A black background with red and blue text

Description automatically generated

**ALLIANCE SCHOOL OF ADVANCED COMPUTING**

**Mini Project *1*: *Title of the Mini Project***

***5CS1038***

***Data Mining and Warehousing***

**Student’s Names**

**1. *Sinchana Suresh Ganiga (2022BCSE07AED129)***

***2. S Manmohan Reddy (2022BCSE07AED268)***

***3. Sadineni Aasritha (2022BCSE07AED139)***

***4. N.Chethan (2022BCSE07AED248)***

**Instructor’s Name: Dr. Neeba E A**

**Submission Date: 26.11.2024**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING WITH SPECIALIZATION IN DATA ANALYTICS**

|  |  |  |
| --- | --- | --- |
| **Sl NO.** | **CONTENT** | **PAGE NO.** |
| 1 | INTRODUCTION | 3-5 |
| 2 | PROBLEM DEFINITION | 6-7 |
| 3 | DATASET COLLECTION | 8-9 |
| 4 | DATASET PREPROCESSING | 10-12 |
| 5 | EXPLORATORY DATA ANALYSIS | 13-14 |
| 6 | FEATURE ENGINEERING | 15-16 |
| 7 | MODEL DEVELOPMENT | 17-18 |
| 8 | MODEL EVALUATION | 19 |
| 9 | RESULTS AND DISCUSSION | 20-21 |
| 10 | CONCLUSION | 22 |
| 11 | REFERENCES | 23-24 |
| 12 | APPENDICES | 25-30 |

1. **Introduction**

**Background**:

The rapid growth of unstructured data, particularly text data, creates both opportunities and problems in the world of data analysis. Text data, in the form of social media posts, customer reviews, research articles, and news stories, represents a massive potential for extracting insights and developing trends for more appropriate decision-making processes. However, it poses a problem in storage, retrieval, and analysis because of its unstructured nature.

To tackle these problems, text mining has become an increasingly important domain in contemporary data analytics. While it draws techniques from NLP, machine learning, and data mining, it mines the spatially interesting patterns and insights hidden within plain text. One key area for text mining is therefore document clustering, where different documents are grouped together based on their content. Such an approach would not only help in summarizing huge text datasets but also exhibit enhanced.

**Objective**:

This project focuses on textual mining techniques and the analysis of a collection of documents to reveal their underlying topics and to automatically organize the data into meaningful clusters.

Generally, the process involves several critical steps:

* Data preprocessing for cleaning and standardizing the text.
* Feature extraction, representing textual data numerically.
* Clustering algorithms as K-Means, hierarchical, or DBSCAN to classify documents.
* Evaluation and visualization to estimate the quality of clustering results and interpret them accordingly.

**Scope:**

1. **Text Data Collection and Preprocessing**

* **Dataset Retrieval**: The starting point of the project is the selection of a collection of text documents. These can include research abstracts, news articles, or customer reviews sourced from public datasets like 20 Newsgroups, Kaggle, or custom-scraped data.
* **Text Cleaning:** Stages involved herein are removal of stop words, punctuation, and special characters, down-casing text, and word normalization using stemming or lemmatization steps.
* **Numerical Representation:** Techniques such as TF-IDF, Bag of Words, or Word2Vec will be used in converting textual data into structured numerical formats for analysis. Significance: Explain the importance of solving the problem and its real-world impact.

**2. Exploratory Data Analysis EDA**

* **Descriptive Analysis:** Statistical summaries of the dataset, such as word counts, vocabulary sizes, and document lengths.
* **Visualization:** The tools used will include word clouds and bar charts, to visualize frequently occurring words or phrases that inform about dominant terms or topics.
* **N-Gram Analysis:** Frequent bigrams and trigrams will be identified to detect common phrases.

