NAME: ISHART MAHAMADSHARIF MULANI
BRANCH:ENTC
DIVISION: ET2
ROLL NO: ET2-58
SUBJECT: EDS
TOPIC: problem statements for a OpinRank Review dataset using Numpy and Pandas

1.Total number of review

```
import pandas as pd
df = pd.read_csv('hotel.csv')
total_reviews = df.shape[0]
```

2.Find the number of missing values in each column.

```
missing_values = df.isnull().sum()
```

3.List unique authors.

```
unique_authors = df['author'].unique()
```

4.Find the review with the highest VADER rating.

```
highest_vader = df.loc[df['Vader_rati'].idxmax()]
```

5.Find the review with the lowest VADER rating

```
lowest_vader = df.loc[df['Vader_rati'].idxmin()]
```

6.First 5 record

```
df.head()
```

7.Last 5 record

```
df.tail()
```

8.first 10 record

```
df.head(10)
```

9.No. of rows and columns

```
df.shape( )
```

10.Last 10 record

```
df.tail(10)
```

11.Find the top 3 reviews with the longest text after cleaning.

```
df['text_length'] = df['cleaned_te'].str.len()
longest_reviews = df.sort_values(by='text_length', ascending=False).head(3)
```

12.Column name

```
df.columns
```

13.Datatype

```
df.dtypes
```

14.Find the median VADER rating.

```
import pandas as pd
import numpy as np
df = pd.read_csv('hotel.csv')
median_vader = np.median(df['Vader_rati'].to_numpy())
```

15.Find the number of reviews where VADER rating is above average.

```
vader_scores = df['Vader_rati'].to_numpy()
average_vader = np.mean(vader_scores)
above_avg_count = np.sum(vader_scores > average_vader)
```

16.Find the 5 smallest VADER ratings.

```
five_smallest_vader = np.sort(vader_scores)[:5]
```

17.Find the average VADER rating for reviews with final_sentiment = 1.

```
vader_positive = vader_scores[df['final_sentiment'].to_numpy() == 1]
average_vader_positive = np.mean(vader_positive)
```

18.Normalize the VADER ratings between 0 and 1.

```
vader_min = np.min(vader_scores)
vader_max = np.max(vader_scores)
vader_normalized = (vader_scores - vader_min) / (vader_max - vader_min)
```

19.Find reviews whose VADER rating is in the top 10%.

```
threshold = np.percentile(vader_scores, 90)
top_10_percent_reviews = df[vader_scores >= threshold]
```

20.Find the index of reviews where VADER score is negative.

```
negative_indices = np.where(vader_scores < 0)[0]
```