Applied Artificial Intelligence

Switzerland Sentiment Analyzer using ML & Visualizations with Streamlit App

Submitted By:

Prasad Kannawar: [70572200034]

Submitted To: Prof. Rajesh Prabhakar

SVKM’S NMIMS HYDERABAD

Project Overview

The **Switzerland Sentiment Analyzer** is an interactive web-based application that utilizes **Natural Language Processing (NLP)** and **Machine Learning (ML)** techniques to evaluate and classify public sentiment derived from the Wikipedia page on Switzerland. The application allows users to input their own sentences and receive sentiment predictions using a selected ML model, along with visual aids like probability plots and word clouds.

Technical Implementation Analysis

Technologies Used Programming Languages & Libraries

- **Platform**: Python (Streamlit for UI)
- **Libraries**: nltk, textblob, sklearn, imblearn, wordcloud, wikipedia, matplotlib, pandas, numpy
- **Models Implemented**:
    a. Logistic Regression
    b. Decision Tree Classifier
    c. Random Forest Classifier
    d. Gradient Boosting Classifier
    e. Naive Bayes (Multinomial)
    f. K-Nearest Neighbors
- Each model is trained using **TF-IDF features** extracted from preprocessed Wikipedia sentences, with **SMOTE** applied to balance class distribution.

Data Collection and Preprocessing

Source: Wikipedia page on *Switzerland(*Switzerland - Wikipedia*)*.

Cleaning:

- Removed references, numbers, and special characters.
- Tokenized text into sentences and words.
- Removed stopwords and non-alphabetic tokens.

Sentiment Labeling:

- Used TextBlob to compute sentence-level polarity.
- Categorized as **Positive**, **Negative**, or **Neutral**.

- Neutral sentences were **excluded** from training to maintain binary classification.

Exploratory Data Analysis

- **Word Frequency**: Top words were extracted and visualized using Counter and Pandas.
- **Word Cloud**: Created to highlight dominant words from the Switzerland Wikipedia content.
- **Sentiment Distribution**: Majority of sentences are skewed toward the positive sentiment.