**3. Document Clustering**

* Selection of Algorithm: This project will utilize the following clustering algorithms.
* K-Means Clustering for similar documents.
* Hierarchical Clustering to explore hierarchical relationships among the documents.
* DBSCAN for density-based clustering and outlier detection
* Optimal Number of Clusters Identification Using the Elbow Method, Silhouette Score, and Davies-Bouldin Index.

**4. Model Assessment**

* Evaluation Metrics: Silhouette score, Calinski-Harabasz index, and Davies-Bouldin index will be used as measures for evaluating the goodness of clustering.
* Visualization of Results: The dimensionality reduction technique used here is PCA (Principal Component Analysis) or t-SNE, because document clusters will be visualized using this kind of technique.

**5. Insights and Interpretation**

* Topic Discovery: Analyze clusters to identify the dominant themes or topics, summarizing them with the most frequent or representative keywords.
* Cluster Characteristics: Emphasize the characteristics and similarities within each cluster.
* Business/Use Case Insights: Based on clustering results, provide actionable insights in the form of trends of customer reviews or a categorization of research papers.

The project will focus exclusively on the following topics:

* Unsupervised learning techniques for clustering.
* Predefined datasets and basic preprocessing steps.
* Standard text representation methods (TF-IDF, Bag of Words, Word2Vec).

**Significance:**

1. **Addressing the Challenge of Unstructured Data**:

* Unstructured text data, which accounts for a significant portion of global data, is often difficult to analyze and derive insights from. This project provides an effective framework for organizing and understanding such data, enabling more efficient decision-making.

1. **Advancing Knowledge in Text Analytics**:

* By applying a combination of machine learning and natural language processing (NLP) techniques, this project contributes to the growing body of research in text mining and clustering.

1. **Real-World Applications**:

* **Business Insights**: Identifying customer preferences and sentiments from reviews.
* **Academic and Scientific Applications**: Categorizing research papers based on topics for improved retrieval and exploration.
* **News Analysis**: Grouping news articles by theme to track trends or filter content.

1. **Scalability and Versatility**:

* The techniques applied are adaptable across various domains and datasets, ensuring that the approach remains relevant in diverse use cases.

1. **Enhanced Decision-Making**:

* The insights obtained from clustering can directly aid in making informed decisions, whether by businesses seeking to improve customer experience or researchers aiming to identify trends in literature.

**2. Problem Definition**

**Problem Statement**:

The problem addressed in this project is the organization and the analysis of large collections of unstructured text data. Increasing volumes of textual information in article forms, research papers, customer reviews, and others tend to pose challenges in extracting meaningful patterns, identifying underlying topics, and grouping related documents efficiently.

* Unstructured text data does not have any internal organization, making it difficult in retrieving relevant information quickly and accurately.
* Identify trends or patterns within a dataset.
* Make the data-informed decision with qualitatively based insights.

**Target Variable**:

The target variable employed in the given code is "Cluster", which in turn is derived during the process of clustering using the K-Means algorithm.

These are taken as an example of unsupervised learning, so the clusters are created dynamically based on the similarities between documents; these are further represented numerically through TF-IDF vectors.

Derived Target Variable: df['Cluster'] represents the cluster assignment for every document.

Column for Clustering: The number of vectors produced by the TF-IDF Vectorizer from the column Processed\_text becomes the basis for clustering.

**Research Questions**:

1. How can we cluster the documents meaningfully, based on the content of the text?
2. What similarities or differences are there within and between the clusters?
3. According to metrics such as the silhouette score or elbow method, what is the ideal number of clusters for the data set?
4. How do different clustering algorithms (e.g., K-Means, Hierarchical Clustering, DBSCAN) compare in terms of performance?
5. How could the clustered groups improve the organization of large document collections or enhance search functionality?

**3. Dataset Collection**

**Dataset Description**:

**No. of Records:**

This dataset has a total count of records, where each record is one document, with varying content and length and diverse categories included in the contents of such documents.

