Applied Artificial Intelligence

Sentiment Analysis on Sweden Using Web Scraping from Wikipedia

Submitted By:

Rachit Jain [70572200036]

Submitted To: Prof. Rajesh Prabhakar

SVKM'S NMIMS HYDERABAD
Deemed to-be UNIVERSITY

1. Project Overview

This project aims to analyze sentiment in textual data related to the Sweden Wikipedia page using machine learning techniques and deploy the trained model as an interactive web application using Streamlit. The system classifies user-input sentences into Positive or Negative sentiments based on their textual features.

The primary objective is to extract, preprocess, and classify textual data about Sweden obtained from sources like Wikipedia. The project utilizes TF-IDF vectorization to convert text into numerical features and employs machine learning classifiers to predict sentiment. The final model is deployed through Streamlit, allowing users to analyze sentiment through a simple web interface.

2. Technologies Used

The project leverages various tools and libraries for different stages of processing, model training, and deployment:

- **Python**: Core programming language used for all processing.
- **Streamlit**: Web framework for building an interactive app.
- **Scikit-learn**: Machine learning library for training models.
- **Pandas & NumPy**: Data manipulation and numerical operations.
- **NLTK (Natural Language Toolkit)**: Text preprocessing and tokenization.
- **Pickle**: For saving and loading trained models.

3. Dataset Description

The dataset for this project was collected through web scraping from Wikipedia articles related to Sweden. The dataset initially contained sentences labeled as Positive, Negative, and Neutral based on sentiment analysis.

Data Distribution Before Processing

positive	**309**
neutral	**270**
negative	**121**

Since the Neutral category had significantly more data than Positive and Negative, it was removed to balance the dataset. After filtering, techniques like SMOTE (Synthetic Minority Over-sampling Technique) can be applied to create a balanced dataset for effective model training.

After processing, the dataset contained an equal number of Positive and Negative samples, ensuring better model performance and preventing bias.

4. Workflow of the Project

The project follows a structured workflow, from data collection to model deployment:

Step 1: Data Preprocessing

- **Web Scraping**: Extracted raw textual data from Wikipedia.
- **Text Cleaning**: Removed unwanted characters, numbers, symbols, and HTML tags.
- **Tokenization**: Split text into individual words for processing.
- **Stop-word Removal**: Eliminated common words (e.g., is, the, and).
- **TF-IDF Vectorization**: Converted text into numerical feature representations.

Step 2: Sentiment Analysis Model Training

To classify the sentiment of sentences, the dataset was trained using multiple machine learning classifiers:

- **Applied SMOTE**: Balanced the dataset to ensure equal representation of Positive and Negative sentiments.
- **Trained on multiple classifiers**:
  - Logistic Regression
  - Decision Tree
  - Random Forest
  - Gradient Boosting
  - Naive Bayes
  - K-Nearest Neighbors (KNN)
- **Best Model Selected**: Random Forest Classifier (86.63%).

Step 3: Model Deployment (Streamlit Web App)

- **The trained model was saved using Pickle.**
- **The Streamlit web app was built to allow users to enter text and get sentiment predictions.**
- **The app was tested for performance and usability.**

5. Machine Learning Models & Evaluation

Multiple models were trained and evaluated for their accuracy in predicting sentiment:

	Model	Precision	Recall	F1-score
0	Logistic Regression	85.92%	85.48%	85.33%
1	Decision Tree	73.32%	73.39%	73.31%
2	Random Forest	86.63%	86.29%	86.32%
3	Gradient Boosting	86.63%	86.29%	86.32%
4	Naive Bayes	84.21%	82.26%	81.71%
5	KNN	30.07%	54.84%	38.84%

