Name - Preeti S. Mittal

PRN - 202401100137

Class - CS6-11

Kaggle Dataset Link:

https://www.kaggle.com

Text Classification Dataset - Problems & Solutions

1. Problem - Total number of records in the dataset.

Solution:

```
total_records = df.shape[0]
```

2. Problem - Number of unique categories in the Category column.

Solution:

```
unique_categories = df['Category'].nunique()
```

3. Problem - Count of images per Category.

Solution:

```
category_counts = df['Category'].value_counts()
```

4. Problem - Average Blur_Level across all images.

Solution:

```
avg_blur = df['Blur_Level'].mean()
```

5. Problem - Maximum Brightness_Level and corresponding image.

Solution:

```
max_brightness = df['Brightness_Level'].max()
brightest_image = df.loc[df['Brightness_Level'].idxmax(), 'Image_ID']
```

6. Problem - Count of images labeled as 'Incorrect'.

Solution:

```
(df['Label'] == 'Incorrect').sum()
```

7. Problem - Average number of Solution_Steps for math problems.

Solution:

```
avg_steps = df[df['Category'] == 'Handwritten Math Solutions']['Solution_Steps'].mean(
```

8. Problem - Unique sign languages used.

Solution:

```
unique_languages = df['Sign_Language'].dropna().unique()
```

9. Problem - Image with the highest Anomaly_Score.

Solution:

```
max_anomaly = df['Anomaly_Score'].max()
anomaly_image = df.loc[df['Anomaly_Score'].idxmax(), 'Image_ID']
```

10. Problem - Records with Bounding_Box_Width > 150.

Solution:

```
(df['Bounding_Box_Width'] > 150).sum()
```

11. Problem - Standard deviation of Contrast_Level.

Solution:

```
contrast_std = df['Contrast_Level'].std()
```

12. Problem - Correlation between Brightness_Level and Noise_Level.

Solution:

```
correlation = df['Brightness_Level'].corr(df['Noise_Level'])
```

13. Problem - Missing values in the Equation column.

Solution:

```
missing_equations = df['Equation'].isna().sum()
```

14. Problem - Replace missing Sign_Translation with 'Unknown'.

Solution:

```
df['Sign_Translation'] = df['Sign_Translation'].fillna('Unknown')
```

15. Problem - Category with highest average Blur_Level.

Solution:

```
avg_blur_by_category = df.groupby('Category')['Blur_Level'].mean().idxmax()
```

16. Problem - Entries labeled 'Uncertain' with Brightness_Level < 0.5.

Solution:

```
df[(df['Label'] == 'Uncertain') & (df['Brightness_Level'] < 0.5)].shape[0]
```

17. Problem - Add new column Box_Area = Width × Height.

Solution:

```
df['Box_Area'] = df['Bounding_Box_Width'] * df['Bounding_Box_Height']
```

18. Problem - Median Box_Area value.

Solution:

```
median_box_area = df['Box_Area'].median()
```

19. Problem - Average Confidence_Score per Label.

Solution:

```
avg_confidence_by_label = df.groupby('Label')['Confidence_Score'].mean()
```

20. Problem - Equations containing the term 'x^2'.

Solution:

```
x2_count = df['Equation'].fillna('').str.contains('x\^2').sum()
```