**Big Data Project**

**Sentiment Analysis**

**WEEKLY REPORT**



**Name: Aman Kumar Singh**

**Course: B.Tech CSE Big Data**

**Batch: Batch 1**

**Sap id: 500095557**

**Weekly Report: Week 1**

**Task Summary**

This week focused on the initial phase of our YouTube sentiment analysis project. The primary goal was to collect and structure data from YouTube comments and store it in a Hadoop Distributed File System (HDFS).

**Tasks Completed**

1. **Data Collection**
   * **Description:** Utilized YouTube Data API to retrieve comments from a specific YouTube video.
   * **Details:**
     + Retrieved comments including user names, comments, and likes.
     + Implemented pagination to retrieve an extensive set of comments.
2. **Data Structuring**
   * **Description:** Structured the collected data into a tabular format.
   * **Details:**
     + Extracted relevant fields: user\_name, comment, and likes.
     + Organized the data into a pandas DataFrame.
3. **Data Storage**
   * **Description:** Stored the structured data in HDFS for further processing.
   * **Details:**
     + Converted the pandas DataFrame into a CSV file.
     + Successfully stored the CSV file in HDFS.

**Challenges and Solutions**

1. **API Rate Limits:**
   * **Challenge:** Encountered rate limits while making API calls.
   * **Solution:** Implemented pagination and handled nextPageToken to ensure continuous data retrieval.
2. **Data Cleaning:**
   * **Challenge:** Ensuring comments were free from HTML tags and special characters.
   * **Solution:** Used Python's libraries to clean the data.
3. **HDFS Storage:**
   * **Challenge:** Ensuring the data was correctly formatted and transferred to HDFS.
   * **Solution:** Verified data integrity post-transfer using HDFS command-line tools.

**Next Steps**

* **Data Cleaning and Preprocessing:**
  + Further clean the data by removing duplicates and irrelevant information.
  + Normalize and tokenize the text data for sentiment analysis.
* **Sentiment Analysis:**
  + Implement sentiment analysis using appropriate tools.
  + Visualize the sentiment distribution of comments.

**Summary**

This week was successful in setting up the foundational data collection and storage mechanisms for our YouTube sentiment analysis project. The collected and structured data is now stored in HDFS, ready for further processing in the upcoming weeks.

**Weekly Report: Week 2**

**Task Summary**

This week was dedicated to processing the collected YouTube comments, conducting data analysis and visualization, and initiating the model building phase for sentiment analysis.

**Tasks Completed**

1. **Data Processing**
   * **Description:** Cleaned and preprocessed the data to prepare it for analysis and modeling.
   * **Details:**
     + Removed duplicates and irrelevant information from the dataset.
     + Normalized and tokenized the text data for further analysis.
     + Applied lemmatization to reduce words to their base form using the spaCy library.
2. **Data Analysis and Visualization**
   * **Description:** Analyzed the cleaned data and created visualizations to understand the sentiment distribution and key trends.
   * **Details:**
     + Counted the occurrences of each sentiment and plotted the sentiment distribution using matplotlib.
     + Generated a word cloud to visualize the most frequently occurring words in the comments.

**Visualizations:**

* + **Sentiment Distribution:**
    - Plotted a bar chart showing the distribution of sentiments (Positive, Negative, Neutral).
  + **Word Cloud:**
    - Combined all comments into a single string and generated a word cloud to highlight the most common words.

1. **Model Building**
   * **Description:** Initiated the model building phase to create a sentiment analysis model.
   * **Details:**
     + Explored different machine learning models suitable for sentiment analysis.
     + Prepared the data for model training by creating training and test datasets.
     + Scaled the data to the range [0,1] using MinMaxScaler to avoid negative values.
     + Started the initial training of the sentiment analysis model using the preprocessed data.

**Next Steps**

* **Complete Model Building:**
  + Finalize the sentiment analysis model.
  + Train and evaluate the model using the prepared datasets.
* **Model Evaluation and Optimization:**
  + Fine-tune the model to improve accuracy and performance.
  + Validate the model using cross-validation techniques.
* **Deployment and Reporting:**
  + Deploy the trained model for real-time sentiment analysis.
  + Generate a comprehensive report on the findings and insights from the sentiment analysis.

**Weekly Report: Week 3**

**Task Summary**

This week focused on finalizing the model building phase, evaluating the trained model, and creating detailed visualizations to compare the model’s predictions with actual sentiments.

**Tasks Completed**

1. **Model Training**
   * **Description:** Completed the training of the sentiment analysis model.
   * **Details:**
     + Split the data into training and test sets.
     + Scaled the data using MinMaxScaler.
     + Trained the model using the preprocessed and scaled data.
2. **Model Evaluation**
   * **Description:** Evaluated the performance of the trained sentiment analysis model.
   * **Details:**
     + Used classification report and confusion matrix to assess the model's performance.
     + Calculated key metrics such as precision, recall, and F1-score.

