**1. Data Cleaning**

**a. Handle Missing Values**

**b. Remove Duplicates**

* Check for duplicate rows based on job title, company name, and description. If duplicates exist, keep only one instance.

**c. Standardize Text**

* Convert all text to lowercase for consistency.
* Remove special characters, extra spaces, and unnecessary formatting.
* Remove punctuation, special characters, and extra spaces.
* Normalize company names (e.g., "Microsoft Inc." vs. "Microsoft").

**d. Handle AI-Generated Data**

* Clearly identify and label AI-generated data. Consider balancing your dataset to avoid over-representing it.

**2. Data Wrangling**

**a. Feature Engineering**

1. **Text-Based Features**:
   * Extract keywords or phrases (e.g., "sponsorship available," "visa provided").
   * Compute text length (number of words or characters).
   * Calculate keyword frequency related to sponsorship.
2. **Location-Based Features**:
   * Create a region-based feature (e.g., "US," "Europe," etc.).
   * Identify if the location is in a sponsorship-friendly region.
3. **Company-Based Features**:
   * Group companies into categories (e.g., large tech companies, small startups).
4. **Label Encoding**:
   * Convert categorical labels into numerical values for machine learning.
     + Sponsorship provided → 1
     + Sponsorship not provided → 0
     + Uncertain → Exclude or analyze separately.

**b. Text Preprocessing for NLP**

* Tokenize job descriptions into words.
* Remove stop words, punctuation, and common uninformative terms (e.g., "apply," "job").
* Perform stemming or lemmatization.
* Use techniques like Term Frequency-Inverse Document Frequency (TF-IDF) or word embeddings (e.g., Word2Vec, GloVe, or BERT).

**3. Data Exploration**

**a. Summary Statistics**

* Check distributions of features:
  + Average word count in job descriptions.
  + Most frequent job titles, companies, and locations.
* Analyze label balance (e.g., how many "Sponsorship Provided" vs. "Not Provided").
* Get distributions for key fields (e.g., title, location)

**b. Label Analysis**

* Explore how often sponsorship is mentioned explicitly in descriptions.
* Investigate correlations between features (e.g., certain job titles or regions frequently offering sponsorship).

**c. Visualization**

* Plot the **distribution of job titles** and sponsorship labels.
* Visualize text-based insights:
  + Word clouds for "Sponsorship Provided" vs. "Not Provided."
  + Keyword frequencies using bar plots.
* Analyze location trends:
  + Use geospatial maps to see where sponsorships are more likely.

Create bar charts for most frequent job titles and locations.

**d. Outlier Detection**

* Identify unusually long or short descriptions.
* Check for abnormal keyword usage (e.g., excessive repetition of "visa").

**e. AI-Generated Data Quality**

* Compare distributions of AI-generated data with human-labeled data.
* Assess whether AI-generated examples introduce biases or unusual patterns.

**4. Handling the 'Uncertain' Label**

If "Uncertain" is significant, consider:

* Treating it as a separate class in a multi-class classification problem.
* Excluding these rows for a binary classification approach.
* Using a semi-supervised learning method to infer sponsorship from other features.

**5. Validation**

Ensure that:

* AI-generated and human-labeled data are split between training and test datasets to avoid information leakage.
* Use stratified sampling for train-test splits to maintain label balance.

**7. Correlations**

* Explore relationships between variables (e.g., certain job titles or locations often associated with sponsorships).

**1. Text-Based Features**

**Challenges**:

* AI-generated descriptions might lack the nuanced language patterns of real-world data.
* Over-reliance on AI-generated features could bias the model.

**Solutions**:

* **Keyword Extraction**: Ensure keywords like "sponsorship available" and "visa provided" are representative of real-world job postings. You can manually curate a keyword list from authentic data.
* **Text Length**: Analyze the text length distribution of real-world and AI-generated data separately to identify discrepancies.
* **Keyword Frequency**: Apply the same logic but avoid directly comparing frequencies between AI-generated and real-world data.

**2. Location-Based Features**

**Challenges**:

* AI-generated data might randomly assign locations, making it difficult to determine sponsorship-friendliness.

