Social Media Data Mining using Hadoop Framework

Student Name	:	Bathula Veera Mahesh
Roll Number	:	22JK1A0525
College Name	:	KITS AKSHAR INSTITUTE OF TECHNOLOGY
Domain Name	:	Data Science And Big Data Analysis
Title of the project	:	Social Media Data Mining using Hadoop Framework
Submitted to	:	Black Bucks

Algorithm and Workflow:

This algorithm and dataset description provides a well-structured and comprehensive overview of a large-scale social media analysis project. It effectively details the pipeline, from data ingestion to advanced analytics and visualization, leveraging the strengths of the Hadoop ecosystem.

Here's a breakdown of the algorithm and dataset aspects, along with some additional considerations for a more robust project:

Algorithm Description:

The project employs a multi-stage data processing and analysis algorithm, primarily orchestrated around the **Hadoop Ecosystem** for scalability and **Python-based NLP** for intelligent text processing.

Core Algorithmic Flow:

1. Data Ingestion (Stream/Batch):

- **Source:** Social Media APIs (Twitter, Facebook).
- **Method:** Real-time (Kafka/Flume) or historical batch collection (Python scripts with API wrappers like tweepy, requests).
- **Algorithm:** Continuous polling for new data (streaming) or scheduled one-time fetches (batch). Data is structured into JSON/CSV for HDFS.

2. Data Preprocessing (ETL/NLP):

- **Frameworks:** MapReduce, Pig, or Hive (for large-scale transformations); Python (for detailed NLP).
- **Sub-algorithms/Techniques:**

Text Normalization:

- Lowercase conversion: text.lower()
- Remove unwanted characters (e.g., HTML tags, special symbols): Regular expressions (re module).
- Remove stopwords: Lookup against NLTK's standard stopword list.
- Remove emojis: Regular expressions (Unicode ranges for emojis).
- Tokenization: nltk.word_tokenize(), spaCy tokenizer.
- Stemming/Lemmatization (Optional but Recommended): Porter Stemmer, Lancaster Stemmer (NLTK), or spaCy's lemmatizer for reducing words to their root form.

- **Feature Extraction (Basic):** Regular expressions for identifying hashtags (#\w+), mentions (@\w+), and URLs (http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\\(\\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+).

3. Exploratory Data Analysis (EDA):

- **Frameworks:** Hive/Pig for aggregation; Visualization tools (Power BI, Tableau, Kibana); Python (Matplotlib, Seaborn, WordCloud, Plotly, Folium).
- **Algorithms/Techniques:**
  - **Frequency Counting:** Group By operations in Hive/Pig to count occurrences of hashtags, users, locations, etc.
  - **Time Series Aggregation:** Group By timestamp intervals (hour, day, week) to show activity trends.
  - **Geospatial Aggregation:** Group by location to identify regional activity.
  - **Visualization Algorithms:**
    - Word Clouds: Frequency-based sizing of words.
    - Trend Graphs: Line plots showing counts over time.
    - Geo-maps: Plotting aggregated data points on a geographical map.

4. Sentiment Analysis (Optional NLP Module):

- **Approach 1: Lexicon-Based:**
  - **Algorithm:**

- **Load Lexicon:** Import a pre-defined sentiment lexicon (e.g., AFINN, VADER, SentiWordNet) which assigns sentiment scores to words.
- **Score Calculation:** Iterate through tokens in a post, sum up the scores of recognized words.
- **Classification:** Apply thresholds (e.g., score > 0 for positive, < 0 for negative, = 0 for neutral).
- *Example using AFINN:* af = Afinn(); score = af.score(text);

- **Approach 2: Machine Learning-Based:**

    - **Algorithm:** Supervised Learning Classification.

        - **Steps:**

- **Feature Engineering:** Convert text to numerical representations (e.g., Bag-of-Words, TF-IDF using TfidfVectorizer from scikit-learn, or more advanced word embeddings like Word2Vec/GloVe/FastText).
- **Model Training:** Train a classification model (e.g., Naive Bayes, Logistic Regression, SVM, or even more complex models like LSTMs/Transformers for deep learning) on a *labeled dataset* of social media posts.
- **Prediction:** Use the trained model to predict sentiment (Positive, Negative, Neutral) for new, unseen social media posts.
- from sklearn.feature_extraction.text import TfidfVectorizer; from sklearn.naive_bayes import MultinomialNB;
- vectorizer = TfidfVectorizer(); X_train = vectorizer.fit_transform(training_texts); model = MultinomialNB().fit(X_train, training_labels);
- predicted_sentiment = model.predict(vectorizer.transform([new_text]));

5. Result Aggregation and Analysis:

- **Frameworks:** Hive (for large-scale data aggregation); Sqoop (for export); MySQL/PostgreSQL (for structured storage); Python (SQLAlchemy, Psycopg2).
- **Algorithms:**

- **SQL/HiveQL Grouping:** GROUP BY operations on time, location, sentiment, hashtags to create summarized datasets for reporting.
- **Join Operations:** Combine processed data with other datasets if necessary (e.g., user profiles).

