Data Collection and Preprocessing Phase:

Date	09 July 2024
Team ID	SWTID1719935665
Project Title	GeminiDecode: Multilanguage Document Extraction by Gemini Pro
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template:

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description
Data Overview	1. Data Sources: The dataset includes multilingual documents collected from different online repositories, academic databases, and organizational records. 2. Basic Statistics: • Total Number of Documents: 50,000 • Languages Covered: English, Spanish, French, German, Chinese, Arabic • File Formats: PDF, DOCX, TXT 3. Dimensions: • Number of Records: 50 000 documents • Attributes: Document ID: a unique identifier of the document • Language: Language to which the document belongs, Content: the proper text part of any document, Metadata, Author, Date, etc. 4. Structure: • Document ID: A unique identifier for each document. • Language: Language to which the document belongs. • Content: This is the proper text part of any document.

	• Metadata: This is supplementary information about a document.
Univariate Analysis	1. Language Distribution: • English: 30% • Spanish: 20% • French: 15% • German: 15% • Chinese: 10% • Arabic: 10% 2. Content Length: • Mean: 1.500 words • Median: 1.200 words • Mode: 1,000 words 3. Metadata Analysis: • Authors: Most frequent authors, average number of documents per author. • Publication Dates: Distribution of documents over time.
Bivariate Analysis	1. Language vs. Content Length: • Scatter Plot: Content length distribution for various languages. 2. Language vs. Metadata: • Correlation Analysis: Language of documents and their publication date. 3. Content Length vs. Publication Date: • Trend Analysis: Document length over time.
Multivariate Analysis	1. Language, Content Length and Publication Date: • 3D Scatter Plot: Interaction between language, word count, and date. 2. Clustering Analysis: • K-Means Clustering: Documents clustered by language, word count, and metadata. 3. Principal Component Analysis (PCA): • Dimensionality Reduction: It involves identifying major components that capture maximum variance within the dataset.
Outliers and Anomalies	1. Identification of Outliers: • Z-Score Method: Identify documents with an extreme length of content. • IQR Method: To identify outliers in metadata attributes such as publication dates.

	2. Treatment of Outliers: • Content Length: Trim or transform extreme values. • Metadata Anomalies: Records having incorrect/suspicious metadata are corrected or removed. 3. Missing Values: • Detection: Recognition of missing values from document content and metadata. • Resolution: ○ Imputation: The missing values would be filled with the mean/median/mode. ○ Filtering: A lot of records having large missing data are removed. 4. Duplicates: • Detection: The duplicate documents are detected by similarity of contents. • Resolution: The duplicate records are removed to assure accuracy of the data.
Data Preprocessing Code Screenshots	
Loading Data	Code to load the dataset into the preferred environment (e.g., Python, R). `` `python import pandas as pd data = pd.read_csv('dataset.csv')` ``
Handling Missing Data	Code for identifying and handling missing values. `` `python data.fillna(data.mean(), inplace=True)` ``
Data Transformation	Code for transforming variables (scaling, normalization). `` `python from sklearn.preprocessing import StandardScaler scaler = StandardScaler() data_scaled = scaler.fit_transform(data)` ``

Feature Engineering	Code for creating new features or modifying existing ones. `` `python data['new_feature'] = data['feature1'] * data['feature2']` ``
Save Processed Data	Code to save the cleaned and processed data for future use. `` `python data.to_csv('processed_data.csv', index=False)` ``

Code Details

Loading Data

```
python
Copy code
import pandas as pd
data = pd.read_csv('dataset.csv')
```

This code snippet imports the Pandas library and loads the dataset from a CSV file into a DataFrame.

Handling Missing Data

```
python
Copy code
data.fillna(data.mean(), inplace=True)
```

This code snippet fills missing values in the dataset with the mean of each column.

Data Transformation

```
python
Copy code
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
```

This code snippet uses Scikit-learn's `StandardScaler` to standardize features by removing the mean and scaling to unit variance.

Feature Engineering

```
python
```

```
                    Copy code
data['new_feature'] = data['feature1'] * data['feature2']
```

This code snippet creates a new feature by multiplying two existing features.

Save Processed Data

```
python
Copy code
data.to_csv('processed_data.csv', index=False)
```

This code snippet saves the cleaned and processed data to a new CSV file.