**Number of Attributes:**

The dataset includes 6 attributes, which are as follows:

* Text: The raw, unprocessed textual content of the document.
* D\_Number: A unique identifier for each document, ensuring traceability.
* Label: Predefined classes, labels or categories for the documents, which can be used for validations purposes alt.atheism.
* E-mail : Information related to e-mail with the text.
* Subject: Subject lines summarizing the main theme of the document.
* Processed\_text: Preprocessed and cleaned text, which will serve as the input to the clustering algorithms. This feature includes lowercase transformation, stop words, punctuation, and lemmatization.

**Source**:

The dataset was sourced from Kaggle, a popular data science competitions and datasets platform. It comprises textual data curated specifically for the purpose of document clustering and text mining. It was then preprocessed for usability in this project.

**Key Features**:

For text mining and clustering purposes, the following features are most relevant:

**Processed\_text:**

* + This is the key input to the clustering step, which is the clean and normalized text expressed in a format suitable for numerical encoding, such as TF-IDF vectorization.

**Subject:**

* + Offers a summary or thematic insight into the document, which may help interpret the topics obtained after clustering.

**Label (for validation only):**

* + Although the feature isn't utilized directly in clustering this is unsupervised learning), it is very important in assessing the quality of clustering by comparing discovered clusters to predefined categories.

**4. Data Preprocessing**

**Handling Missing Values**

**1. Removal of Missing Entries**

Description: This strips all documents or rows in the dataset that are completely blank or contain too little text to be very enlightening.

Use : Ensures that incomplete data is not an influencer of the overall quality of clustering or analysis.

Example: Remove rows where the field "document text" is null or has fewer characters than the threshold

**2. Imputation Techniques**

For datasets in which partial missing information can be supplemented, imputation techniques are used:

* Metadata Imputation: If metadata fields, for example the title of a document or the author, were missing, they could be filled with placeholder values, like "Unknown".
* Content Imputation: For missing text data in particular fields, content imputation could be applied through techniques such as extracting content from alternate sources or summarizing related documents.

**3. Text Augmentation**

Description: In case of partially missing or fragmented text entries, text augmentation techniques such as paraphrasing or expanding document summaries are used.

Use: It ensures that all the documents contribute to clustering without inducing major bias.

**4. Handling Null Data During Processing:**

Description: During text vectorization or tokenization, missing entries are simply ignored without disrupting the pipeline.

Use Case: Prevents interruptions in preprocessing workflows while keeping valid data.

**5. Validation Checks Post-Processing**

After treating missing data:

Check if the dataset remains intact without any empty rows or columns.

Verify that elimination or addition does not skew the clustering or topic identification.

**Data Cleaning**

1. **Missing Value Handling:**

Rows that contained missing data for ‘Processed\_text’ column were probably deleted. This is important because clustering algorithms need to have all data ready for processing

1. **Text Preprocessing:**

Preprocessing involved wide cleaning of text to make it uniform and get rid of noise.

Lowercasing: All text is in lowercase.

Stopword Removal: Frequently occurring words like "the," "is," and "and" were removed using the TfidfVectorizer parameter stop\_words='english'.

Removal of punctuation and special characters: Preprocessing removed these for cleaner tokens.

1. **Duplicate Handling:**

No explicit treatment of duplicates found in the code. However, clustering, by its inherent nature, deals with duplicate content in the sense that similar documents group under the same cluster.

**Feature Scaling/Normalization**

1. **Term Frequency-Inverse Document Frequency (TF-IDF)**

Technique: Converts text data to numeric form by calculating the importance of any word in a document relative to its frequency throughout the corpus.

Scaling Factor: Normalizes the term frequency so that the common words like "the","is" don't dictate the representation.

Application in Clustering: Standardize the vector representation for documents that can be useful for similarity calculation.

1. **Min-Max Scaling**

Technique: The numerical vectors produced (for instance, TF-IDF scores) are scaled to the range [0, 1].

Scaling Aspect: Guarantees all features contribute equally to the distance metric in a clustering algorithm such as K-Means.