6. Visualization & Reporting:

- **Tools:** Tableau, Power BI, Kibana, Grafana; Python Dashboards (Dash, Plotly, Streamlit).
- **Algorithms:** Data visualization algorithms inherent to these platforms to render charts, graphs, and maps based on the aggregated data.

Conditions & Loops (Pythonic Examples):

for loop for token/post iteration:

Program: Python

```
import nltk
from nltk.corpus import stopwords
import re
text_content = "This is a great #project!    Check it out: http://example.com"
tokens = nltk.word_tokenize(text_content.lower())
cleaned_tokens = []
stop_words = set(stopwords.words('english'))
for token in tokens:
    if token.isalpha() and token not in stop_words:
        cleaned_tokens.append(token)
    elif token.startswith("#"):
        # Extract and keep hashtags (remove '#' symbol)
        cleaned_tokens.append(token[1:])
    elif "http" in token:
        # Skip URLs entirely
        continue
    # Future enhancements:
    # - elif token.startswith("@"):  # Handle mentions
    # - elif re.match(r'[^\w\s]', token):  # Remove special characters/emojis
# Print result
print(cleaned_tokens)
```

if condition for sentiment threshold:

Program: Python

```
from afinn import Afinn
af = Afinn()
text = "I love this product, it's amazing!"
score = af.score(text)
if score > 0:
   sentiment = "Positive"
elif score < 0:
   sentiment = "Negative"
else:
   sentiment = "Neutral"
print(f"Text: '{text}', Sentiment: {sentiment}, Score: {score}")
```

Output: Text:

'I love this product, it's amazing!', Sentiment: Positive, Score: 4.0

while loop for continuous ingestion (conceptual):

Program: Python

```python
# This would be more complex, involving Kafka consumer logic or API polling with delays
# Conceptual:
# while True:
#     new_data = kafka_consumer.poll() # or twitter_api.get_stream_data()
#     if new_data:
#         process_data_and_store_in_hdfs(new_data)
#     time.sleep(polling_interval) # Wait before next poll
```

Dataset Description

The project utilizes social media data, which is typically unstructured or semi-structured.

Key Characteristics of the Dataset:

- **Source:** Primarily public social media platforms (Twitter, Facebook).
- **Format:** Raw data usually in JSON or CSV, stored in HDFS.
- **Volume:** Large-scale, potentially petabytes of data, continuously growing (high-velocity).
- **Variety:** Diverse content (text, emojis, URLs, images/videos - though only text is processed here), various user types, geographic locations.
- **Veracity:** Can be noisy, contain slang, sarcasm, typos, spam, and bot-generated content, requiring extensive preprocessing.

Data Fields (Inputs):

- **Tweet/Post ID**: Unique identifier for each social media post.
- **User ID**: Unique identifier for the user who posted.
- **Text Content**: The actual text of the tweet/post. This is the primary field for NLP.
- **Hashtags**: List or string of hashtags used in the post (e.g., #bigdata, #AI).
- **Mentions**: List or string of users mentioned in the post (e.g., @handle).
- **Timestamp**: Date and time of the post, crucial for temporal trend analysis.
- **Location (if available)**: Geographical information (e.g., city, country) where the post originated. Note: Location data from social media APIs can be limited or require user consent.
- **Device/Platform**: (Optional) The device or platform used to post (e.g., "Twitter for Android", "Web").
- **Engagement Metrics**: Numerical values indicating user interaction:
    - Likes (or Favorites)
    - Retweets (for Twitter)
    - Shares (for Facebook)
    - Comments (for Facebook, or replies for Twitter)

