Data Visualization of the Bengali Song Mood Predictor Dataset

Proper Data visualization of the dataset allows us to gain insights into the distribution of data points, relationships between features, and any potential outliers or patterns that might influence the model.

The dataset consists of a collection of Bengali songs, each labelled with a corresponding mood category. The mood categories may include labels like "Happy," "Sad," which is denoted as 1, 0 respectively. The dataset is organized in a tabular format, with each row representing a song and each column representing a feature or attribute of the song.

Attributes included in the dataset:

- **No**: Serial Number of each Bengali Songs
- **Mean**: The Mean of the Song required for statistical analysis
- **Standard Deviation**: The Standard Deviation of the Song required for statistical analysis
- **BPM:** In the language of music, the phrase "beats per minute" (BPM) the number of beats in one minute.
- **KEY**: That musical note/pitch or chord on which we build or center a song.
- **Accousticness:** This value describes how acoustic a song is.
- **Danceability**: Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.01 is least danceable and 1.0 is most danceable

- **Energy:** Energy represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy.
- **Instrumentalness:** This value represents the amount of vocals in the song. The closer it is to 1.0, the more instrumental the song is.
- **Liveness:** This value describes the probability that the song was recorded with a live audience.
- **Loudness:** The parameter to calculate how loud is the song.
- **Valence:** A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative
- **Popularity:** It defines how much the song is popular among the audience.
- **Genre / Mood:** The mood categories may include labels like "Happy," "Sad," which is denoted as 1, 0 respectively

Procedure:

Step 1: Importing all the important python libraries like **Numpy, Pandas, Seaborn.** And loading the csv file.

```
import numpy as np
import pandas as pd
import seaborn as sn
df=pd.read_csv("train.1v.csv")
```

Step 2: Checking for null values and calculating the sum of NULL values.

```
df.isnull().sum()
```

```
no                      0
Mean                    7
Standard Deviation      7
BPM                     0
KEY                     0
Accousticness           0
Danceability            0
Energy                  0
Instrumentalness        0
Liveness                0
Loudness(in %)          1
Valence                 0
Popularity              0
Loudness(in dB)         9
Genre / Mood            0
dtype: int64
```

Step 3: The first 5 rows of the data set.

```
df.head()
```

no	Mean	Standard Deviation	BPM	KEY	Accousticness	Danceability	Energy	Instrumentalness	Liveness	Loudness(in %)	Valence	Popularity	Loudness(in dB)	Genre / Mood
0	1	88.675421	88.675421	136	F Minor	0.02	0.46	0.55	0.02	0.18	0.89	0.39	0.48	-6.527	0
1	2	96.122737	16.607486	111	A Major	0.00	0.42	0.98	0.02	0.32	0.90	0.28	0.35	-5.774	0
2	3	96.207487	16.932029	100	A Major	0.14	0.36	0.69	0.00	0.12	0.86	0.23	0.09	-8.390	0
3	4	97.333898	18.109841	130	B Major	0.38	0.56	0.51	0.00	0.23	0.86	0.18	0.47	-8.665	0
4	5	98.707848	17.422191	169	D Major	0.23	0.63	0.46	0.00	0.06	0.83	0.36	0.42	-10.423	0

