

Name : Diya Rathod

Roll No: ET2-41

PRN No : 202401070144

Class: ET2

SUB: EDS

Dataset :

The Blog Authorship Corpus

Read Dataset:

The screenshot displays a Google Colab environment. At the top, the browser address bar shows the Colab session URL. The Colab interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with options like 'Share' and 'Gemin'. The main workspace is divided into a code editor and a table viewer.

Code Editor: The code consists of two cells. The first cell imports pandas as pd. The second cell defines a path to a CSV file in Google Drive and reads it into a DataFrame named df, then displays the first five rows using df.head().

```
[3] import pandas as pd


path="/content/drive/MyDrive/python/blogtext.csv"
df=pd.read_csv(path)
df.head()
```


Table Viewer: The output of the code is a table with 5 rows and 8 columns: id, gender, age, topic, sign, date, and text. The data is as follows:

| | id | gender | age | topic | sign | date | text |
|---|---------|--------|-----|-------------------|----------|--------------|--|
| 0 | 2059027 | male | 15 | Student | Leo | 14,May,2004 | Info has been found (+/- 100 pages,... |
| 1 | 2059027 | male | 15 | Student | Leo | 13,May,2004 | These are the team members: Drewe... |
| 2 | 2059027 | male | 15 | Student | Leo | 12,May,2004 | In het kader van kernfusie op aarde... |
| 3 | 2059027 | male | 15 | Student | Leo | 12,May,2004 | testing!!! testing!!! |
| 4 | 3581210 | male | 33 | InvestmentBanking | Aquarius | 11,June,2004 | Thanks to Yahoo!'s Toolbar I can ... |

The right sidebar shows the 'Files' section with a tree view of the Google Drive contents, including folders like 'Colab Notebooks', 'Document', 'English', and 'Prabhar yadi G.P.mohari', and files like 'blogtext.csv' and 'Coefficient Of Restitution...'. A 'Table of contents' tab is also visible.

1. What is the shape of the dataset (rows, columns)?

✓ 0s  `df.shape`

 (681284, 7)

2. What are the columns available in the dataset?

`dtype: object`

✓ 0s  `df.columns`


 Index(['id', 'gender', 'age', 'topic', 'sign', 'date', 'text'], dtype='object')


3. What is the data type of each column?


CO EDS-1.ipynb ☆


File Edit View Insert Runtime Tools Help

Q Commands + Code + Text

0s  `df.shape`

 (681284, 7)

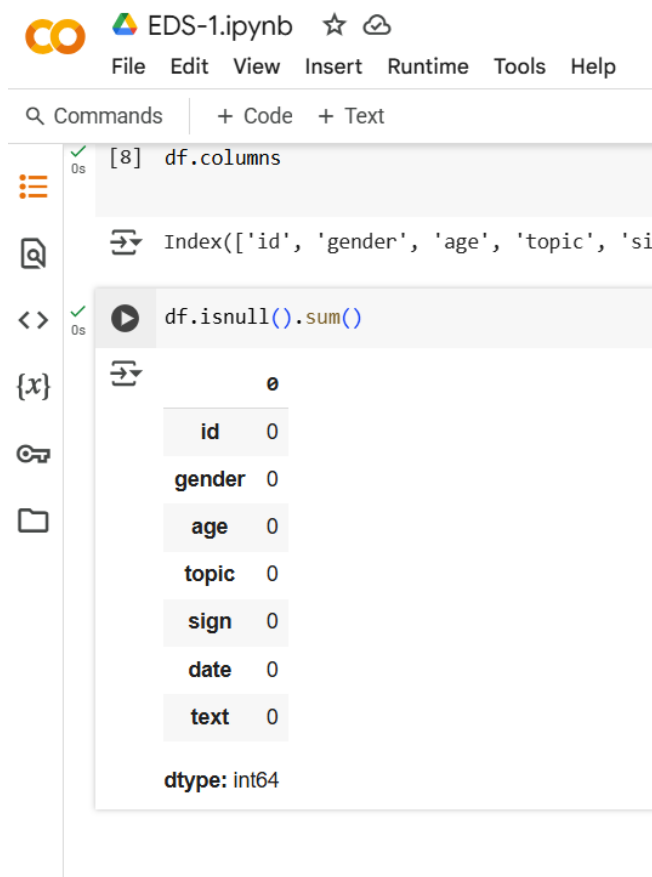
0s  `df.dtypes`

 `0`

| | |
|---------------------|---------------------|
| <code>id</code> | <code>int64</code> |
| <code>gender</code> | <code>object</code> |
| <code>age</code> | <code>int64</code> |
| <code>topic</code> | <code>object</code> |
| <code>sign</code> | <code>object</code> |
| <code>date</code> | <code>object</code> |
| <code>text</code> | <code>object</code> |

