Text Sentiment Analysis

This repository contains a Jupyter Notebook for performing text sentiment analysis. The notebook implements various techniques to preprocess text data, build machine learning models, and evaluate their performance for sentiment prediction tasks.

Features

- **Data Preprocessing**: Techniques such as tokenization, stopword removal, stemming/lemmatization, and vectorization (e.g., TF-IDF).
- **Model Training**: Includes machine learning models such as Logistic Regression, Naive Bayes, or any other specified algorithms.
- **Evaluation Metrics**: Performance evaluation using metrics like accuracy, precision, recall, F1-score, and confusion matrices.
- **Visualizations**: Visualization of results and data insights.

Files

- `Task 2 Text Sentimental Analysis.ipynb`: Main notebook containing the sentiment analysis workflow.
- `README.md`: Documentation for understanding and running the project.

Prerequisites

- Python 3.7 or later
- Jupyter Notebook
- Required Python libraries (see below)

Installation

1. Clone this repository:

   ```bash
   git clone https://github.com/your-username/text-sentiment-analysis.git
   ```

2. Navigate to the project directory:

   ```bash
   cd text-sentiment-analysis
   ```

3. Install dependencies:

   ```bash
   pip install -r requirements.txt
   ```

Usage

1. Launch Jupyter Notebook:

   ```bash
   jupyter notebook
   ```

2. Open the notebook `Task 2 Text Sentimental Analysis.ipynb`.

3. Follow the steps outlined in the notebook to preprocess data, train models, and evaluate performance.

Dependencies

The following libraries are required:

- pandas
- numpy
- sklearn

- matplotlib
- seaborn
- nltk

You can install them using the following command:

```bash
pip install pandas numpy scikit-learn matplotlib seaborn nltk
```

Example Workflow

1. **Load Dataset**: Load the text data and corresponding labels.
2. **Preprocess Text**: Tokenize, remove stopwords, and apply lemmatization.
3. **Feature Extraction**: Convert text to numerical features using TF-IDF or CountVectorizer.
4. **Train Model**: Train machine learning models like Logistic Regression or Naive Bayes.
5. **Evaluate Model**: Use metrics like accuracy, precision, recall, and F1-score to evaluate performance.
6. **Visualize Results**: Plot confusion matrix and performance metrics.

Results

Include a summary of results, such as model accuracy or other metrics achieved during evaluation.

License

Contributing

Feel free to fork this repository and submit pull requests with improvements or new features!

Acknowledgments

- Inspired by common sentiment analysis use cases.

- Developed to demonstrate text preprocessing and machine learning workflows.