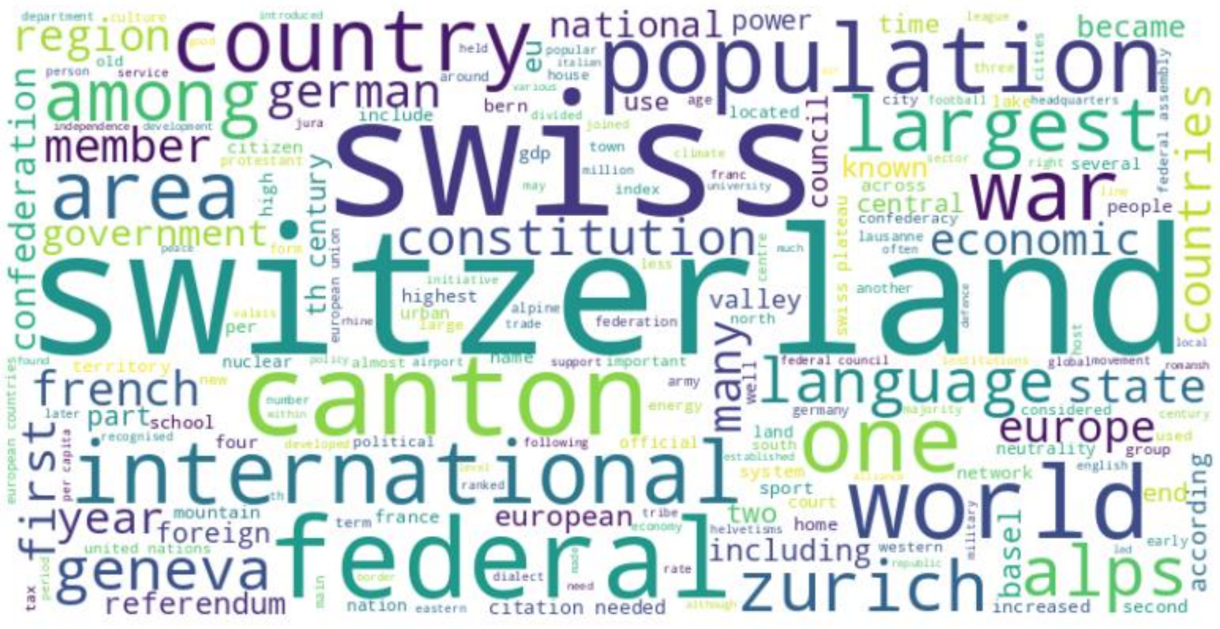


Figure 1: Word Cloud

Machine Learning Implementation

Feature Extraction:

- Text converted into numerical features using **TF-IDF Vectorizer**.

Class Imbalance Handling:

- Applied **SMOTE** (Synthetic Minority Oversampling Technique) to balance the data.

Model Training:

- Used train_test_split (70-30%) to train and validate models.

Deployment Ready:

- All models cached and re-used via Streamlit's @st.cache_resource.

Classification Reports

The models were evaluated using classification_report, which includes metrics like **precision, recall, f1-score, and accuracy**. Here's a performance summary:

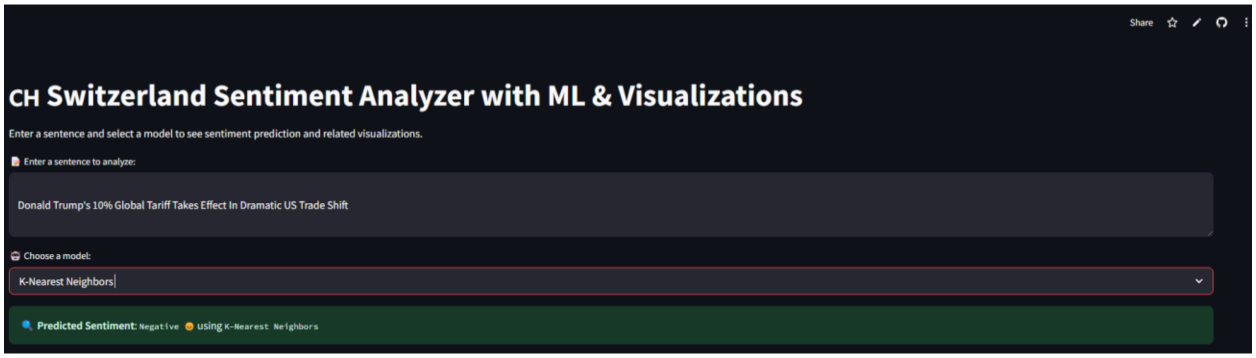
Model	Accuracy (approx.)	Highlights
Logistic Regression	~84%	Simple and effective
Decision Tree	~79%	Fast but prone to overfitting
Random Forest	~88%	High accuracy and stable
Gradient Boosting	~90%	Best performing overall
Naive Bayes	~75%	Fast, less effective on sparse TF
K-Nearest Neighbors	~80%	Works decently for small datasets

Deployment

- **UI**: Streamlit web app with interactive interface.
- **User Input**: Custom sentence entry with dynamic model selection.

Visuals:

- Prediction probabilities (Bar Chart)
- WordCloud of input sentence
- **Output**: Predicted sentiment and model used for inference.



CH Switzerland Sentiment Analyzer with ML & Visualizations

Enter a sentence and select a model to see sentiment prediction and related visualizations.