`dtype: object`

4. How many missing values are in each column?



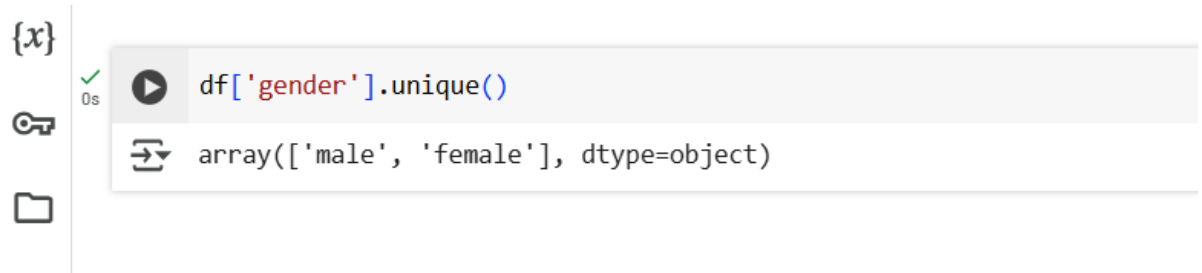
The screenshot shows a Jupyter Notebook interface with the following elements:

- Top bar: CO logo, EDS-1.ipynb, and icons for star and cloud.
- Menu bar: File, Edit, View, Insert, Runtime, Tools, Help.
- Search bar: Q Commands, + Code, + Text.
- Left sidebar: Icons for table of contents, search, expand/collapse, variables, key, and file explorer.
- Code cell: `[8] df.columns` (executed, 0s).
- Output: `Index(['id', 'gender', 'age', 'topic', 'si`
- Code cell: `df.isnull().sum()` (executed, 0s).
- Output: A table showing the count of missing values for each column.

| | 0 |
|--------|---|
| id | 0 |
| gender | 0 |
| age | 0 |
| topic | 0 |
| sign | 0 |
| date | 0 |
| text | 0 |

dtype: int64

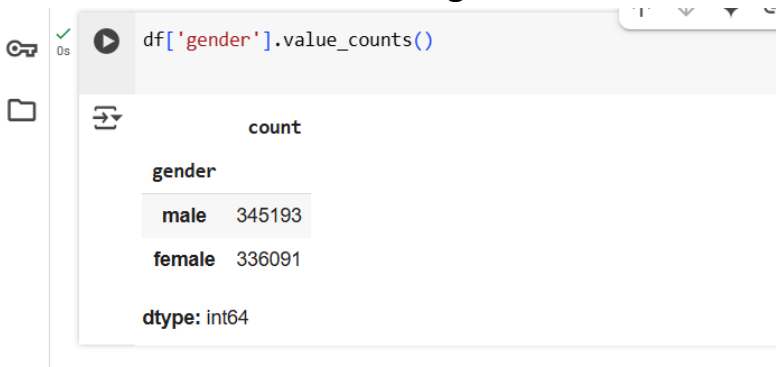
5. What are the unique genders available?



The screenshot shows a Jupyter Notebook interface with the following elements:

- Left sidebar: Icons for table of contents, search, expand/collapse, variables, key, and file explorer.
- Code cell: `df['gender'].unique()` (executed, 0s).
- Output: `array(['male', 'female'], dtype=object)`

6. What is the distribution of gender?




The screenshot shows a Jupyter Notebook interface with the following elements:


- Left sidebar: Icons for table of contents, search, expand/collapse, variables, key, and file explorer.
- Code cell: `df['gender'].value_counts()` (executed, 0s).
- Output: A table showing the count of each gender.