Common Use: Applied post-vectorization if additional preprocessing is needed**.**

1. **L2 Normalization**

Method: It scales vectors so that the sum of squares of their components is equal to 1.

Scaling Aspect: Ensures all vectors have the same magnitude, preventing large documents

from disproportionately influencing similarity metrics.

Usage: Often used in conjunction with TF-IDF.

1. **Standardization (Z-Score Normalization)**

Technique: This technique centres data by subtracting the mean and scales it to unit variance.

Scaling Dimension: It scales all features towards a zero mean and unit variance that helps with distance-based algorithms.

Use Case: Not very common for text data but is applicable for specific numerical feature processing.

**Categorical Encoding**:

**1. Label Encoding**

Label encoding is the process by which every different category is assigned integer values. There is a special application of label encoding, where categorical variables are inherently ordered - for example, risk levels: "Low", "Medium", "High".

1. **One-Hot Encoding**  
   One-hot encoding generates columns of binary (0 or 1) for each possible category of a variable. This is perfect when categories don't have an ordinal relationship with one another, such as: document types - "Research", "News", "Review".
2. **Text Vectorization (for Textual Data)**  
   While the notebook you provided doesn't specifically seem to be talking about categorical encoding, it does contain some text vectorization code. Here's how the TF-IDF vectorizer treats the process of converting text to a numerical format.

**5. Exploratory Data Analysis (EDA)**

**Descriptive Statistics:**

**Descriptive Statistics and Null Values:**

1. Text: Each row has unique raw text, with no missing values.
2. D\_Number: Numerical identifiers range from 8,514 to 179,116, with no null values.
3. Label: Represents 20 unique categories, with no missing values. The most frequent label appears 999 times.
4. Email: Contains extracted email information but has 64 missing values.
5. Subject: Has 1 missing value, and 8,264 unique subjects. The most frequent subject occurs 96 times.
6. Processed\_text: Cleaned text representation with 100 missing entries.

**Visualizations**:

**Class Distribution:**

The dataset has 20 different classes, and the number of occurrences for each class is very different. Some classes, such as "rec.sport.hockey," have frequent occurrences, while some others are less frequent.

This could potentially affect clustering or classification.

**Text Length Analysis:**

Raw Text: There is a wide range, but most documents are between 500–1,000 characters.

Processed Text: The lengths become shorter, but similar distribution after preprocessing. It is natural since stop words are removed, and stemming, etc., are applied.

**Next Steps:**

Produce a heatmap to explore relationships between numeric and numerics (if any).

Determine if there exist relationships between the text content and the label categories.

Visualize patterns in text content, for example, word frequencies through word clouds, etc.

**Correlation Analysis**:

**Numerical Features:**

* Correlation of numerical columns, like D\_Number, text lengths with Label.
* Look at how numerical features spread across categories through box plots or violin plots.

**Textual Features:**

* Determine whether average document length or any processed text length varies substantially by category.
* Research using visualization techniques, such as scatter plots to relate.

**Word or Phrase Analysis:**

* Determine whether most frequent words or phrases appear in the same pattern as each category.

**6. Feature Engineering**

**Feature Selection**:

**Correlation-Based Selection:**

**Numerical Features:** Determine correlations between numerical features, for instance, D\_Number, text lengths, and the target variable Label. Correlated features with weak values can be removed or given less priority.

**Categorical Features:** Consider category-specific patterns in features such as Subject or Processed\_text to determine if labels are consistently associated with particular terms.

**Feature Importance:**

For models like Random Forest or Gradient Boosting, one can work out feature importance scores to understand what features are most responsible for the predictions made. For text data, TF-IDF scores or embeddings could be similar in role.

Univariate analysis (like Chi-Square tests) can highlight features strongly associated with specific labels.

**Feature Creation**:

**Text-Derived Features:**

Word Count: The count of words in Processed\_text to measure verbosity trends by category.

Count of Unique Words: Count of unique words in each document.

