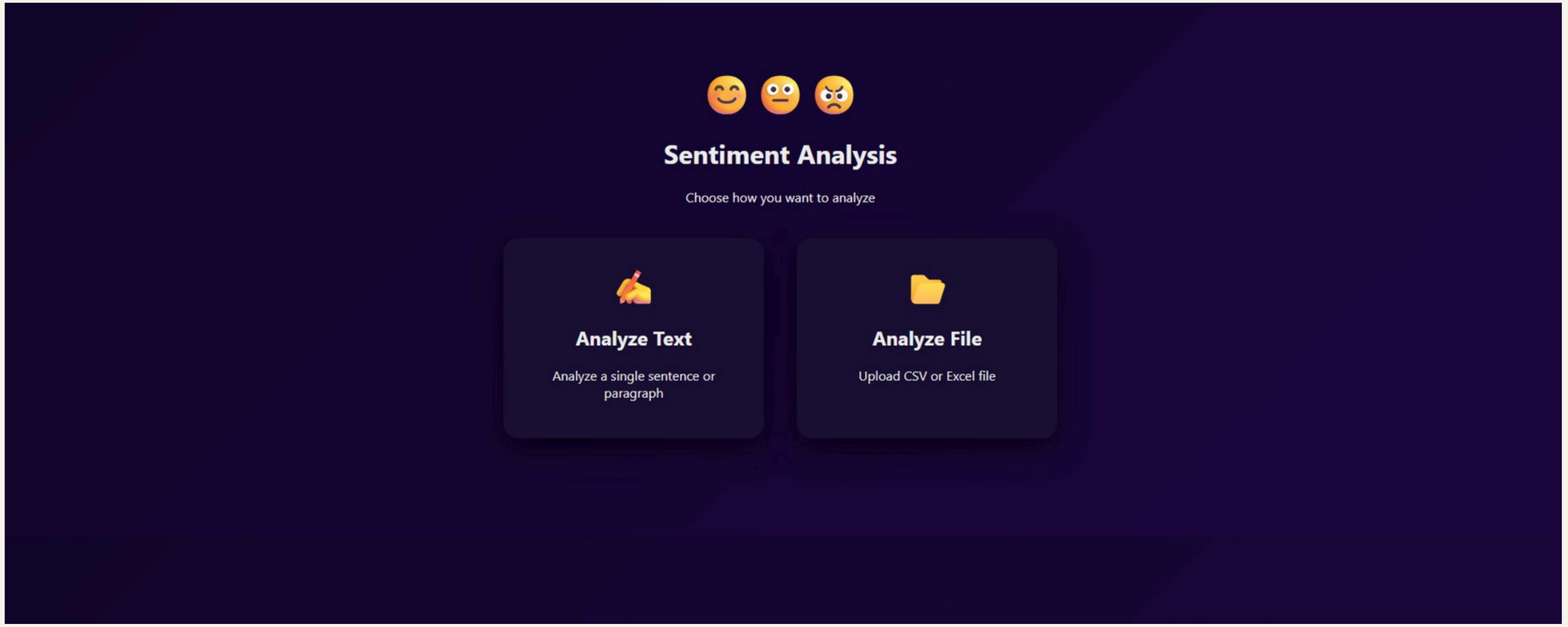
The user interface was designed to be simple, clean, and user-friendly,
allowing users with no machine learning or technical background to easily interact with the sentiment analysis system.

The UI guides the user through a clear workflow, starting from choosing the analysis **type (text or file),**
submitting the input, and finally **receiving and downloading** the sentiment results.