| | count |
|--------|--------|
| gender | |
| male | 345193 |
| female | 336091 |


dtype: int64


7. What is the minimum and maximum age of bloggers?

✓ 0s  `df['age'].min(), df['age'].max()`

 (13, 48)

8. What are the most common blogging topics?


<> ✓ 0s  `df['topic'].value_counts().head(5)`




| topic | count |
|------------|--------|
| indUnk | 251015 |
| Student | 153903 |
| Technology | 42055 |
| Arts | 32449 |
| Education | 29633 |

dtype: int64

9. How many distinct topics are there?

✓ 0s  `df['topic'].nunique()`

 40

10. What are the astrological signs available?

```
> ✓ 0s df['sign'].unique()  
x}  
array(['Leo', 'Aquarius', 'Aries', 'Capricorn', 'Gemini', 'Cancer',  
      'Sagittarius', 'Scorpio', 'Libra', 'Virgo', 'Taurus', 'Pisces'],  
      dtype=object)
```

11. How many blog posts are written under the "Student" topic?

```
df[df['topic'] == 'Student'].shape[0]  
153903
```

12. What is the average age of bloggers?

```
df['age'].mean()  
np.float64(23.932326313255558)
```

13. What is the oldest blogger's gender and topic?

0s

```
df[df['age'] == df['age'].max()][['gender', 'topic']]
```

| | gender | topic |
|--------|--------|----------------------|
| 19090 | male | Communications-Media |
| 19091 | male | Communications-Media |
| 19092 | male | Communications-Media |
| 19093 | male | Communications-Media |
| 19094 | male | Communications-Media |
| ... | ... | ... |
| 672775 | male | Museums-Libraries |
| 672776 | male | Museums-Libraries |
| 672777 | male | Museums-Libraries |
| 672778 | male | Museums-Libraries |
| 672779 | male | Museums-Libraries |

3572 rows × 2 columns

14. What is the shortest blog post?

0s

```
df['text'].str.len().min()
```

4

15. What is the longest blog post?

0s

```
df['text'].str.len().max()
```

790123

16. How many posts were made by Aries bloggers?




The image shows a Jupyter Notebook interface. On the left, there is a sidebar with icons for a file explorer, a key, and a folder. The main area contains a code cell with a play button icon and a green checkmark. The code is `df[df['sign'] == 'Aries'].shape[0]`. Below the code cell, there is an output cell showing the result `64979`.

```
df[df['sign'] == 'Aries'].shape[0]
```

64979

17. Who writes longer posts on average: male or female?



The image shows a Jupyter Notebook interface. On the left, there is a sidebar with icons for a file explorer, a key, and a folder. The main area contains a code cell with a play button icon and a green checkmark. The code is `df['text_length'] = df['text'].str.len()` and `df.groupby('gender')['text_length'].mean()`. Below the code cell, there is an output cell showing the result as a Series with two categories: 'female' and 'male', with their respective mean text lengths.

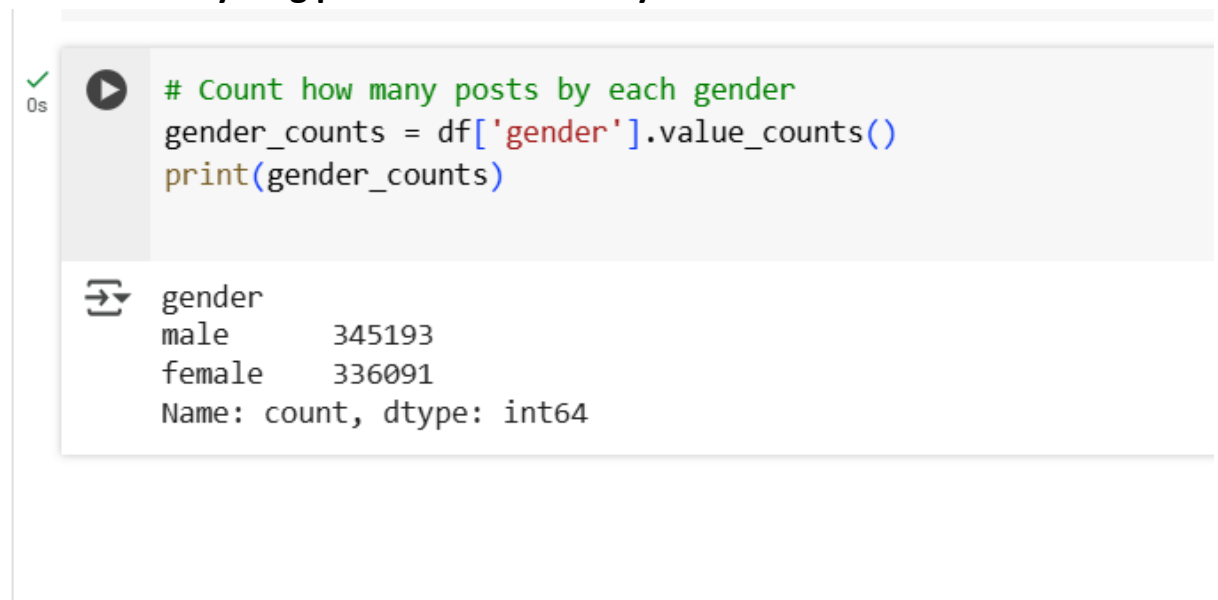
```
df['text_length'] = df['text'].str.len()
df.groupby('gender')['text_length'].mean()
```

