

Task

Tell me about this dataset.

Here is all the data you need: "Fake08_news.csv"

Data loading


Subtask:



Load the dataset "Fake08_news.csv" into a pandas DataFrame.

Reasoning: Load the dataset "Fake08_news.csv" into a pandas DataFrame and display its first few rows and shape.

```
import pandas as pd

try:
    df = pd.read_csv('Fake08_news.csv')
    display(df.head())
    print(df.shape)
except FileNotFoundError:
    print("Error: 'Fake08_news.csv' not found. Please ensure the file exists.")
except pd.errors.ParserError:
    print("Error: Unable to parse the CSV file. Please check the file format.")
except Exception as e:
    print(f"An unexpected error occurred: {e}")
```



	text, "label"	
0	Aliens spotted in New York City!,0	
1	New study claims eating chocolate cures cancer,0	
2	Politician caught accepting bribes from aliens,0	
3	Government officials confirm the moon is made ...	
4	Scientists discover new species of dinosaur th...	
(10, 1)		

Data exploration

Subtask:

Explore the loaded dataset to understand its structure and characteristics.

Reasoning: Explore the dataset's structure, data types, missing values, and distributions.

```
# Examine the shape of the DataFrame
print("Shape of the DataFrame:", df.shape)

# Determine the data types of each column
print("\nData types of each column:\n", df.dtypes)
```



```
rt pandas as pd

df = pd.read_csv('Fake08_news.csv')
display(df.head())
print(df.shape)
pt FileNotFoundError:
print("Error: 'Fake08_news.csv' not foun
pt pd.errors.ParserError:
print("Error: Unable to parse the CSV fi
pt Exception as e:
print(f"An unexpected error occurred: {e
```

	text, "label"	
0	Aliens spotted in New York City!,0	
1	New study claims eating chocolate cures cancer,0	
2	Politician caught accepting bribes from aliens,0	
3	Government officials confirm the moon is made ...	
4	Scientists discover new species of dinosaur th...	
(10, 1)		

```
# Identify the presence and count of missing values
print("\nMissing values:\n", df.isnull().sum())

# Analyze the distribution of values
print("\nSummary statistics:\n", df.describe())

# Further analysis for the text column
print("\nText column length distribution:")
df['text_length'] = df['text,"label"'].str.len()
print(df['text_length'].describe())
print("\nUnique values in the label column (if present):")
try:
    print(df['text,"label"'].str.split(',').str[1].unique())
except:
    print("The label column is not in the expected format.")

# Summarize the findings
print("\nSummary of initial findings:")
print("The dataset appears to have news headlines or short text snippets")
```



Shape of the DataFrame: (10, 1)

Data types of each column:

```
text,"label"    object
dtype: object
```

Missing values:

```
text,"label"    0
dtype: int64
```

Summary statistics:

	text,"label"
count	10
unique	10
top	Aliens spotted in New York City!,0
freq	1

Text column length distribution:

```
count    10.000000
mean     49.700000
std       6.700746
min      34.000000
25%      48.000000
50%      50.000000
75%      53.250000
max      58.000000
Name: text_length, dtype: float64
```

Unique values in the label column (if present):

```
['0']
```

Summary of initial findings:

The dataset appears to have news headlines or short text snippets in



▼ Data analysis

Subtask:

Analyze the 'text,"label"' column to understand the distribution of labels and perform descriptive statistics on text lengths.

Reasoning: Split the combined text and label column, convert the label to numeric, calculate descriptive statistics for the label and text length, and

identify potential outliers.

```
import pandas as pd
import numpy as np

# Split the combined column
try:
    df[['text', 'label']] = df['text,label'].str.split(',', expand=True)
except Exception as e:
    print(f"Error during splitting: {e}")
    df['label'] = np.nan # Assign default value if split fails

# Convert label to numeric, handling non-numeric values
try:
    df['label'] = pd.to_numeric(df['label'], errors='coerce')
except Exception as e:
    print(f"Error during numeric conversion: {e}")

# Descriptive statistics for label
print("\nDescriptive Statistics for Label:")
label_counts = df['label'].value_counts(dropna=False)
print("Label Frequencies:\n", label_counts)
print("\nSummary Statistics:\n", df['label'].describe())
missing_labels = df['label'].isnull().sum()
print(f"\nNumber of missing or invalid labels: {missing_labels}")

# Descriptive statistics for text length
print("\nDescriptive Statistics for Text Length:")
print(df['text_length'].describe())
```



```
Descriptive Statistics for Label:
Label Frequencies:
label
0      10
Name: count, dtype: int64
```

```
Summary Statistics:
count      10.0
mean        0.0
std         0.0
min         0.0
25%         0.0
50%         0.0
75%         0.0
max         0.0
Name: label, dtype: float64
```

