Theory Activity No. 1

Name :- Tejas Appa Gohad

Div :- CC Roll No. :- CC-63

PRN – 202401050068

Dataset Name :- Twitter US Airline Sentiment

1. Load the Data:

```
import pandas as pd
import numpy as np

# Load the dataset
df = pd.read_csv('Tweets.csv')
```

---

2. Problem Statements and Solutions:

---

1. How many tweets are there in total?

```
total_tweets = df.shape[0]
print(total_tweets)
```

---

2. How many unique airlines are present?

```
unique_airlines = df['airline'].nunique()
print(unique_airlines)
```

---

3. Count of tweets per airline.

```
tweets_per_airline = df['airline'].value_counts()
print(tweets_per_airline)
```

---

4. Find the percentage of each sentiment type (positive, neutral, negative).

```
sentiment_percentage = df['airline_sentiment'].value_counts(normalize=True) * 100
print(sentiment_percentage)
```

---

5. Find the airline with the highest number of negative tweets.

```
negative_tweets = df[df['airline_sentiment'] == 'negative']['airline'].value_counts()
most_negative_airline = negative_tweets.idxmax()
print(most_negative_airline)
```

---

6. Find the average confidence for sentiment classification.

```
avg_sentiment_confidence = df['airline_sentiment_confidence'].mean()
print(avg_sentiment_confidence)
```

---

7. Find how many tweets have a sentiment confidence above 0.8.

```
high_confidence_tweets = df[df['airline_sentiment_confidence'] > 0.8].shape[0]
print(high_confidence_tweets)
```

---

8. Identify the most common reason for negative tweets.

```
most_common_negative_reason = df['negativereason'].mode()[0]
print(most_common_negative_reason)
```

---

9. Find the airline with the least number of positive tweets.

```
positive_tweets = df[df['airline_sentiment'] == 'positive']['airline'].value_counts()
least_positive_airline = positive_tweets.idxmin()
print(least_positive_airline)
```

---

10. Find the earliest and latest tweet date.

```
df['tweet_created'] = pd.to_datetime(df['tweet_created'])
earliest_tweet = df['tweet_created'].min()
latest_tweet = df['tweet_created'].max()
print(earliest_tweet, latest_tweet)
```

---

11. Calculate the average length of a tweet.

```
df['tweet_length'] = df['text'].apply(len)
avg_tweet_length = df['tweet_length'].mean()
print(avg_tweet_length)
```

---

12. Find which timezone has the most tweets.

```
most_active_timezone =
df['user_timezone'].mode()[0]
print(most_active_timezone)
```

---

13. Find the proportion of negative tweets that mention "late flight" as the negative reason.

```
late_flight_tweets = df[df['negativereason'] == 'Late Flight']
proportion_late_flight = (late_flight_tweets.shape[0] / df[df['airline_sentiment'] == 'negative'].shape[0]) * 100
print(proportion_late_flight)
```

---

14. For each airline, find the average sentiment confidence.

```
avg_confidence_per_airline = df.groupby('airline')['airline_sentiment_confidence'].mean()
```

```
print(avg_confidence_per_airline)
```

---

15. Find how many tweets are missing the "negative reason" information.

```
missing_neg_reason = df['negativereason'].isnull().sum()
print(missing_neg_reason)
```

---

16. Find the top 3 airlines with the highest proportion of positive tweets.

```
positive_counts = df[df['airline_sentiment'] == 'positive']['airline'].value_counts()
```

```
total_counts = df['airline'].value_counts()
positive_ratio = (positive_counts / total_counts).sort_values(ascending=False)
top_3_positive_airlines = positive_ratio.head(3)
print(top_3_positive_airlines)
```

---

17. Identify how many unique reasons are cited for negative sentiment.

```
unique_negative_reasons = df['negativereason'].nunique()
print(unique_negative_reasons)
```

---

18. Create a pivot table showing count of sentiment per airline.

```
sentiment_pivot = pd.pivot_table(df, index='airline', columns='airline_sentiment', values='tweet_id', aggfunc='count', fill_value=0)
print(sentiment_pivot)
```

---

19. Find the tweet with the maximum sentiment confidence.

```
max_confidence_tweet = df.loc[df['airline_sentiment_confidence'].idxmax()]
print(max_confidence_tweet[['text', 'airline_sentiment', 'airline_sentiment_confidence']])
```

---

20. Calculate correlation between tweet length and sentiment confidence.

```
correlation = df['tweet_length'].corr(df['airline_sentiment_confidence'])
print(correlation)
```

Summary:

These 20 problems cover descriptive statistics, groupings, missing values, correlations, and pivot tables.

They use Numpy and Pandas in practical ways.

Would you also like me to show visualizations (like graphs) for some of these problems? It will make the insights even clearer!

(Example: Bar plots for sentiment counts, Pie charts for airline performance, Heatmaps for correlations...)

Would you like that?