DA5402: Assign 9

Nikshay Jain | MM21B044

This project implements sentiment analysis on Amazon reviews dataset using distributed computing with Apache Spark and Hugging Face's Transformers library. The implementation follows a map-reduce paradigm for parallel processing across available CPUs.

Project Structure

```
project/
├── main.py              # Main script that ties everything together
├── data_processing.py   # Utilities for data processing
├── visualization.py     # Utilities for visualization
...    ...
├── requirements.txt     # Python dependencies
├── README.md            # This file
└── output/              # Directory for output files
    ├── confusion_matrix.csv
    ├── metrics.txt
    └── sample_results.csv
```

Requirements

- Python 3.8+
- Apache Spark
- Transformers library by Hugging Face
- scikit-learn
- pandas
- matplotlib
- seaborn

Install the dependencies using:

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

Installation

1. Clone the repository
2. Install the dependencies
3. Make sure you have Java installed for Apache Spark

Setup and Run (Step by Step)

1. Clone the Repository and Navigate to Project Directory

```
 git clone https://github.com/Nikshay-Jain/DA5402-Assign-9.git
cd DA5402-Assign-9
```

2. Create and Activate Virtual Environment

```
 # Create virtual environment
python -m venv venv

# Activate it (Windows)
venv\Scripts\activate

# Activate it (macOS/Linux)
source venv/bin/activate
```

3. Install Dependencies

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

4. Run the Main Script

Assuming your Amazon reviews dataset is placed at `data/amazon_reviews.csv`:

```
python main.py --input data/amazon_reviews.csv --output output
```

5. Check Results

```
# View metrics
cat output/metrics.txt

# View confusion matrix
cat output/confusion_matrix.csv
```

Docker Quick Start

If you prefer using Docker:

```
# Start the Docker container
docker-compose up -d

# Run the script inside the container
docker exec -it spark-sentiment-analysis bash -c "cd work && python main.py --input data/amazon_reviews.csv --output output"
```

You can also access the Jupyter notebook at http://localhost:8888 and navigate to `sentiment_analysis_guide.ipynb` for an interactive experience.

Testing

Run the tests to verify the solution:

```
python test.py
```

Tasks

Task 1: Sentiment Analysis

The script uses the pretrained sentiment analysis pipeline from Hugging Face's Transformers library to classify each review text as POSITIVE or NEGATIVE. The processing is parallelized across available CPUs using Apache Spark's distributed computing capabilities.

Task 2: Model Evaluation

The script evaluates the sentiment analysis model by:

1. Converting the original star ratings to binary labels (POSITIVE if rating >= 3.0, else NEGATIVE)
2. Computing the confusion matrix
3. Calculating precision and recall metrics

Output

The script generates the following outputs:

- Confusion matrix (CSV and visualization)
- Precision and recall metrics (TXT file)
- Sample of processed data with true and predicted sentiment labels (CSV)

Data Processing Utilities

The `data_processing.py` module provides utilities for:

- Text cleaning and preprocessing
- Data sampling for testing
- Handling imbalanced data
- Feature extraction using TF-IDF

Visualization Utilities

The `visualization.py` module provides utilities for:

- Plotting confusion matrices
- Visualizing sentiment distribution
- Exploring the relationship between ratings and sentiment
- Comparing different metrics

Notes

- The script assumes that the input file is either in CSV or JSON format
- For large datasets, consider using a proper Spark cluster instead of local mode
- The default sentiment analysis model is from Hugging Face's Transformers library