Introduction to Advanced EDA and Data Analysis

Exploratory Data Analysis (EDA) is a critical step in the data analysis process, aimed at understanding the structure, patterns, and potential insights within a dataset. This documentation covers advanced EDA techniques using Python, including text preprocessing, sentiment analysis, data visualization, and word embeddings. The code snippets are applied to a dataset of airline tweets to demonstrate their practical application.

Snippet 1: Text Preprocessing

Introduction:
Text preprocessing is an essential step in natural language processing. This snippet demonstrates how to clean and prepare text data for analysis by removing noise and standardizing text.

Use Case:
Text preprocessing is used when working with unstructured text data to improve the quality and relevance of textual information. It prepares text for tasks like sentiment analysis, text classification, or topic modeling.

Explanation:
1. Importing Libraries: Import necessary libraries for text preprocessing, including regular expressions (`re`), the Natural Language Toolkit (`nltk`), and modules for stopwords, word tokenization, and stemming.

2. Text Preprocessing Function: Define the `preprocess_text` function, which removes special characters, numbers, converts text to lowercase, tokenizes the text, removes stop words, and applies stemming.

3. Applying Preprocessing: Apply the preprocessing function to the 'text' column in the dataset, storing the preprocessed text in a new column, 'cleaned_text'.

Snippet 2: TF-IDF Vectorization

Introduction:
TF-IDF (Term Frequency-Inverse Document Frequency) vectorization is a technique to convert text data into numerical features for machine learning.

Use Case:
TF-IDF vectorization is commonly used for text analysis tasks like document classification and information retrieval, where text data needs to be transformed into a numerical format.

Explanation:
4. TF-IDF Vectorization: Employ the `TfidfVectorizer` from scikit-learn to create TF-IDF vectors from the cleaned text data. The feature space is limited to 5000 features, and the result is stored in the variable `X`.

Snippet 3: Sentiment Analysis

Introduction:
This section focuses on sentiment analysis, where the goal is to determine the sentiment (positive, negative, neutral) of text data.

Use Case:
Sentiment analysis is widely used to understand customer opinions, brand sentiment, and public reactions. It offers insights into the sentiment distribution in text data.

Explanation:
5. Splitting Data: Split the dataset into training and testing sets using `train_test_split`.

6. Logistic Regression Model: Create and train a Logistic Regression model using the training data.

7. Making Predictions: Utilize the trained model to predict sentiment labels on the test data.

8. Evaluating Accuracy: Calculate model accuracy by comparing predicted sentiment labels with actual labels in the test data.

9. Sentiment Distribution: Calculate the distribution of sentiment classes in the dataset.

10. Common Negative Reasons: Identify and count the most common reasons for negative sentiments among negative tweets.

11. Airline Sentiment Confidence: Analyze the relationship between airline sentiment and confidence by computing the mean confidence for each sentiment class.

12. Sentiment by Airline: Explore the sentiment distribution across different airlines.

Snippet 4: Word2Vec Model

Introduction:

Word2Vec is a technique for converting words into dense vector representations and discovering semantic relationships between words.

Use Case:
Word2Vec models are applied to various NLP tasks, including text similarity, document clustering, and recommendation systems.

Explanation:
13. Tokenization: Tokenize the cleaned text data to prepare it for training a Word2Vec model.

14. Word2Vec Model Training: Train a Word2Vec model using tokenized data. The model learns vector representations of words considering their context in sentences.

15. Similar Words: Find and print words similar to 'flight' based on the trained Word2Vec model.

Snippet 5: Data Visualization

Introduction:
Data visualization is a critical component of data analysis, enabling the exploration of data distribution, patterns, and insights.

Use Case:
Data visualization assists in understanding data distribution, relationships, and patterns. It is essential for exploratory data analysis (EDA) and effectively communicating findings.

Explanation:
16. Sentiment Distribution by Airline: Use Seaborn and Matplotlib to create a countplot showing the distribution of sentiment by airline.

17. Interactive Scatter Plot: Generate an interactive scatter plot using Plotly Express to visualize the relationship between retweet count, sentiment, and airline sentiment.

18. Interactive Box Plot: Create an interactive box plot to visualize the distribution of airline sentiment confidence across different airlines and sentiment classes.

19. Sentiment Distribution by Airline: Produce an interactive bar plot displaying sentiment distribution by airline.

20. Sentiment Breakdown by Airline: Utilize a sunburst chart to visualize the sentiment breakdown by airline.

21. Correlation Heatmap: Calculate the correlation matrix for numeric columns and create an interactive heatmap using Plotly Express to visualize correlations.