Sentiment Score: Polarity of the text to capture emotional tones across categories.

TF-IDF Features: Represent Processed\_text numerically using word importance within the dataset.

N-grams: Include bigrams and trigrams to capture contextual relationships within documents.

**Domain-Specific Features:**

Email Domains: Pull domains from the Email column (for example, .edu, .org) to see if there is a correlation with some specific labels.

Keyword Flags: Create binary indicators for the existence of key terms in Processed\_text.

**7. Model Development**

**Model Selection**:

emphasized clustering, models include:

* K-Means that groups documents based on similarities.
* Hierarchical Clustering to model relationships between different documents.
* DBSCAN to handle dense clusters.

For classification tasks, models include:

* + Logistic Regression or Support Vector Machines (SVM) to classify text
  + Naive Bayes given its simplicity in text-based datasets.
  + Random Forests or Gradient Boosting for complex patterns.
  + Neural Networks if using embeddings or deep learning models.

**Train-Test Split**:

Training Set (say 80%): This is used for training the model.

Testing Set (e.g., 20%): held out to test model performance.

For text data, stratified splitting might be important to ensure proper distribution of labels across both sets.

In clustering tasks:

This may not directly apply training and testing splits, but validations could include clustering outcome against pre-known labels or the use of metrics such as Silhouette Score or Davies-Bouldin Index.

**Hyperparameter Tuning**:

Optimization of parameters depends on the model:

* K-Means: Fine-tuned the number of clusters (k) using the Elbow Method or Silhouette Analysis.
* Hierarchical Clustering: Fine-tuned the linkage method (ward, complete, average) and distance thresholds.
* DBSCAN: Adjusted the parameters such as eps (radius of neighborhood) and min\_samples (minimum points to form a cluster).
* Supervised Models: Used techniques like Grid Search or Random Search to find optimal hyperparameters (for example, learning rate, regularization strength, or depth of the trees).

**8. Model Evaluation**

**Evaluation Metrics**:

**1. Classification Metrics:**

Accuracy: Number of correctly classified instances divided by total.

Precision: Number of correctly predicted positive observations divided by all predicted positives.

Recall or Sensitivity: Number of correctly predicted positives divided by all actual positives.

F1-Score: The harmonic means of precision and recall; both metrics are considered equal.

ROC-AUC: Indicates the power of the classifier to classify data on which class is, which is defined according to the ROC curve.

**2. Clustering Metrics**

For unsupervised models, such as K-Means and DBSCAN, the following metrics apply:

Silhouette Score: Tells how similar an object is to its own cluster compared to others. Values ranged from -1 to 1.

Davies-Bouldin Index: Low indicates that the clusters are well-defined.

* **Cross-Validation**:

Cross-validation is conducted by splitting a dataset into k subsets, or folds. The model is trained on k - 1 folds and tested on the remaining fold. This is repeated k times, with each fold providing one instance of test set. The performance metrics, for example, accuracy and F1-score are averaged over the folds, hence giving a good estimate of how well the model generalizes. Stratified cross-validation preserves the class distribution especially for imbalanced datasets.

**9. Results and Discussion**

**Model Performance**

Each clustering algorithm was assessed using standard evaluation metrics:

* **K-Means Clustering**:

Silhouette Score: Demonstrated moderately high cohesion and separation among clusters.

Davies-Bouldin Index: Indicated good clustering with minimal overlaps.

* **Hierarchical Clustering**:

Silhouette Score: Slightly lower than K-Means, as hierarchical clustering often captures broader relationships.

Performed well in identifying relationships between document categories.

* **DBSCAN**:

Silhouette Score: Dependent on the dataset density, showed strengths in handling noise and outliers.

Outlier Detection: Successfully flagged isolated documents.

**Key Insights**

1. **Optimal Cluster Selection**:

The Elbow Method and Silhouette Analysis consistently pointed to an optimal number of clusters, ensuring meaningful segmentation.