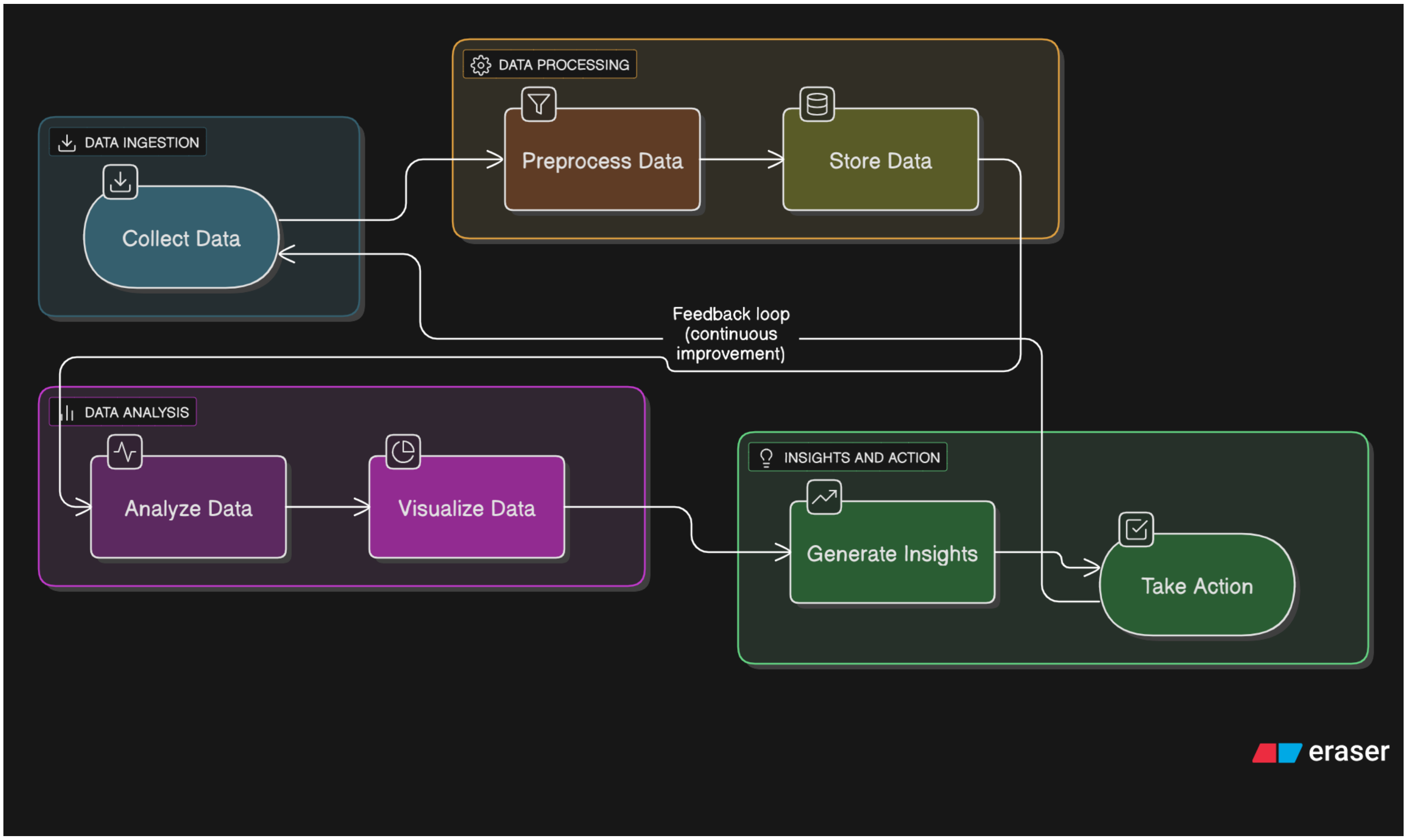
Outputs (Derived Datasets/Reports):

- **Cleaned and Structured Social Media Data:** Preprocessed text, normalized fields, ready for analysis (e.g., stored as Parquet/ORC in HDFS for efficient Hive querying).
- **Top Hashtags/Topics per Region/Time:** Aggregated counts and lists of trending items.
- **Sentiment-Labeled Posts:** Original posts augmented with a Sentiment label (Positive, Negative, Neutral) and potentially a Sentiment Score.
- **Engagement and Reach Metrics per User/Topic:** Aggregated counts of likes, retweets, shares for specific users or topics.
- **Interactive Dashboards and Summary Reports:** Visual representations of trends, sentiment distribution, influential users, etc.

Additional Considerations for the Dataset:

- **Ethical Considerations & Data Privacy:** Especially for real-time collection, ensure adherence to platform API terms of service and data privacy regulations (GDPR, CCPA, etc.). Anonymization of user IDs or PII might be necessary.
- **Sampling vs. Full Stream:** For extremely high-volume platforms like Twitter, often only a sample of the full stream is available via public APIs.
- **Data Skew:** Social media data often exhibits high data skew (e.g., a few highly active users, many inactive ones; popular hashtags versus obscure ones). Hadoop processes handle this well, but it's a characteristic to be aware of.
- **Multilinguality:** Social media is global. If the project aims for global insights, handling multiple languages in NLP becomes a significant challenge requiring language detection and language-specific models/lexicons.
- **Ground Truth for ML Models:** For ML-based sentiment analysis, obtaining a high-quality, *labeled dataset* is critical for training accurate models. This often involves manual annotation or leveraging existing benchmark datasets.

Flow Chart:


DATA PROCESSING
Preprocess Data
Store Data
DATA INGESTION
Collect Data
Feedback loop
(continuous
improvement)
DATA ANALYSIS
Analyze Data
Visualize Data
INSIGHTS AND ACTION
Generate Insights
Take Action
eraser