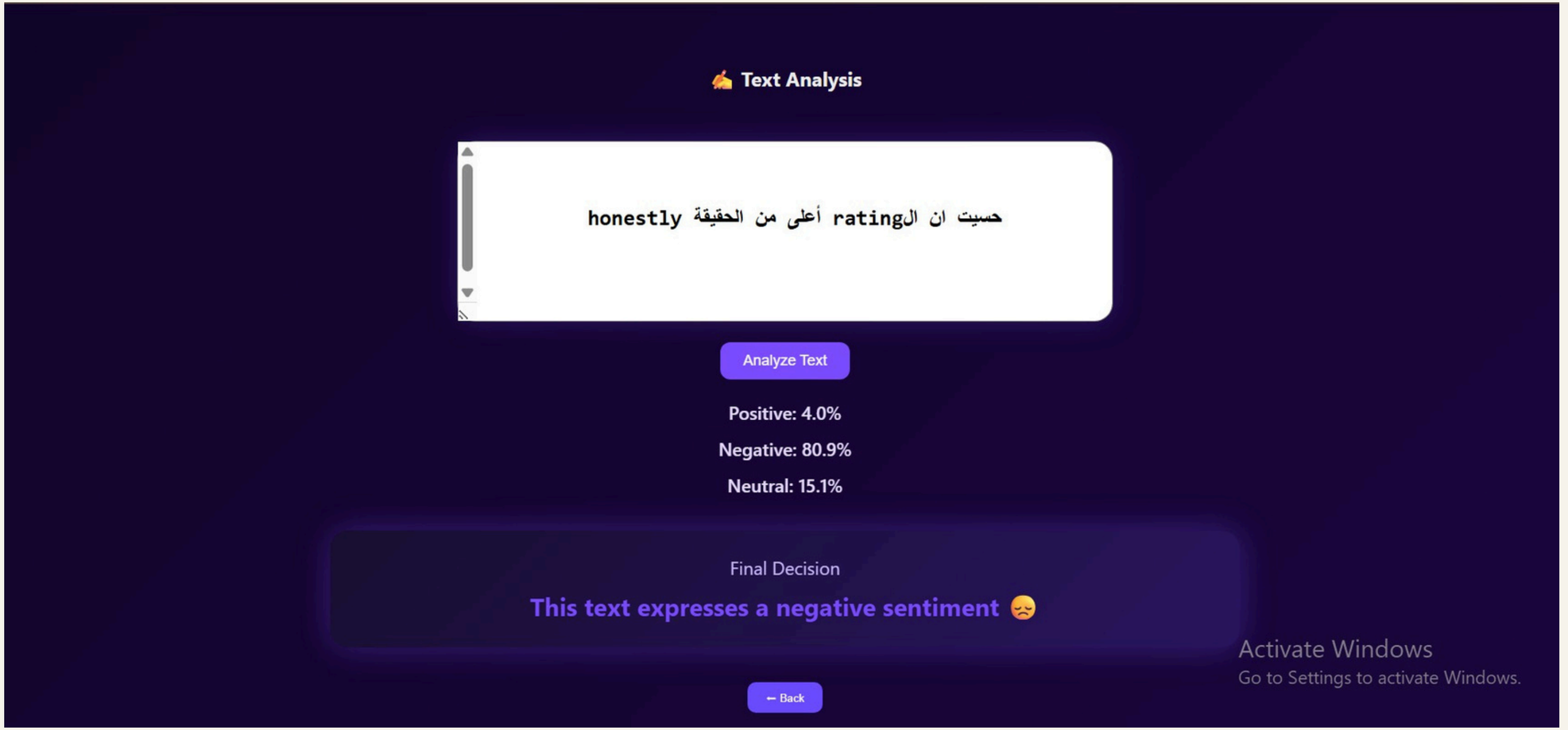
The interface focuses on clarity and usability, hiding the complexity of the backend,
while providing fast responses and visual feedback through **charts and summaries**.

Overall, the UI acts as a **bridge between the user and the machine learning model,**
making advanced sentiment analysis accessible to everyone

1. the user can choose how they want to analyze the data
   analyzing a single text or uploading a file for batch analysis.