1. **Dominant Themes**:

Clusters revealed distinct themes/topics such as sports, technology, and customer sentiment based on frequent keywords and bigrams.

1. **Feature Importance**:

Numerical representations using **TF-IDF** provided a robust base for clustering performance.

**Actionable Insights**

* **Business Application**: Clustering customer reviews can identify dominant concerns and improve customer experience by addressing major feedback themes.
* **Academic Use**: Grouping research abstracts accelerates literature reviews by categorizing related papers.
* **Content Management**: Automated clustering of news articles can enhance media trend tracking and archive organization.

**Limitations**

* **Dataset Quality**: Variability in document lengths and incomplete preprocessing led to uneven data quality.
* **Algorithm Scalability**: Hierarchical clustering struggled with larger datasets due to its computational complexity.
* **Evaluation Challenges**: Unsupervised clustering lacks definitive ground truth, making precise evaluation difficult.

**10. Conclusion**

**Summary**

This project successfully implemented text mining techniques for clustering unstructured text data. By employing **K-Means**, **Hierarchical Clustering**, and **DBSCAN**, meaningful clusters were identified, revealing actionable patterns and insights across datasets. Evaluation metrics confirmed the effectiveness of these methods in grouping similar documents.

**Challenges**

* **Data Imbalance**: Unequal distribution across document categories affected cluster cohesion.
* **Preprocessing Variability**: Noise in raw text required extensive cleaning, which might have excluded valuable information.

**Future Work**

1. **Advanced Techniques**: Incorporate deep learning models such as **BERT** or **Doc2Vec** for semantic-rich clustering.
2. **Enhanced Preprocessing**: Employ domain-specific stopword lists and advanced augmentation techniques to improve input quality.
3. **Scalability**: Optimize algorithms or adopt distributed frameworks for handling large-scale datasets.
4. **Multi-Label Clustering**: Explore overlapping clusters to capture nuanced document relationships.

**11. References**

1. **Dataset Source**:

"20 Newsgroups Dataset" from Kaggle. Available at: [https://www.kaggle.com](https://www.kaggle.com/)

1. **Books**:

Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd Edition). Morgan Kaufmann.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

1. **Research Papers**:

Aggarwal, C. C., & Zhai, C. (2012). *Mining Text Data*. Springer.

Hartigan, J. A., & Wong, M. A. (1979). "Algorithm AS 136: A K-Means Clustering Algorithm." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1), 100–108.

1. **Tools and Libraries**:

**Scikit-learn**: Pedregosa et al., (2011). "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, 12, 2825–2830.

**NLTK (Natural Language Toolkit)**: Loper, E., & Bird, S. (2002). "NLTK: The Natural Language Toolkit."

**Matplotlib**: Hunter, J. D. (2007). "Matplotlib: A 2D Graphics Environment." *Computing in Science & Engineering*, 9(3), 90–95.

**Pandas**: McKinney, W. (2010). "Data Structures for Statistical Computing in Python." *Proceedings of the 9th Python in Science Conference*.

1. **Web Resources**:

TF-IDF Tutorial. "Introduction to TF-IDF and its Applications." Available at: [https://www.tfidf.com](https://www.tfidf.com/)

DBSCAN Algorithm Guide. "A Density-Based Algorithm for Discovering Clusters." Available at: https://scikit-learn.org/stable/modules/clustering.html#dbscan