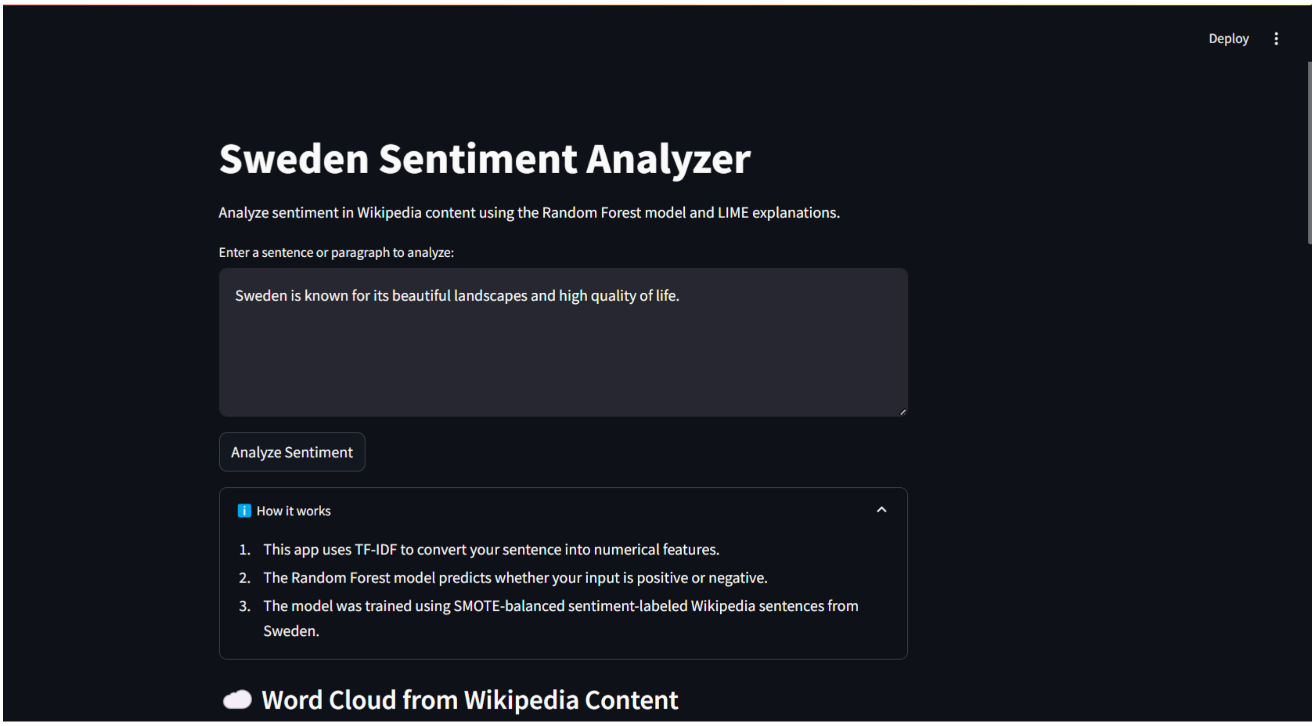
Random Forest performed the best with 86.63% accuracy and was selected as the final model for deployment.

6. Web Application Development (Streamlit)

The Streamlit Web App provides an easy-to-use interface where users can enter a sentence and receive a sentiment prediction in real-time.

Features of the Web App

- **User Input Box**: Allows users to enter a sentence.
- **Sentiment Prediction**: Classifies the input text as Positive or Negative.
- **Probability Scores**: Displays the model’s confidence in the prediction.
- **Visualization**: Displays the model’s confidence with a bar chart.



Sweden Sentiment Analyzer

Analyze sentiment in Wikipedia content using the Random Forest model and LIME explanations.