**Solutions**:

* Restrict location-based features to real-world data. For AI-generated data, mark these features as unknown or impute values based on the available dataset patterns. Set Location and Company name to be null
* If AI-generated data includes sponsorship-relevant locations, verify its consistency with real-world data before using it.

**3. Company-Based Features**

**Challenges**:

* AI-generated company names might not correspond to real companies or reflect their likelihood of offering sponsorship.

**Solutions**:

* Treat AI-generated companies separately. Assign them to a generic "AI-generated" category or exclude them from company-based feature engineering.
* Use AI-generated data primarily for text-related insights (e.g., sentiment, keyword analysis) and rely on real-world data for company-level patterns.

**4. Labeling AI-Generated Data**

**Why This Matters**:

* AI-generated data often mimics patterns in the training dataset, which might not fully represent real-world complexities.

**Solutions**:

* Create a binary feature, is\_ai\_generated, to distinguish between AI-generated and real-world data.
* Analyze model performance with and without AI-generated data to understand its impact.

**5. Strategies for Using AI-Generated Data**

* **Augmentation, Not Replacement**: Use AI-generated data to supplement areas where real-world data is sparse but avoid letting it dominate.
* **Focus on Text Patterns**: Use AI-generated data for general language modeling (e.g., keyword extraction, sentiment analysis) but treat features like location and company with skepticism.
* **Separate Model Training**: Train models separately on real-world and AI-generated data to observe performance differences and ensure the AI-generated data does not bias the model.

**Example Code for Exploration**

python

Copy code

import matplotlib.pyplot as plt

from wordcloud import WordCloud

# Analyze label distribution

label\_distribution\_real = real\_data['sponsorship\_available'].value\_counts()

label\_distribution\_ai = ai\_data['sponsorship\_available'].value\_counts()

# Visualize sponsorship categories for real vs AI-generated data

fig, ax = plt.subplots(1, 2, figsize=(12, 6))

label\_distribution\_real.plot(kind='bar', ax=ax[0], title='Real Data')

label\_distribution\_ai.plot(kind='bar', ax=ax[1], title='AI-Generated Data')

plt.show()

# Generate a word cloud for descriptions

wordcloud\_real = WordCloud().generate(' '.join(real\_data['description'].dropna()))

wordcloud\_ai = WordCloud().generate(' '.join(ai\_data['description'].dropna()))

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)

plt.imshow(wordcloud\_real, interpolation='bilinear')

plt.title('Real Data Descriptions')

plt.axis('off')

plt.subplot(1, 2, 2)

plt.imshow(wordcloud\_ai, interpolation='bilinear')

plt.title('AI-Generated Data Descriptions')

plt.axis('off')

plt.show()

This step-by-step approach ensures the is\_ai\_generated column is effectively leveraged for exploration and further preprocessing.

**Next Steps: Data Exploration, Visualization, and Analysis**

Once the initial data cleaning and preprocessing are complete, the focus should shift to exploring, visualizing, and analyzing the data to uncover patterns and insights. Here's a step-by-step guide to structuring these tasks:

**Sponsorship Insights**

* Analyze sponsorship distribution across top job titles:

python

Copy code

sponsorship\_titles = data.groupby('title')['sponsorship\_available'].value\_counts(normalize=True).unstack()

sponsorship\_titles.plot(kind='bar', stacked=True)

plt.title("Sponsorship Availability Across Job Titles")

plt.show()

**AI-Generated vs Real Data**

* Compare word\_count between AI-generated and real records:

python

Copy code

sns.boxplot(data=data, x='is\_ai\_generated', y='word\_count')

plt.title("Word Count Distribution: AI-Generated vs Real Data")

plt.show()

**Keyword Analysis**

* Investigate sponsorship-related keywords (e.g., "visa", "sponsorship"):

python

Copy code

def keyword\_presence(text):

keywords = ['visa', 'sponsorship', 'relocation']

return any(keyword in text.lower() for keyword in keywords)

data['contains\_keyword'] = data['description'].apply(keyword\_presence)

sns.countplot(data=data, x='contains\_keyword', hue='sponsorship\_available')

plt.title("Sponsorship by Keyword Presence")

plt.show()

**2**

**3. Frequent Words**

* **Objective**: Find the most commonly used words.
* **Examples**:
  + Create a word frequency table or visualization (e.g., bar chart, word cloud).
  + Identify overly repeated words that might skew analysis.
* **Usage**: Highlights the dominant themes and potential stopwords.

