Name : Madhura Patil
Roll No: CS2-20
PRN : 202401040094

EDS Theory Activity 1

Paper Review Dataset : 20 problem statements

Problem Statements

```
[ ] import pandas as pd
    import numpy as np

    df = pd.read_excel('reviews_converted.xlsx')
```

```
169
```

1. How many unique papers are in the dataset?

```
[7] df['Paper_ID'].nunique()
```

```
169
```

2. How many reviews were written in each language?

```
[8] df['Language'].value_counts()
```

Language	count
es	388
en	17

dtype: int64

3. What is the average confidence score per language?

```
df.groupby('Language')['Confidence'].mean()
```

Language	Confidence
en	3.705882
es	3.567358

dtype: float64

4. Find the paper with the highest average evaluation score.

```
[10] df.groupby('Paper_ID')['Evaluation'].mean().idxmax()
```

```
np.int64(2)
```

5. What percentage of papers were preliminarily accepted vs. rejected?

```
[11] df['Preliminary_Decision'].value_counts(normalize=True) * 100
```

Preliminary_Decision	proportion
accept	64.691358
reject	30.123457
probably reject	4.938272
no decision	0.246914

dtype: float64

6. How many reviews have missing remarks?

```
[12] df['Remarks'].isna().sum()
```

```
np.int64(294)
```

7. Calculate the mean and standard deviation of confidence scores.

```
[13] df['Confidence'].agg(['mean', 'std'])
```

	Confidence
mean	3.573201
std	0.844341

dtype: float64

8. How many reviews were submitted on each date?

```
[14] df['Timespan'] = pd.to_datetime(df['Timespan'])
     df['Timespan'].value_counts().sort_index()
```

Timespan	count
2010-07-05	117
2013-07-05	82
2014-07-05	126
2015-07-05	80

dtype: int64

9. Identify the most common evaluation score.

```
[15] df['Evaluation'].mode()[0]
```

```
np.int64(2)
```

10.Find papers with at least one review where confidence is 5.

```
[16] df[df['Confidence'] == 5]['Paper_ID'].unique()
```

```
array([  1,   5,   7,   8,   9,  11,  16,  27,  29,  32,  44,  53,  54,
        55,  58,  59,  63,  66,  72,  73,  76,  78,  82,  93,  94,  95,
        99, 103, 112, 117, 127, 129, 141, 142, 143, 155, 164, 165, 167,
       170])
```

11. Get average orientation per decision type.

```
[17] df.groupby('Preliminary_Decision')['Orientation'].mean()
```

Preliminary_Decision	Orientation
accept	0.179389
no decision	-1.000000
probably reject	-0.800000
reject	-0.950820

dtype: float64

12. Find the paper with the most reviews.

```
[18] df['Paper_ID'].value_counts().idxmax()
```

```
np.int64(128)
```

13. Determine if there is any correlation between confidence and evaluation.

```
[19] df[['Confidence', 'Evaluation']].corr()
```

	Confidence	Evaluation
Confidence	1.000000	-0.038315
Evaluation	-0.038315	1.000000

14. List papers that have conflicting preliminary decisions (if any). (Only possible if decisions differ per review — may not apply here if uniform per paper.)

```
[20] df.groupby('Paper_ID')['Preliminary_Decision'].nunique().gt(1).sum()
```

```
np.int64(0)
```

15. How many reviews contain non-null review text?

```
[21] df['Text'].notna().sum()
```

```
np.int64(399)
```

16. Find the average number of words per review.

```
[22] df['Text'].dropna().apply(lambda x: len(x.split())).mean()
```

```
np.float64(162.9298245614035)
```

17. Which paper has the longest average review text (by word count)?

FPS N/A | GPU 0 % | CPU 26 % | LAT N/A

```
[24] df['word_count'] = df['Text'].dropna().apply(lambda x: len(x.split()))
     df.groupby('Paper_ID')['word_count'].mean().idxmax()
```

```
np.int64(136)
```

18. What is the distribution of orientation values?

```
df['Orientation'].value_counts()
```

Orientation	count
-1	142
0	117
1	96
-2	35
2	15

dtype: int64

19. Identify the top 5 most frequent words across all reviews (after lowercasing).

```
[26] from collections import Counter
     words = ' '.join(df['Text'].dropna().str.lower()).split()
     Counter(words).most_common(5)
```

```
[('de', 4350), ('la', 2309), ('el', 2053), ('en', 1937), ('que', 1398)]
```

Code cell output actions

20. What proportion of reviews gave the highest evaluation score?

```
[27] max_eval = df['Evaluation'].max()
     (df['Evaluation'] == max_eval).mean() * 100
```

```
np.float64(26.913580246913583)
```