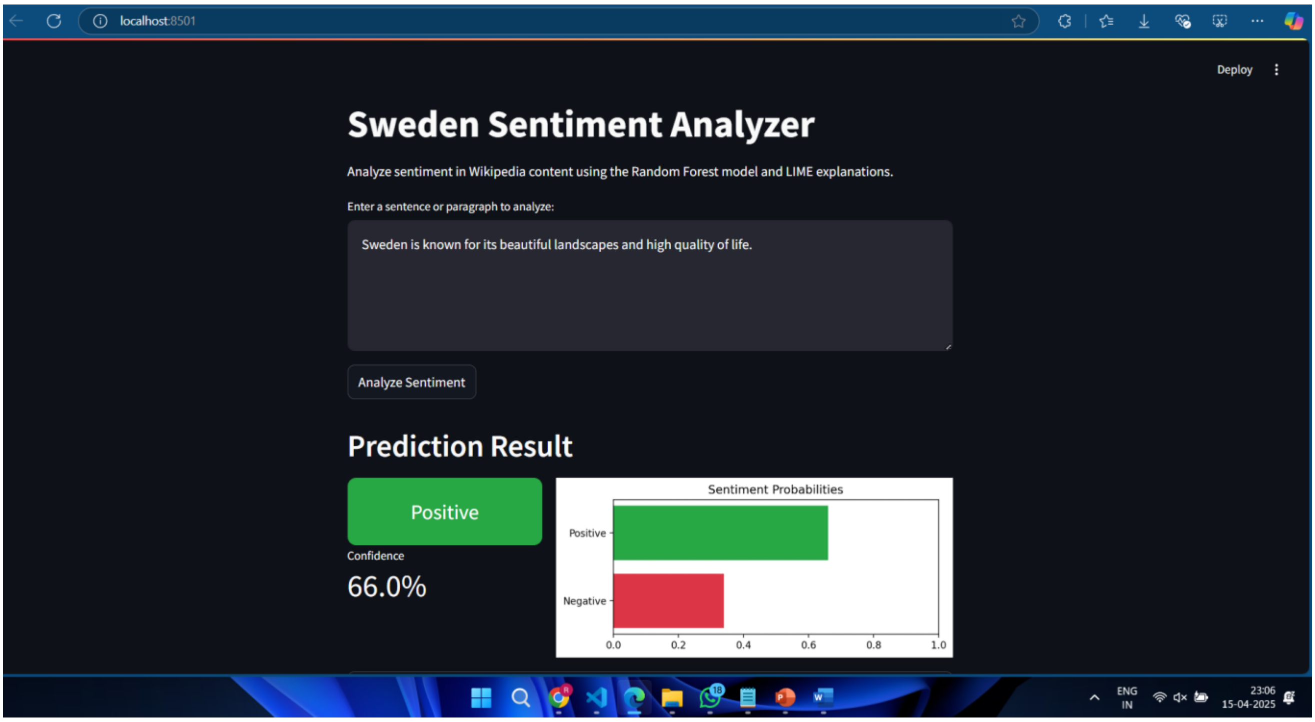
Enter a sentence or paragraph to analyze:

Sweden is known for its beautiful landscapes and high quality of life.

How it works

1. This app uses TF-IDF to convert your sentence into numerical features.
2. The Random Forest model predicts whether your input is positive or negative.
3. The model was trained using SMOTE-balanced sentiment-labeled Wikipedia sentences from Sweden.

Word Cloud from Wikipedia Content



Sweden Sentiment Analyzer

Analyze sentiment in Wikipedia content using the Random Forest model and LIME explanations.

Enter a sentence or paragraph to analyze:

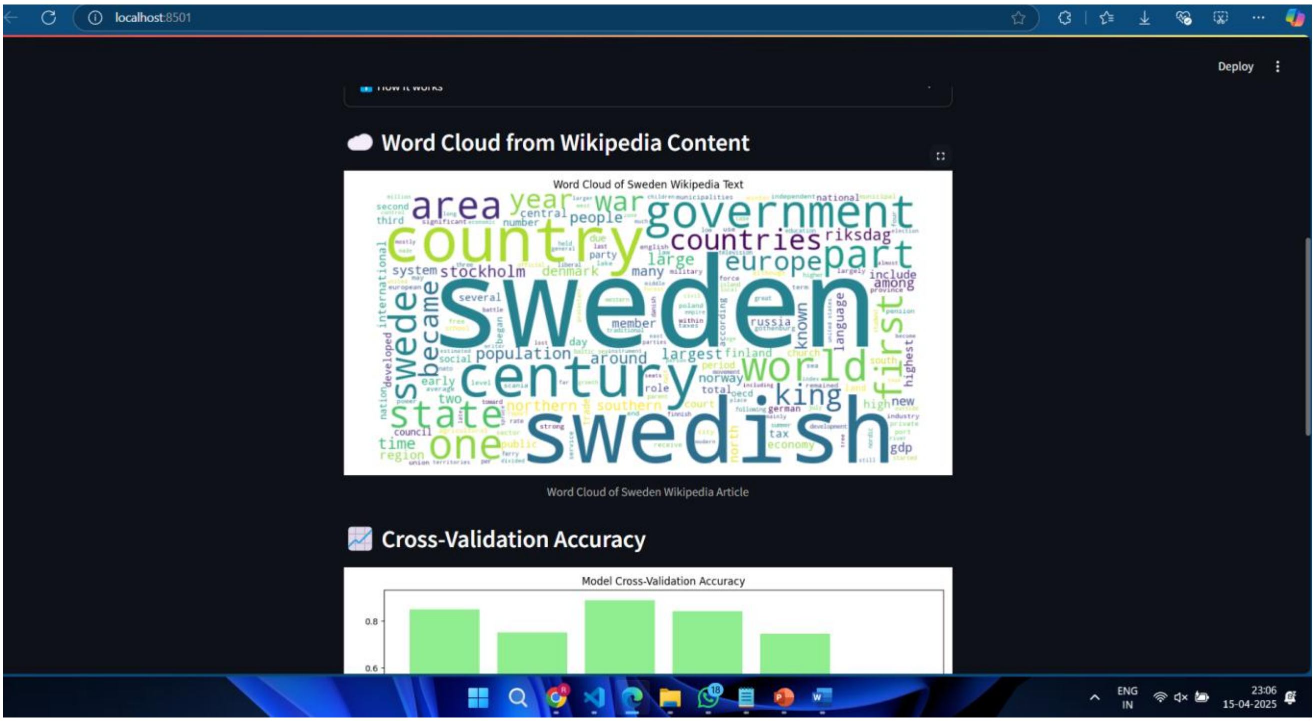
Sweden is known for its beautiful landscapes and high quality of life.