**Evaluation Metrics:**

* + **Confusion Matrix:**
    - Visualized the confusion matrix using seaborn to show the accuracy of predictions.
  + **Classification Report:**
    - Generated a report showing the precision, recall, and F1-score for each sentiment class.

1. **Data Visualization**
   * **Description:** Created visualizations to compare true sentiments with predicted sentiments and analyze the distribution of sentiments.
   * **Details:**
     + Plotted the distribution of true sentiments using seaborn.
     + Compared the distribution of true vs. predicted sentiments using a count plot.

**Visualizations:**

* + **Confusion Matrix:**
    - Displayed a heatmap of the confusion matrix to visualize prediction accuracy.
  + **True Sentiment Distribution:**
    - Created a count plot to show the distribution of true sentiments.
  + **Comparison of True vs. Predicted Sentiments:**
    - Plotted a comparison of true vs. predicted sentiments using a count plot to highlight discrepancies.

**Challenges and Solutions**

1. **Model Accuracy:**
   * **Challenge:** Achieving high accuracy in sentiment prediction.
   * **Solution:** Fine-tuned the model parameters and scaled the input data for better performance.
2. **Visualization Complexity:**
   * **Challenge:** Effectively visualizing the comparison between true and predicted sentiments.
   * **Solution:** Used seaborn and matplotlib to create clear and informative visualizations.

**Next Steps**

* **Model Deployment:**
  + Deploy the trained sentiment analysis model for real-time sentiment analysis on new data.
* **Continuous Improvement:**
  + Monitor model performance and make iterative improvements based on new data and feedback.
* **Reporting:**
  + Compile a comprehensive report summarizing the entire process, key findings, and insights.

**Summary**

This week was successful in finalizing the sentiment analysis model, evaluating its performance, and creating detailed visualizations to compare the model’s predictions with actual sentiments. The model is now ready for deployment and further optimization based on real-time data.

**Weekly Report: Week 4**

**Task Summary**

This week was focused on the final deployment of the sentiment analysis project using a Streamlit web application. The app allows users to input comments and receive sentiment predictions (Positive, Negative, Neutral).

**Tasks Completed**

1. **Streamlit Application Setup**
   * **Description:** Set up the Streamlit environment to develop a web application for sentiment analysis.
   * **Details:**
     + Created a Streamlit application with navigation options for Home, Predict, and About pages.
2. **Preprocessing and Vectorization**
   * **Description:** Implemented text preprocessing and vectorization functions.
   * **Details:**
     + Used spaCy for lemmatization and removal of stop words and punctuation.
     + Vectorized text using spaCy's language model to prepare it for sentiment prediction.
3. **Model Loading and Prediction**
   * **Description:** Loaded the pre-trained sentiment analysis model and integrated it into the Streamlit app for predictions.
   * **Details:**
     + Loaded the model from a serialized file using joblib.
     + Created a function to preprocess, vectorize, and predict the sentiment of user input.
4. **User Interface Design**
   * **Description:** Designed the user interface for the Streamlit app.
   * **Details:**
     + Home Page: Introduced the app and its functionality.
     + Predict Page: Allowed users to input comments and receive sentiment predictions.
     + About Page: Provided information about the project and the developer.
5. **Final Testing and Deployment**
   * **Description:** Tested the Streamlit application for functionality and performance.
   * **Details:**
     + Verified the accuracy of sentiment predictions.
     + Ensured smooth navigation and user experience across all pages.

**Challenges and Solutions**

1. **Model Integration:**
   * **Challenge:** Integrating the pre-trained model with the Streamlit application.
   * **Solution:** Used joblib to load the model and ensured it worked seamlessly with the app's prediction logic.
2. **User Experience:**
   * **Challenge:** Designing an intuitive and user-friendly interface.
   * **Solution:** Created clear and concise navigation options and ensured the app responded promptly to user inputs.

**Next Steps**

* **Monitor and Improve:**
  + Continuously monitor the app's performance and user feedback.
  + Implement improvements and optimizations as needed.
* **Extend Functionality:**
  + Explore adding more features, such as visualizing the sentiment distribution of batch inputs.
  + Enhance the app to handle more complex text inputs and provide detailed analysis.

**Summary**

This week successfully culminated in the deployment of the sentiment analysis project using Streamlit. The application allows users to input comments and receive sentiment predictions, providing a practical interface for sentiment analysis. The project is now live and ready for further enhancements based on user feedback.