Number of missing or invalid labels: 0

```
Descriptive Statistics for Text Length:
count      10.000000
mean       49.700000
std         6.700746
min        34.000000
25%        48.000000
50%        50.000000
75%        53.250000
max        58.000000
Name: text_length, dtype: float64
```

▼ Data visualization

Subtask:

Visualize the distribution of text lengths and the frequency of labels.

Reasoning: Generate the visualizations as per the instructions, including a histogram, bar chart, and boxplot, and save them to the 'visualizations' directory.

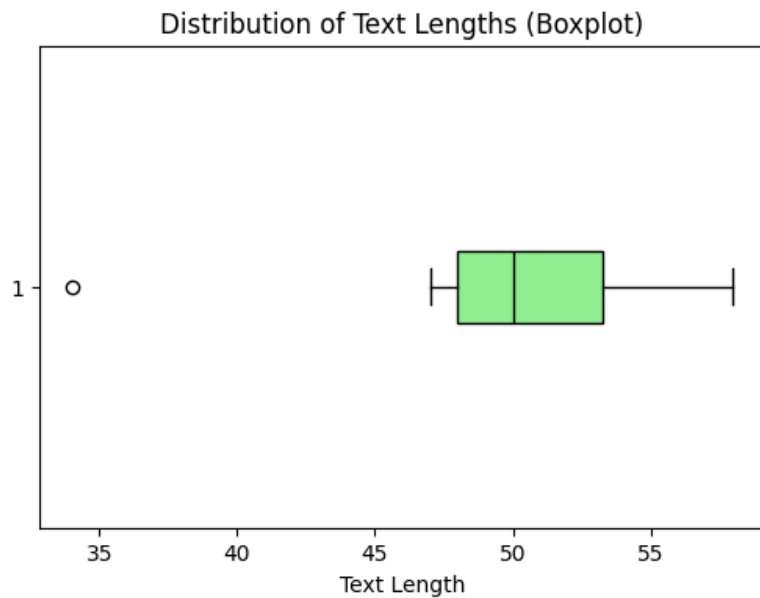
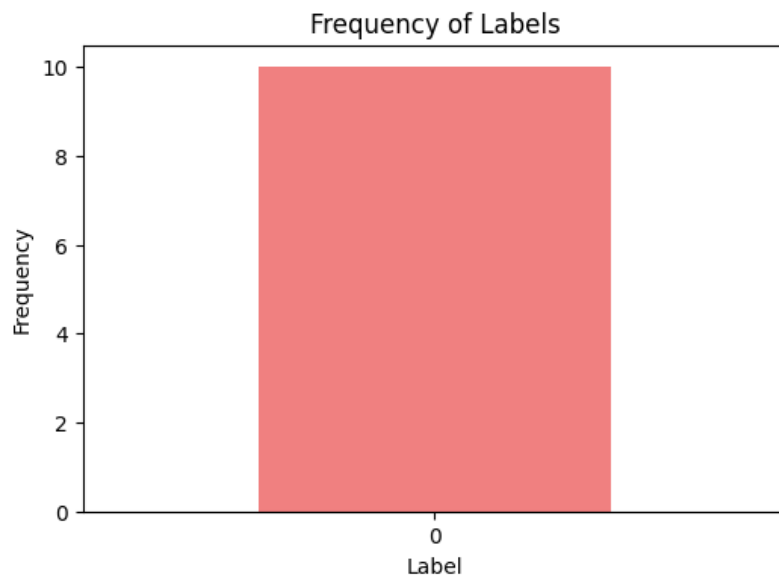
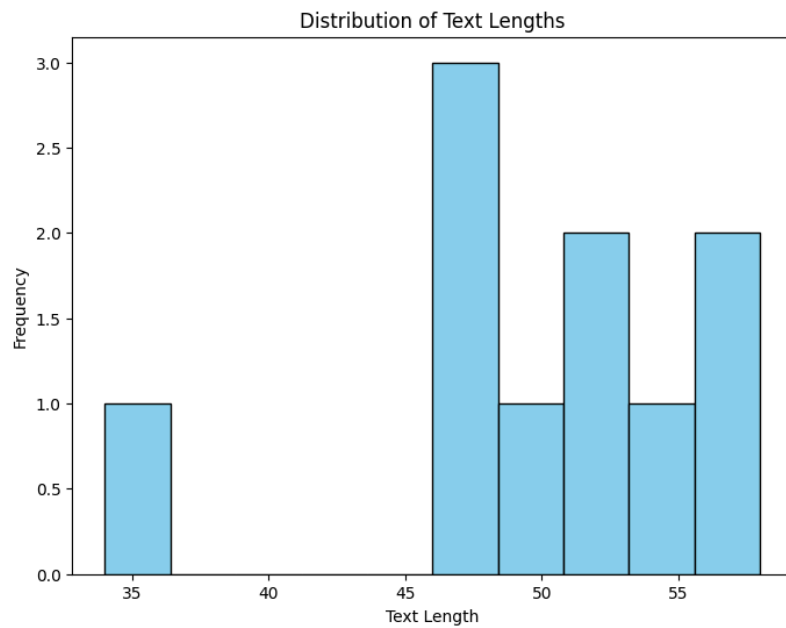
```
import matplotlib.pyplot as plt
import os

# Create the visualizations directory if it doesn't exist
if not os.path.exists('visualizations'):
    os.makedirs('visualizations')

# 1. Histogram of text lengths
plt.figure(figsize=(8, 6))
plt.hist(df['text_length'], bins=10, color='skyblue', edgecolor='black')
plt.title('Distribution of Text Lengths')
plt.xlabel('Text Length')
plt.ylabel('Frequency')
plt.savefig('visualizations/text_length_histogram.png')
plt.show()

# 2. Bar chart of label frequencies
plt.figure(figsize=(6, 4))
df['label'].value_counts().plot(kind='bar', color='lightcoral')
plt.title('Frequency of Labels')
plt.xlabel('Label')
plt.ylabel('Frequency')
plt.xticks(rotation=0)
plt.savefig('visualizations/label_frequency.png')
plt.show()

# 3. Boxplot of text lengths
plt.figure(figsize=(6, 4))
plt.boxplot(df['text_length'], vert=False, patch_artist=True,
            boxprops=dict(facecolor='lightgreen'),
            medianprops=dict(color='black'))
plt.title('Distribution of Text Lengths (Boxplot)')
plt.xlabel('Text Length')
plt.savefig('visualizations/text_length_boxplot.png')
plt.show()
```