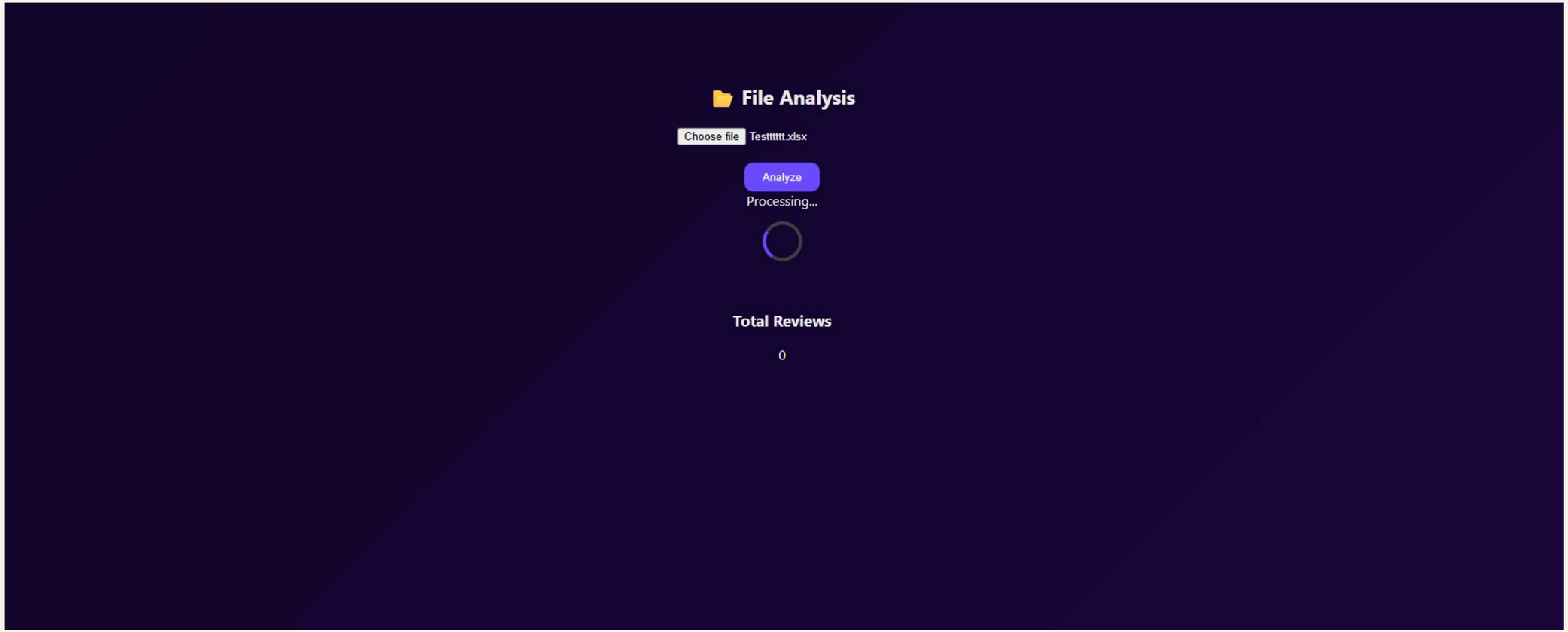


2- Text input + Analyze button



3- Upload CSV / Excel

The user uploads a file that contains a single column called **text.**



Large files are processed in the background using Celery, so the user does not have to wait on the page

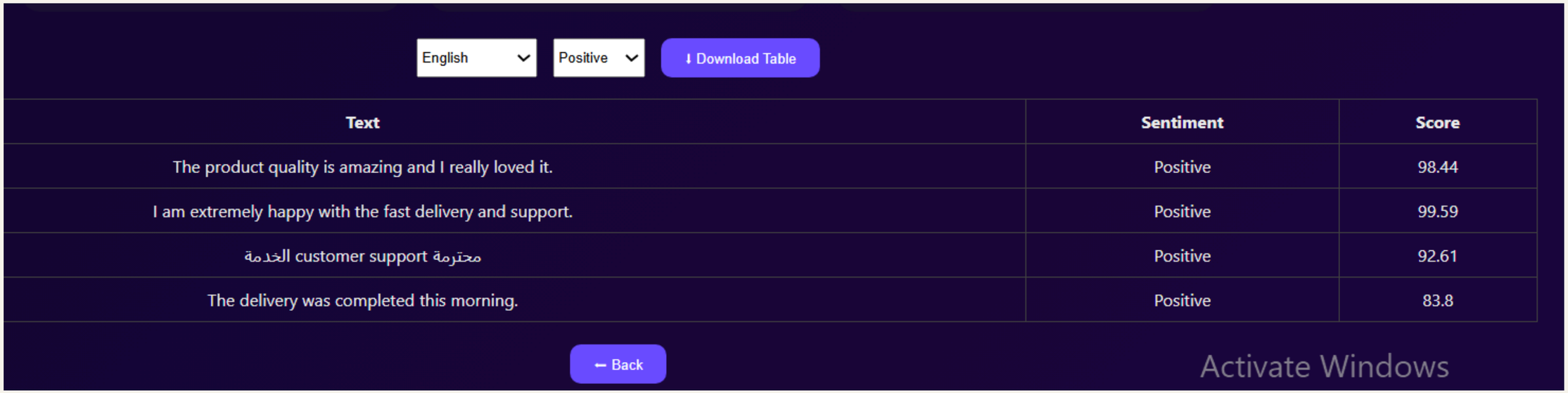
A separate word cloud is generated for each sentiment label, showing the most frequent words. The larger the word appears, the more frequently it occurs within that label.