**4. Bigram and Trigram Analysis**

* **Objective**: Examine common word pairs or triplets.
* **Examples**:
  + Phrases like *"machine learning"* or *"job opportunity"*.
* **Usage**: Helps identify context-specific terms or phrases for further focus.

**6. Stopword Prevalence**

* **Objective**: Measure how much of the text consists of common stopwords.
* **Examples**:
  + Ratio of stopwords to total words.
* **Usage**: Indicates whether removing stopwords could significantly alter the dataset.

**7. Sentiment Overview**

* **Objective**: Gauge the general sentiment of the text before processing.
* **Examples**:
  + Basic sentiment scoring to identify positive, negative, or neutral entries.
* **Usage**: Provides a baseline for sentiment analysis.

**8**

**9. Contextual Themes**

* **Objective**: Identify overarching topics or themes.
* **Examples**:
  + Keywords related to specific domains (e.g., "job," "hiring," "AI").
* **Usage**: Provides insight into the dataset’s primary focus areas.

**10. Distribution of Labels (if available)**

* **Objective**: Examine the balance of target labels (e.g., "Sponsorship Provided," "Not Provided").
* **Examples**:
  + Ratio of positive to negative samples.
* **Usage**: Ensures balanced training data for models.

**Integrated Workflow:**

1. **Importing Data**
   * Load the dataset from its source into your environment.
2. **Fixing Data Types**
   * Ensure that all columns have the correct data types (e.g., strings for text, integers for numerical features). This step ensures that subsequent operations behave as expected.
3. **Initial Data Exploration (Before Cleaning)**
   * Perform exploratory analysis on the raw text to identify patterns:
     + Word count, text length distribution.
     + High-frequency words and potential stopwords.
     + Sentiment distribution (if applicable).
4. **Data Cleaning**
   * Remove duplicates and handle missing values.
   * Identify and handle anomalies in the dataset (e.g., unrealistic values or empty records).
5. **Data Labeling and Imbalance Handling**
   * If the dataset has labeled data (e.g., sponsorship categories), analyze class distribution and apply techniques to handle imbalances, such as oversampling or undersampling.
6. **Data Wrangling**
   * Transform and reshape the data for analysis:
     + Tokenize the text into words or phrases.
     + Apply stemming or lemmatization to standardize word forms.
     + Remove stopwords *after* lemmatization for accuracy.
     + Encode categorical variables, such as sponsorship categories, using techniques like one-hot encoding.
7. **Refined Statistical Analysis and Visualization**
   * After preprocessing, revisit visualization and statistical analysis:
     + Analyze token frequencies and cleaned text for normalized insights.
     + Visualize relationships between features in the dataset.
8. **Final Data Preparation**
   * Ensure the dataset is fully formatted and ready for machine learning or further analysis:
     + Aggregate or scale features if necessary.
     + Perform final checks on data consistency.

**Key Adjustments:**

* Perform **stopword removal after lemmatization** for accuracy.
* Conduct **initial visualizations before preprocessing** for high-level insights, then follow up with **refined visualizations post-cleaning** for normalized trends.
* Integrate **data wrangling after cleaning and before modeling** to ensure the data is in its most usable form.

**Workflow for Initial Data Exploration**

1. **Load the Raw Dataset**
   * Import the dataset and inspect its structure (e.g., columns, data types, and missing values).
2. **Examine Text Lengths**
   * Analyze the distribution of text lengths to identify overly short or long entries.
3. **Generate Word Frequency Distributions**
   * Create word clouds or bar charts to visualize the most frequently occurring words.
4. **Identify Stopwords**
   * Examine high-frequency words to detect potential stopwords that may need removal.
5. **Check for Anomalies**
   * Look for irregular patterns, such as unexpected characters, duplicate entries, or non-relevant text.
6. **Basic Statistical Analysis**
   * Calculate summary statistics such as average word count or sentence length.
7. **Document Insights**
   * Record observations to guide data cleaning and wrangling steps.