Prediction Result

Positive

Confidence

66.0%

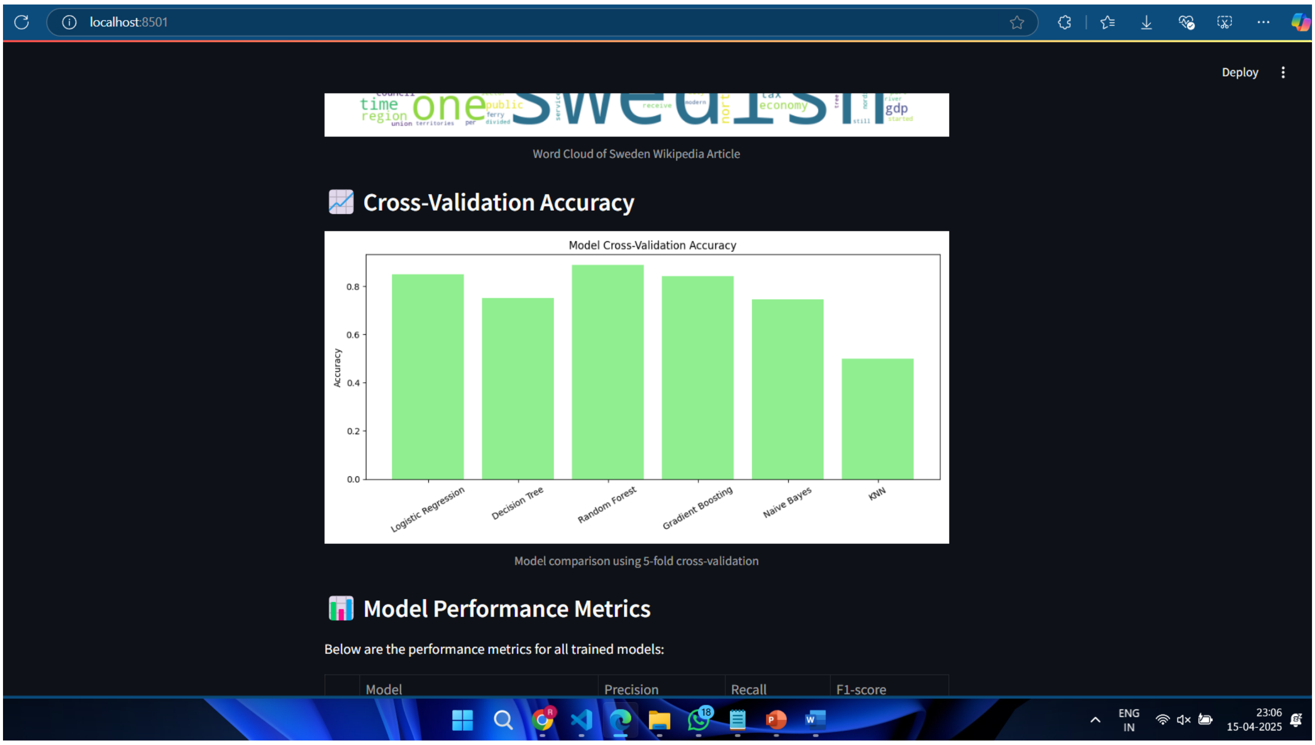


Word Cloud from Wikipedia Content

Word Cloud of Sweden Wikipedia Article

Cross-Validation Accuracy

Model Cross-Validation Accuracy



Word Cloud of Sweden Wikipedia Article

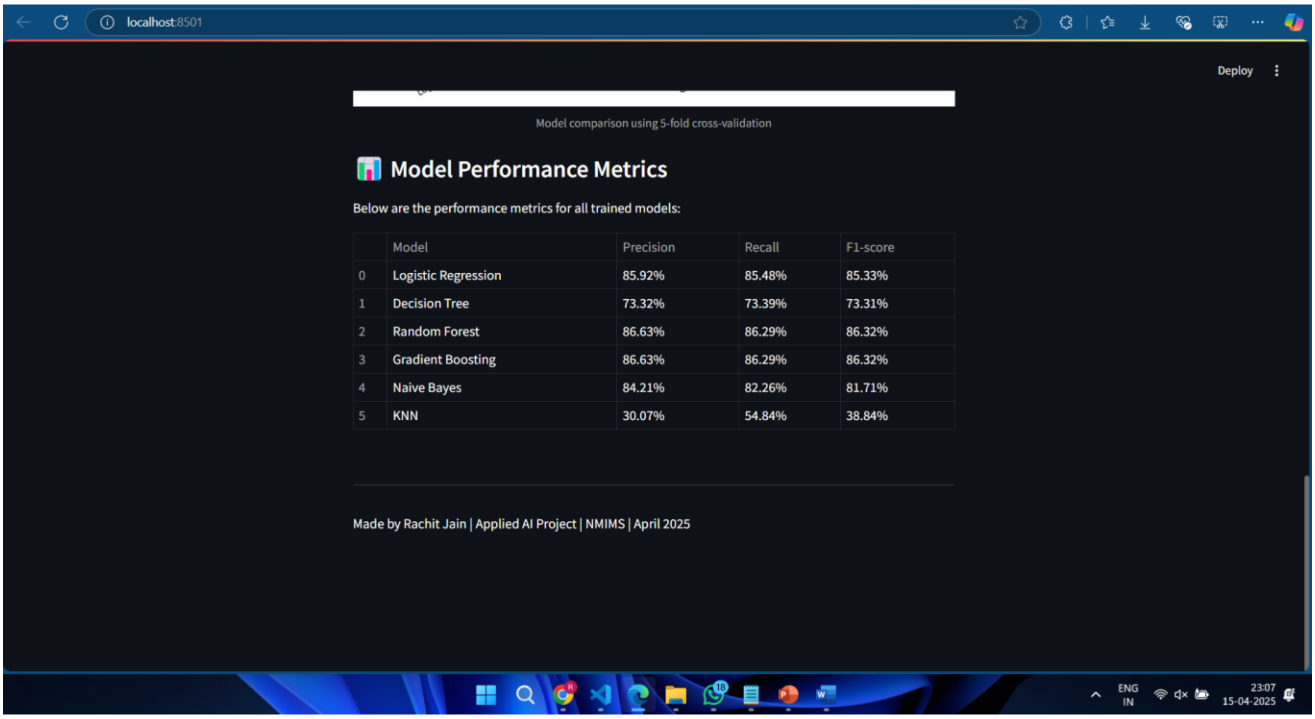
Cross-Validation Accuracy

Model comparison using 5-fold cross-validation

Model Performance Metrics

Below are the performance metrics for all trained models:

	Model	Precision	Recall	F1-score



Model comparison using 5-fold cross-validation

📊 Model Performance Metrics

Below are the performance metrics for all trained models:

	Model	Precision	Recall	F1-score
0	Logistic Regression	85.92%	85.48%	85.33%
1	Decision Tree	73.32%	73.39%	73.31%
2	Random Forest	86.63%	86.29%	86.32%
3	Gradient Boosting	86.63%	86.29%	86.32%
4	Naive Bayes	84.21%	82.26%	81.71%
5	KNN	30.07%	54.84%	38.84%

Made by Rachit Jain | Applied AI Project | NMIMS | April 2025

7. File Structure

```
SWEDEN_SENTIMENT
│
├── data
│   ├── processed
│   │   ├── frequent_words.png
│   │   ├── wordcloud.png
│   │   └── sentences_sentiment.csv
│   └── sweden_wikipedia.txt
│
├── models
│   ├── cross_validation_results.png
│   ├── decision_tree_model.joblib
│   ├── gradient_boosting_model.joblib
│   ├── knn_model.joblib
│   ├── logistic_regression_model.joblib
│   ├── naive_bayes_model.joblib
│   ├── random_forest_model.joblib
│   ├── tfidf_vectorizer.pkl
│   └── model_metrics.json
│
├── src
│   ├── __init__.py
│   ├── app.py
│   ├── model_trainer.py