Text	Sentiment	Score
المنتج ممتاز وجودته عالية	Positive	94.77
التوصيل كان سريع والتعامل محترم	Positive	91.31
السعر مرتفع مقارنة بالجودة	Neutral	59.32
أنا سعيد جدًا بالشراء من هذا المكان	Positive	95.55
لم يعجبني المنتج إطلاقًا	Negative	99.55
The product quality is amazing and I really loved it.	Positive	98.44

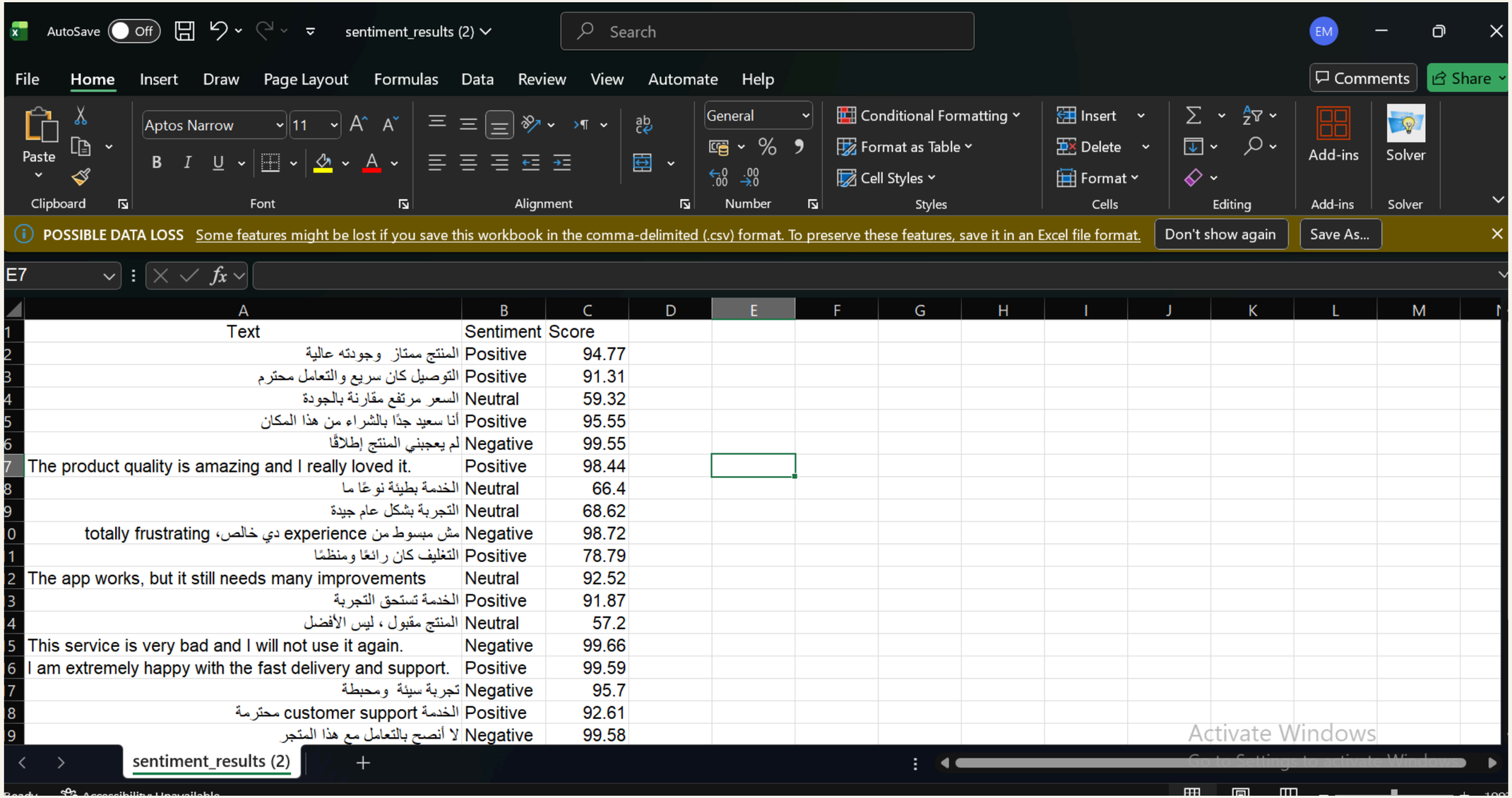
The system **automatically analyzes each text** entry and **generates** the corresponding **sentiment label and sentiment score.**

The results are displayed in a table where users can **filter the data** by **language** or **sentiment label.**

After reviewing the results, the user can **download** the processed file.



