Index

I. Data Overview:

1. Data Shape and Size: Number of rows and columns in the dataset.

2. Data Types: Data types of each column (e.g., title: string, text: string, label: categorical).

II. Data Cleaning:

1. Handling Missing Values:

- **Detection:** Identify missing values in the dataset.
- **Removal:** Remove rows or columns with missing values if necessary.

2. Handling Outliers:

- **Detection:** Identify outliers in the dataset using techniques such as Z-score or modified Z-score.
- **Removal:** Remove outliers from the dataset if necessary.

3. Handling Duplicate Records:

- **Detection:** Identify duplicate records in the dataset.
- **Removal:** Remove duplicate records from the dataset.

4. Language Detection:

- Detect the language of the text data to ensure consistency.

5. Text Preprocessing:

- **Tokenization:** Split text into individual words or tokens.
- **Stopword Removal:** Remove stopwords from the text data.
- **HTML Tag Removal:** Remove HTML tags from the text data.
- **Punctuation Removal:** Remove punctuation from the text data.
- **Lemmatization:** Reduce words to their base form using lemmatization.
- **Removing Special Characters:** Remove special characters from the text data.
- **Removing Non-ASCII Characters:** Remove non-ASCII characters from the text data.

III. Univariate Analysis:

1. Title Column:

- **Title Length Analysis:**
  - Highest Word Count.
  - Lowest Word Count.
- Most Common Words in Titles.

- Analysis of Skewness in Word Counts for 'FAKE' and 'REAL' Titles.
- Title Character Distribution.
- Distribution of Title Word Count.
- Title Sentiment Analysis.

2. Text Column:

- **Text Length Analysis:**
  - Highest Word Count.
  - Lowest Word Count.
- Most Common Words in Texts.
- Analysis of Skewness in Word Counts for 'FAKE' and 'REAL' Texts.
- Text Readability Analysis.
- Text Sentiment Analysis.

3. Label Column:

- Class Distribution.
- Class Balance Analysis.

4. Summary

IV. Bivariate Analysis:

1. Title vs. Label:

- **Top 10 Most Frequent Words** in "Real" and "Fake" Articles.
- **Distribution of Word Count** for "Fake" and "Real" Titles.
- **Shapiro-Wilk Test** for Fake and Real Titles.
- **Two-sample T-test** for Fake and Real Titles.
- **Title Length Analysis:**
  - Alphabetic Count.
  - Shapiro-Wilk Test for Alphabetic Count.
  - Two-sample T-test for Alphabetic Count.
- Title Length by Label.
- Title Sentiment by Label.

2. Text vs. Label:

- **Top 10 Most Frequent Words** in "Real" and "Fake" Articles.
- **Distribution of Word Count** for "Fake" and "Real" Texts.
- **Shapiro-Wilk Test** for Fake and Real Texts.
- **Two-sample T-test** for Fake and Real Texts.
- **Text Length Analysis:**
  - Alphabetic Count.
  - Shapiro-Wilk Test for Alphabetic Count.
  - Two-sample T-test for Alphabetic Count.
- Text Length by Label.
- Text Sentiment by Label.

V. Text Analysis:

1. Sentiment Analysis:

- Perform sentiment analysis on the text data to determine sentiment polarity (positive, negative, neutral) of the articles.

2. Co-occurrence Analysis:

- Analyze the co-occurrence of words to identify common word pairs or groups.

3. Named Entity Recognition (NER):

- Named Entity Distribution: Bar chart of top named entities in titles and texts.

4. Part-of-Speech (POS) Analysis:

- POS Distribution: Bar chart of top POS tags in titles and texts.
- POS Co-occurrence: Heatmap of POS tag co-occurrences in titles and texts.

5. Topic Modeling:

- **LDA and LSA:** Identify underlying topics in texts and visualize using dimensionality reduction techniques (t-SNE or PCA).
- **Topic Coherence:** Calculate topic coherence scores to evaluate topic quality.

6. Association Rule Mining:

- **Apriori Algorithm:** Identify frequent itemsets and generate association rules.
- **With Topic Modeling and Without Topic Modeling.**
- **Rule Evaluation:** Evaluate rules using metrics like support, confidence, and lift.
- **Rule Visualization:** Use graph visualization techniques.

VI. Modeling:

1. Feature Engineering:

- TF-IDF (Term Frequency-Inverse Document Frequency):

- Converts textual data into numerical format.
- Effective for traditional machine learning models.

- GloVe Vectors (Global Vectors for Word Representation):

- Pre-trained word embeddings capturing semantic meanings.
- Suitable for deep learning models.

2. Traditional Machine Learning Models:

- Logistic Regression (LR):

- Simple, interpretable, and performs well on binary classification tasks.
- Uses TF-IDF features.

3. Ensemble Machine Learning Models:

- Random Forest (RF):

- Handles overfitting better than decision trees.
- Uses TF-IDF and word embeddings.

4. Deep Learning Models:

- Multi-Layer Perceptron (MLP):

- General-purpose, models non-linear relationships.
- Uses TF-IDF or dense word embeddings.

- Convolutional Neural Networks (CNN):

- Captures local patterns, effective for text classification.
- Uses word embeddings like GloVe.

- Recurrent Neural Networks (RNN):

- Effective for sequential data, captures temporal dependencies.
- Uses word embeddings as input.

- Long Short-Term Memory (LSTM):

- Addresses vanishing gradient problem in RNNs.
- Captures long-term dependencies using word embeddings.

- Transformers (Basic Implementation):

- Captures long-range dependencies with parallel processing.
- Uses token embeddings.

- BERT (Bidirectional Encoder Representations from Transformers - Base Uncased):

- Pre-trained model achieving state-of-the-art performance on NLP tasks.
- Fine-tuning BERT on the dataset for classification.