Step 4: Importing the python library **matplotlib.pyplot** and print the column names.
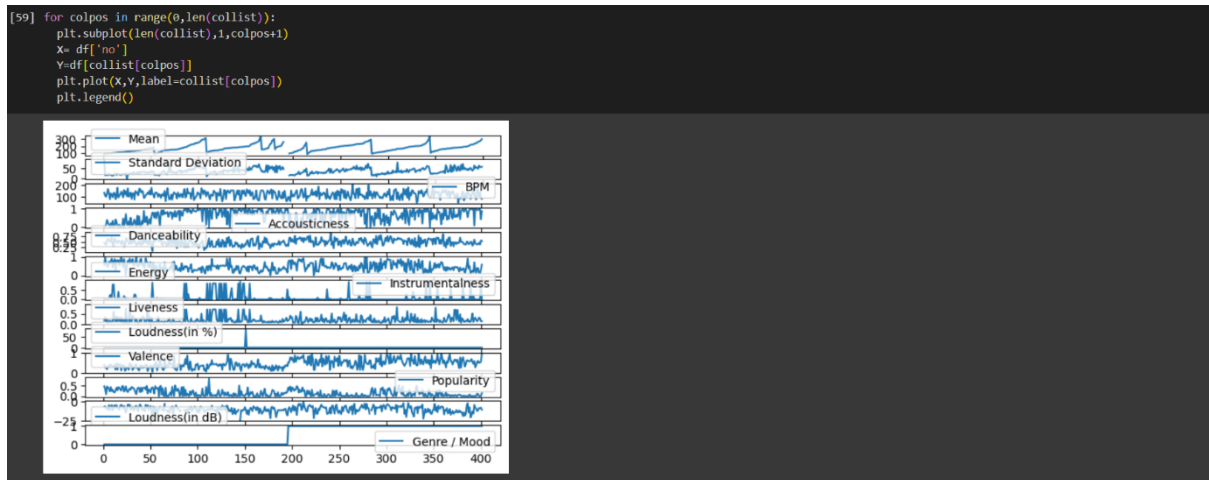
```
[38] import matplotlib.pyplot as plt
```

```
print(df.columns)

Index(['no', 'Mean', 'Standard Deviation', 'BPM', 'KEY', 'Accousticness',
       'Danceability', 'Energy', 'Instrumentalness', 'Liveness',
       'Loudness(in %)', 'Valence', 'Popularity', 'Loudness(in dB)',
       'Genre / Mood'],
      dtype='object')
```

Step 5: Taking a variable name **collist** where we store the columns names of the dataset

```
collist= [ 'Mean', 'Standard Deviation', 'BPM', 'Accousticness',
        'Danceability', 'Energy', 'Instrumentalness', 'Liveness',
        'Loudness(in %)', 'Valence', 'Popularity',
        'Loudness(in dB)', 'Genre / Mood']
```

Step 6: We are using the **subplot() function** to visualize the data points of each attribute or column in the data set. In subplot() function we pass 3 parameters first is the rows, second is the column and third is the index of the subplot.

```
[59] for colpos in range(0,len(collist)):
        plt.subplot(len(collist),1,colpos+1)
        X= df['no']
        Y=df[collist[colpos]]
        plt.plot(X,Y,label=collist[colpos])
        plt.legend()
```


Step 7: A histogram is a graphical representation that displays the distribution of a numerical attribute. In this context, a histogram was used to visualize the distribution of each attribute. The steps involved were:

- Extracted the attribute from the dataset.
- Divided the range of tempo values into bins.
- Plotted a histogram to visualize the frequency of songs within each bin.
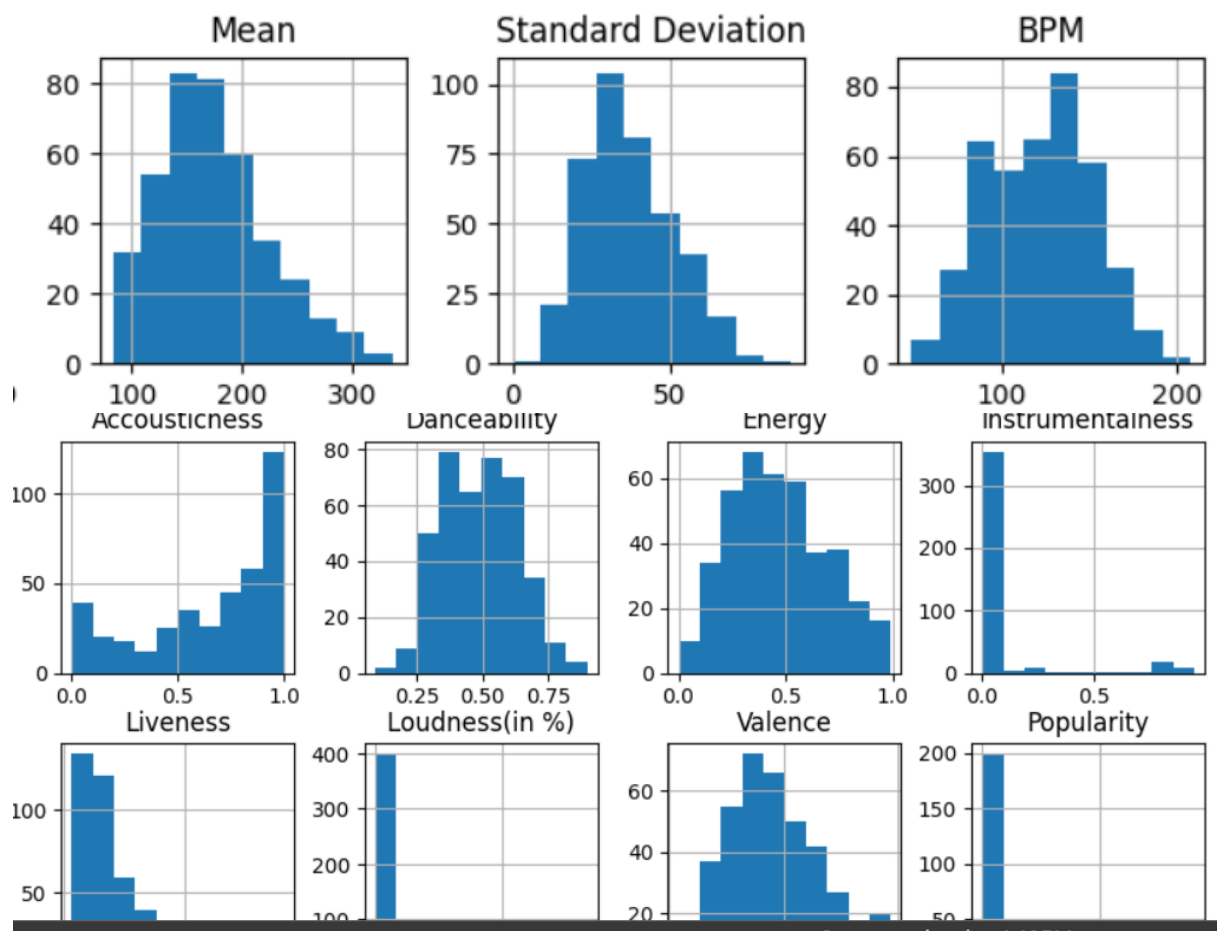
Step 8: A correlation heatmap is a graphical representation of the correlation matrix, which shows the pairwise correlations between numerical attributes. In the context of the Bengali Song Mood Prediction Model dataset. We used seaborn library for creating the heatmap.

Interpretation:

- Positive correlations (values close to +1) indicate that when one attribute increases, the other tends to increase as well.
- Negative correlations (values close to -1) indicate that when one attribute increases, the other tends to decrease.
- Values closer to 0 indicate weaker or no linear correlation

This visualization helps in identifying attributes that might have high or low correlation, which can guide feature selection and model building.
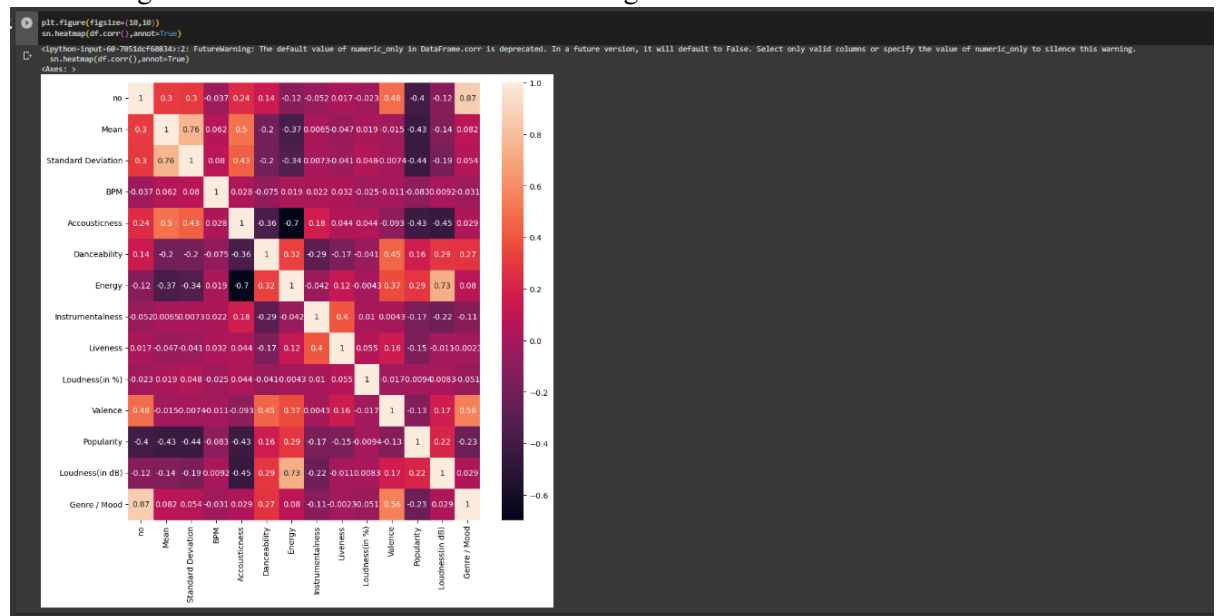

Step 9: A **bar plot** based on mood and valence (an emotional measure ranging from negative to positive) can provide insights into how different moods are associated with varying levels of valence. The steps taken were:

- Extracted the mood and valence attributes from the dataset.
- Grouped the data by mood categories.
- Calculated the average valence for each mood category.
- Created a bar plot to visualize using seaborn library

Interpretation:
- The bar plot shows how positive or negative the valence tends to be for each mood category.
- High positive valence could correspond to moods like "Happy" or "Energetic," while low valence might be associated with "Sad" or "Calm" moods.

This visualization helps in understanding the emotional context of different mood categories.


```
[55] sn.barplot(data=df,x=df["Genre / Mood"],y=df["Valence"])
```
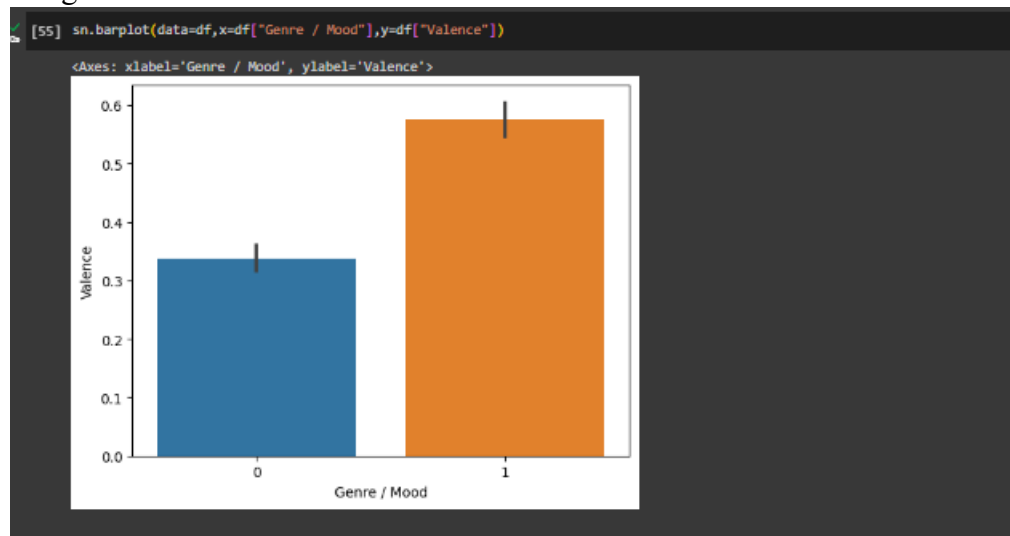
Step 9: A **countplot** with mood is a visualization that shows the distribution of songs across different mood categories. The process involves:

- Extracting the mood attribute from the dataset.
- Creating a countplot to visualize the frequency of each mood category.

Interpretation:
The countplot provides insights into whether the dataset is balanced or skewed towards specific moods.

It helps in understanding the availability of data for each mood category.
This visualization is crucial for assessing the class distribution and potential class imbalance issues.

Count Plot

```
sn.countplot(x=df["Genre / Mood"])
```