4- Downloaded Results Data



	A	B	C
1	Text	Sentiment	Score
2	المنتج ممتاز وجودته عالية	Positive	94.77
3	التوصيل كان سريع والتعامل محترم	Positive	91.31
4	السعر مرتفع مقارنة بالجودة	Neutral	59.32
5	أنا سعيد جدًا بالشراء من هذا المكان	Positive	95.55
6	لم يعجبني المنتج إطلاقًا	Negative	99.55
7	The product quality is amazing and I really loved it.	Positive	98.44
8	الخدمة بطيئة نوعًا ما	Neutral	66.4
9	التجربة بشكل عام جيدة	Neutral	68.62
10	مش مبسوط من experience دي خالص، totally frustrating	Negative	98.72
11	التغليف كان رائعًا ومنظمًا	Positive	78.79
12	The app works, but it still needs many improvements	Neutral	92.52
13	الخدمة تستحق التجربة	Positive	91.87
14	المنتج مقبول ، ليس الأفضل	Neutral	57.2
15	This service is very bad and I will not use it again.	Negative	99.66
16	I am extremely happy with the fast delivery and support.	Positive	99.59
17	تجربة سيئة ومحبطة	Negative	95.7
18	الخدمة customer support محترمة	Positive	92.61
19	لا أنصح بالتعامل مع هذا المتجر	Negative	99.58

Project Architecture

```
SENTIMENT_ANALYSIS/
│
├── .conda/                  # Virtual environment
├── Data/                    # Raw datasets (Arabic & English)
├── Data_Predictions/        # Saved prediction outputs
│
├── Docker/                  # Docker Configuration
│   ├── Dockerfile
│   ├── docker-compose.yml
│   ├── .dockerignore
│   ├── .env
│   └── .env.example
│
├── src/                     # Main source code
│   │
│   ├── app/                 # Core application logic
│   │   ├── inference.py     # Model inference logic
│   │   ├── model_loader.py  # Load XLM-RoBERTa model
│   │   └── schemas.py       # Request & response schemas
│   │
│   ├── frontend/            # Frontend interface
│   │   ├── templates/       # HTML pages
│   │   ├── static/          # CSS & JavaScript
│   │   └── summary.py       # Frontend summary logic
│   │
│   ├── helper/              # Helper functions
│   │   └── config.py        # Configuration & label mapping
│   │
│   ├── model/               # Trained models
│   │   ├── xlm_sentiment_model
│   │   └── arabic-english-sentiment-model
│   │
│   ├── notebook_files/      # Jupyter notebooks
│   │   ├── Cleaning & Augmentation
│   │   ├── EDA
│   │   └── Training
│   │
│   ├── routes/              # API endpoints
│   │   ├── analyze_text.py  # Text sentiment analysis
│   │   ├── analyze_file.py  # File sentiment analysis
│   │   ├── predict.py       # Prediction endpoint
│   │   ├── download.py      # Download results
│   │   └── task_result.py   # Async task results
│   │
│   ├── tasks/               # Background processing
│   │   └── tasks.py         # Celery tasks
│   │
│   ├── utils/               # Utility functions
│   │   ├── pie_chart.py     # Visualization utilities
│   │   └── wordcloud_utils.py  # Word cloud generation
│   │
│   ├── celery_app.py        # Celery configuration
│   └── main.py              # FastAPI entry point
│
├── .env                     # Environment variables
├── .env.example             # Example environment file
├── .gitignore               # Ignored files
├── README.md                # Project documentation
└── requirements.txt         # Python dependencies
```

Github-Link

multilingual-sentiment-analysis

Model Link

Arabic-English-sentiment-Model

System Demo

System-demo

THANK YOU

For your attention