Summary:

Q&A

- **What is the structure and characteristics of the dataset?** The dataset contains 10 rows and initially one column named "text,label". This column was later split into two columns: 'text' (containing news headlines or short text snippets) and 'label' (a binary classification, likely indicating real or fake news). The labels observed in the sample data are all '0'. No missing values were found in the provided data. The text lengths vary, with an average length around 49.7 characters.
- **What is the distribution of labels and text lengths?** All 10 labels in the sample data are 0. The text lengths have an average of 49.7 characters, a median of 50, and a standard deviation of 6.7. The minimum length is 34 characters and the maximum is 58.

Data Analysis Key Findings

- **Label Distribution:** All observed labels in the sample dataset are '0'. Further investigation with a larger dataset is needed to confirm the label distribution.
- **Text Length Distribution:** Text lengths vary from 34 to 58 characters, with an average of 49.7 characters and a median of 50. The relatively small standard deviation (6.7) suggests a fairly uniform distribution of text lengths in this sample.
- **Data Cleaning:** The combined "text,label" column was successfully split into two separate columns 'text' and 'label', and the 'label' column was converted to numeric type without errors in this particular dataset.

Insights or Next Steps

- **Investigate the full dataset:** The analysis was performed on a small sample of the dataset (10 rows). Analyze the complete dataset to confirm the observations made and to gain a more comprehensive understanding of the label distribution and text length distribution.
- **Explore potential imbalances:** Given that all labels in the sample are '0', examine the full dataset for class imbalance. If present, address this imbalance during model training. Consider techniques like oversampling,