│   ├── preprocessor.py
│   └── scraper.py
│
├── README.md
└── requirements.txt
```

8. Challenges and Solutions

1. **Class Imbalance Issue**
   - **Challenge**: The dataset contained significantly more Neutral sentiments than Positive and Negative.
   - **Solution**: Removed Neutral data and applied SMOTE to balance the dataset.
2. **Feature Name Mismatch**
   - **Challenge**: Some models (e.g., Random Forest) raised errors due to missing feature names.
   - **Solution**: Converted data into a Pandas DataFrame before training.

9. Future Improvements

- **Expand Dataset**: Include more diverse data sources for better generalization.
- **Add Neutral Sentiment Classification**: Implement three-way classification (Positive, Negative, Neutral).
- **Improve Model Performance**: Experiment with deep learning models (LSTMs, Transformers).
- **Enhance User Experience**: Add visualizations to display sentiment trends over time.

10. Conclusion

This project successfully implements Sentiment Analysis using Machine Learning and deploys it as a real-time web app using Streamlit. The trained Random Forest model achieved 86.63% accuracy, making it effective for text classification tasks.

By integrating text preprocessing, machine learning, and web development, this project demonstrates the practical application of AI in Natural Language Processing (NLP). Future enhancements can further improve model accuracy and usability, making it a valuable tool for real-world sentiment analysis applications.

11. Links

- **GitHub repo**
- **Live App**