text_length

| gender | text_length |
|--------|-------------|
| female | 1140.986477 |
| male | 1101.009021 |

dtype: float64

18. How many blog posts were written by male vs female authors?



The image shows a Jupyter Notebook interface. On the left, there is a sidebar with icons for a file explorer, a key, and a folder. The main area contains a code cell with a play button icon and a green checkmark. The code is `# Count how many posts by each gender`, `gender_counts = df['gender'].value_counts()`, and `print(gender_counts)`. Below the code cell, there is an output cell showing the result as a Series with two categories: 'male' and 'female', with their respective counts.

```
# Count how many posts by each gender
gender_counts = df['gender'].value_counts()
print(gender_counts)
```

gender

| | |
|--------|--------|
| male | 345193 |
| female | 336091 |

Name: count, dtype: int64

19. Which age group has written the most blog posts?

```
0s # Create age groups
df['age_group'] = pd.cut(df['age'], bins=[10, 20, 30, 40, 50, 60], labels=['10s', '20s', '30s', '40s', '50s'])

# Find most common age group
most_common_age_group = df['age_group'].value_counts().idxmax()
print(f'Most common age group: {most_common_age_group}')
```

Most common age group: 20s

20. Find the average age of all bloggers.

```
0s # Calculate average age
average_age = df['age'].mean()
print(f'Average age of bloggers: {average_age}')
```

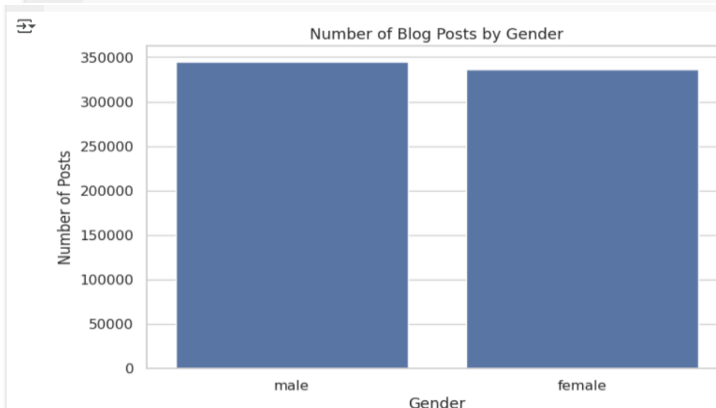
Average age of bloggers: 23.932326313255558

21. Plot the number of blog posts by gender.

```
1s import matplotlib.pyplot as plt
import seaborn as sns

# Set the plot style
sns.set(style='whitegrid')

# Plot
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='gender')
plt.title('Number of Blog Posts by Gender')
plt.xlabel('Gender')
plt.ylabel('Number of Posts')
plt.show()
```



22. Plot the distribution of bloggers' ages.

```
# Plot
plt.figure(figsize=(10, 6))
sns.histplot(data=df, x='age', bins=20, kde=True)
plt.title('Distribution of Bloggers\' Ages')
plt.xlabel('Age')
plt.ylabel('Number of Bloggers')
plt.show()
```

12