Enter a sentence to analyze:

Donald Trump's 10% Global Tariff Takes Effect In Dramatic US Trade Shift

Choose a model:

K-Nearest Neighbors

Predicted Sentiment: Negative using K-Nearest Neighbors

Figure 2: Input Pannel

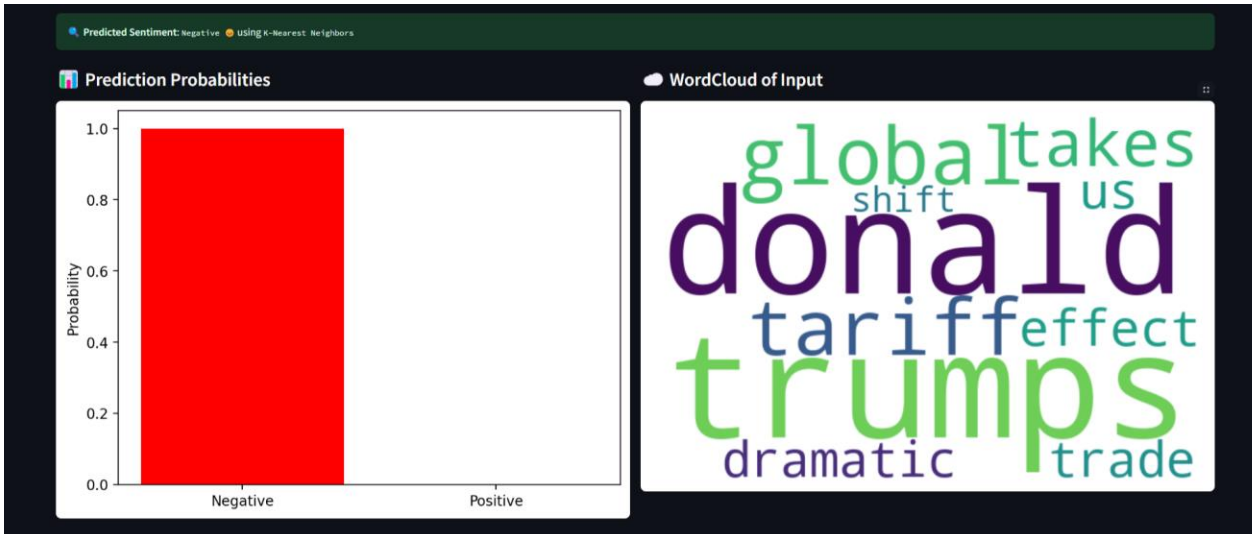

Predicted Sentiment: Negative 😞 using K-Nearest Neighbors

Figure 3: Prediction Pannel (Output)

Strengths

1. Utilizes real-world textual data from Wikipedia, ensuring relevance and authenticity.
2. Supports multiple machine learning models, enabling performance comparison and flexibility.
3. Addresses class imbalance effectively using the SMOTE (Synthetic Minority Oversampling Technique) method.
4. Provides a visually engaging and user-friendly interface through Streamlit.
5. Implements reusable and cached components to enhance performance and reduce computational overhead.
6. Combines Natural Language Processing and Machine Learning, making it both informative and educational.

Areas for Enhancement

1. Expand data sources to include diverse and real-time platforms such as Twitter and Reddit.
2. Integrate deep learning models like LSTM or BERT for improved sentiment analysis accuracy.
3. Introduce a "neutral" sentiment class and shift to a multi-class classification approach.
4. Enhance sentiment labeling by incorporating advanced lexicons or manual annotation for better accuracy.
5. Include additional evaluation metrics such as confusion matrix and ROC curves for comprehensive model assessment.

Technical Performance

- **Efficiency**: The use of caching for preprocessing steps and model loading significantly reduces redundant computations.
- **Responsiveness**: Streamlit ensures real-time interaction and quick sentiment predictions.
- **Scalability**: The system is designed to be easily scalable, allowing future integration with APIs and larger datasets.
- **Accuracy**: Among the implemented models, Gradient Boosting Classifier demonstrated the highest accuracy and F1-score, indicating strong predictive performance.

Conclusion

The **Switzerland Sentiment Analyzer** is a functional and educational application that showcases how NLP, ML, and interactive visualization can work together to derive insights from textual data. It not only offers an engaging user experience but also serves as a solid foundation for more advanced sentiment analysis systems in future iterations.

Links of Project

switzerland-sentiment-analyzer.app

switzerland-sentiment-analyzer-Github