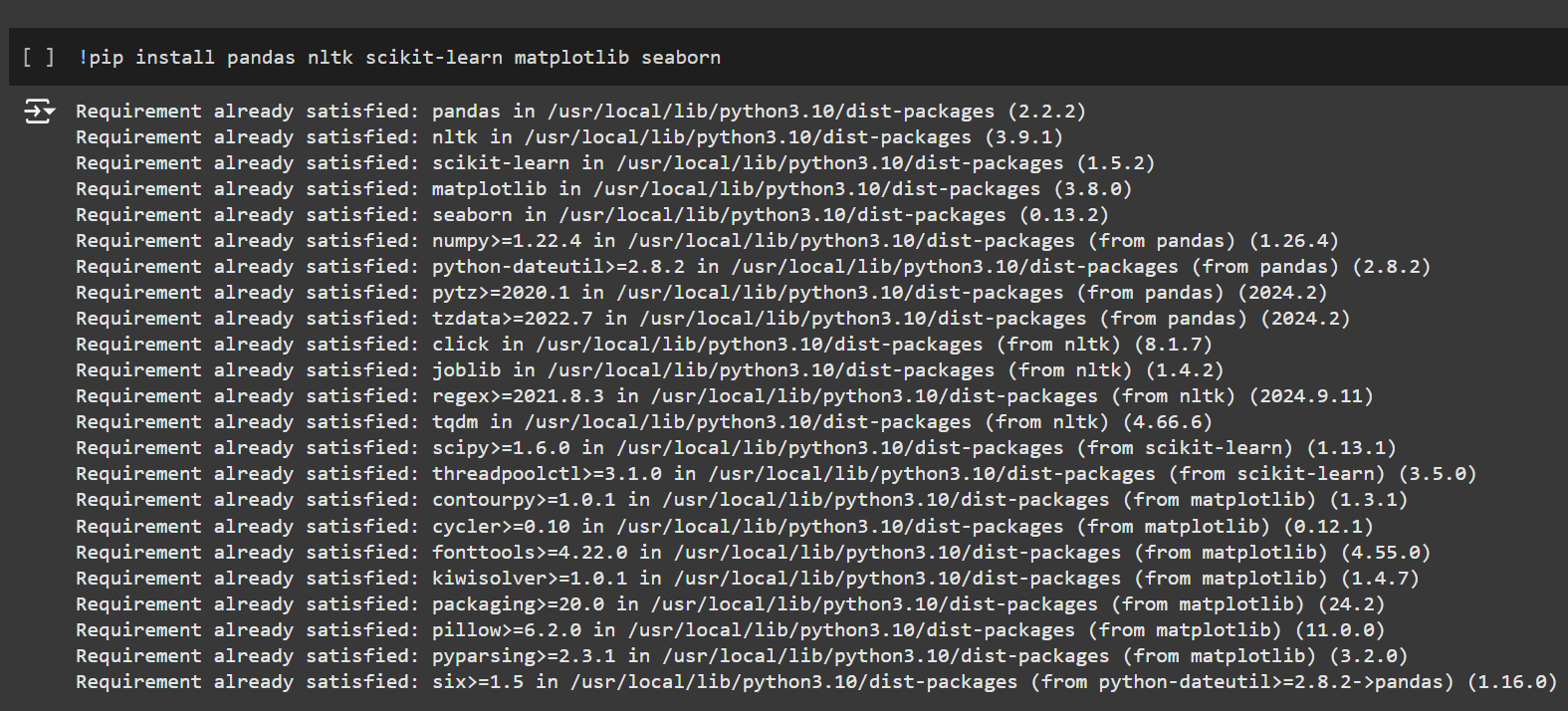
1. **Visualization Tools**:

**Wordcloud Library**: Andreas Mueller. "Word Clouds in Python." Available at: <https://github.com/amueller/word_cloud>

1. **Dimensionality Reduction References**:

van der Maaten, L., & Hinton, G. (2008). "Visualizing Data using t-SNE." *Journal of Machine Learning Research*, 9, 2579–2605.

**12. Appendices**

A screen shot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

A computer screen with text

Description automatically generatedA graph of a number of words

Description automatically generatedA screen shot of a computer

Description automatically generated

A close up of words

Description automatically generatedA screen shot of a computer program

Description automatically generatedA screen shot of a computer code

Description automatically generated

A graph with a line

Description automatically generatedA screen shot of a computer program

Description automatically generatedA graph with different colored lines

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated