Topic:- Enron Email Dataset

Name:- Oshin Kalbande
Roll no:- CS4-27
Prn no:- 202401040189

1. Total Number of Emails

Problem: How many emails are in the dataset?
Solution (Pandas):

total_emails = df.shape[0]

```
Python

total_emails = df.shape[0]
```

---

2. Unique Email Addresses

Problem: How many unique email addresses are involved (as senders or recipients)?
Solution:

unique_emails = pd.unique(df['From'].append(df['To'])).shape[0]

```
unique_emails =
pd.unique(df['From'].append(df['To'])).shape[0]
```

---

3. Top 10 Most Active Senders

Problem: Who are the top 10 senders by number of emails sent?
Solution:

top_senders = df['From'].value_counts().head(10)

```
top_senders = df['From'].value_counts().head(10)
```

---

4. Top 10 Most Contacted Recipients

Problem: Who are the most frequently contacted recipients?

Solution:

top_recipients = df['To'].value_counts().head(10)

```
top_recipients = df['To'].value_counts().head(10)
```

---

5. Average Email Length

Problem: What is the average number of characters in email bodies?
Solution:

df['BodyLength'] = df['Body'].str.len()
avg_length = df['BodyLength'].mean()

```
df['BodyLength'] = df['Body'].str.len()
avg_length = df['BodyLength'].mean()
```

---

6. Distribution of Email Lengths

Problem: How are email body lengths distributed?
Solution (NumPy):

length_distribution = np.histogram(df['BodyLength'], bins=10)

```
length_distribution =
np.histogram(df['BodyLength'], bins=10)
```

---

7. Emails Sent Per Month

Problem: How many emails were sent each month?
Solution:

df['Date'] = pd.to_datetime(df['Date'])
monthly_emails = df.groupby(df['Date'].dt.to_period('M')).size()

```
df['Date'] = pd.to_datetime(df['Date'])
monthly_emails =
df.groupby(df['Date'].dt.to_period('M')).size()
```

---

8. Most Common Email Subjects

Problem: What are the most frequent email subject lines?
Solution:

common_subjects = df['Subject'].value_counts().head(10)

```
common_subjects =
df['Subject'].value_counts().head(10)
```

---

9. Count of Emails with Attachments

Problem: How many emails include attachments?
Solution:

attachment_count = df['Subject'].str.contains('attachment', case=False, na=False).sum()

```
attachment_count =
df['Subject'].str.contains('attachment', case=False,
na=False).sum()
```

---

10. Emails Sent Outside Business Hours

Problem: How many emails were sent outside of 9 AM to 5 PM?
Solution:

```
df['Hour'] = df['Date'].dt.hour
outside_business_hours = df[(df['Hour'] < 9) |
(df['Hour'] > 17)].shape[0]
```

---

11. Longest Email

Problem: Which email has the longest body?
Solution:

longest_email = df.loc[df['BodyLength'].idxmax()]

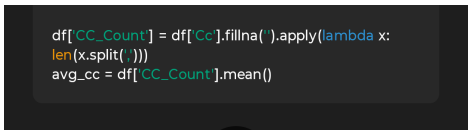
```
longest_email = df.loc[df['BodyLength'].idxmax()]
```

---

12. Number of CCs per Email

Problem: What is the average number of CCs per email?
Solution:

df['CC_Count'] = df['Cc'].fillna('').apply(lambda x: len(x.split(',')))
avg_cc = df['CC_Count'].mean()



---

13. Emails Containing Specific Keywords (e.g., "fraud")

Problem: How many emails contain the keyword “fraud”?
Solution:

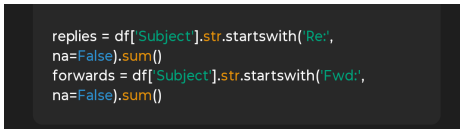
fraud_emails = df['Body'].str.contains('fraud', case=False, na=False).sum()

---

14. Number of Replies vs Forwards

Problem: How many emails are replies and how many are forwards?
Solution:

replies = df['Subject'].str.startswith('Re:', na=False).sum()
forwards = df['Subject'].str.startswith('Fwd:', na=False).sum()



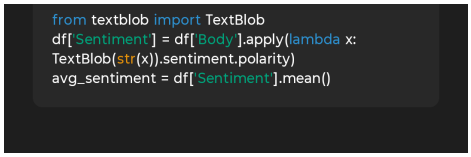
---

15. Sentiment Analysis on Emails

Problem: What is the average sentiment polarity of email bodies?
(Requires external library like TextBlob)

from textblob import TextBlob

df['Sentiment'] = df['Body'].apply(lambda x: TextBlob(str(x)).sentiment.polarity)
avg_sentiment = df['Sentiment'].mean()

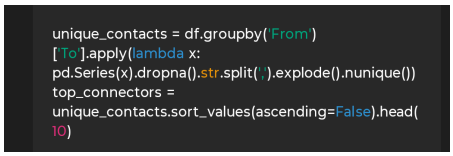


---

16. Most Connected Employees

Problem: Which employee communicates with the most unique recipients?
Solution:

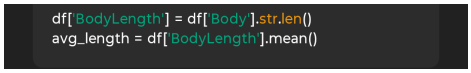
unique_contacts = df.groupby('From')['To'].apply(lambda x:
pd.Series(x).dropna().str.split(',').explode().nunique())
top_connectors = unique_contacts.sort_values(ascending=False).head(10)



---

17. Average Response Time

Problem: What is the average time it takes to get a reply to an email?
(Requires matching replies with original emails, advanced parsing)

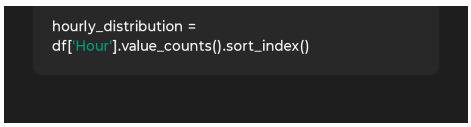


---

18. Hourly Email Distribution

Problem: How are emails distributed throughout the day by hour?
Solution:

hourly_distribution = df['Hour'].value_counts().sort_index()



---

19. Percentage of Emails with Empty Body

Problem: What percentage of emails have no body content?
Solution:

empty_body_pct = df['Body'].isna().mean() * 100

```
empty_body_pct = df['Body'].isna().mean() * 100
```

---

20. Emails Per Employee Per Day

Problem: What's the average number of emails sent per employee per day?
Solution:

emails_per_day = df.groupby(['From', df['Date'].dt.date]).size().groupby('From').mean()

```
emails_per_day = df.groupby(['From',
df['Date'].dt.date]).size().groupby('From').mean()
```