Theory Activity – 1

Name: - Om Ravindra Deore

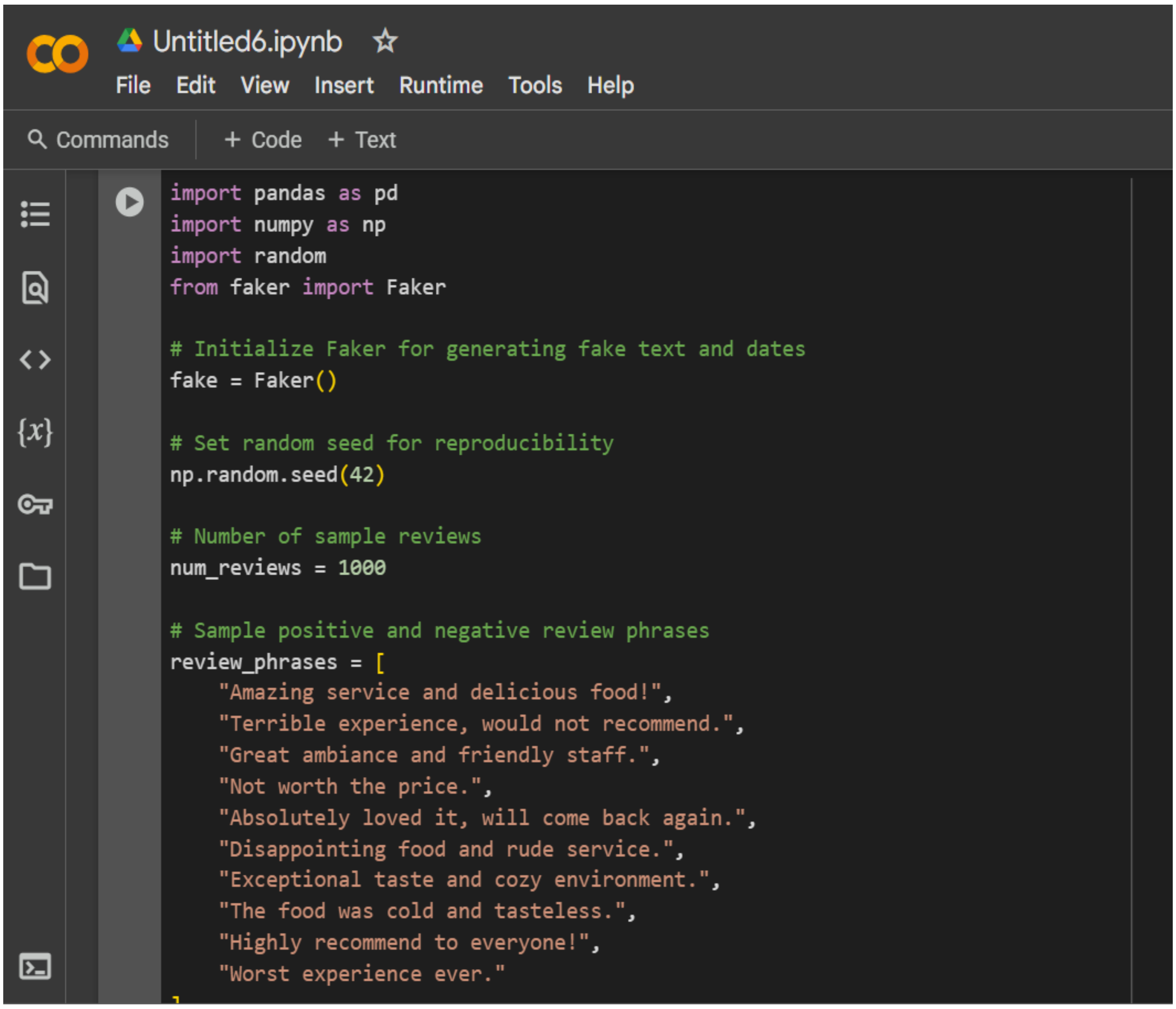
Div: - CS5

Roll: - CS5-40

PRN: - 202401100093

Dataset: - Yelp Reviews

Creation of dataset on YELP Review:



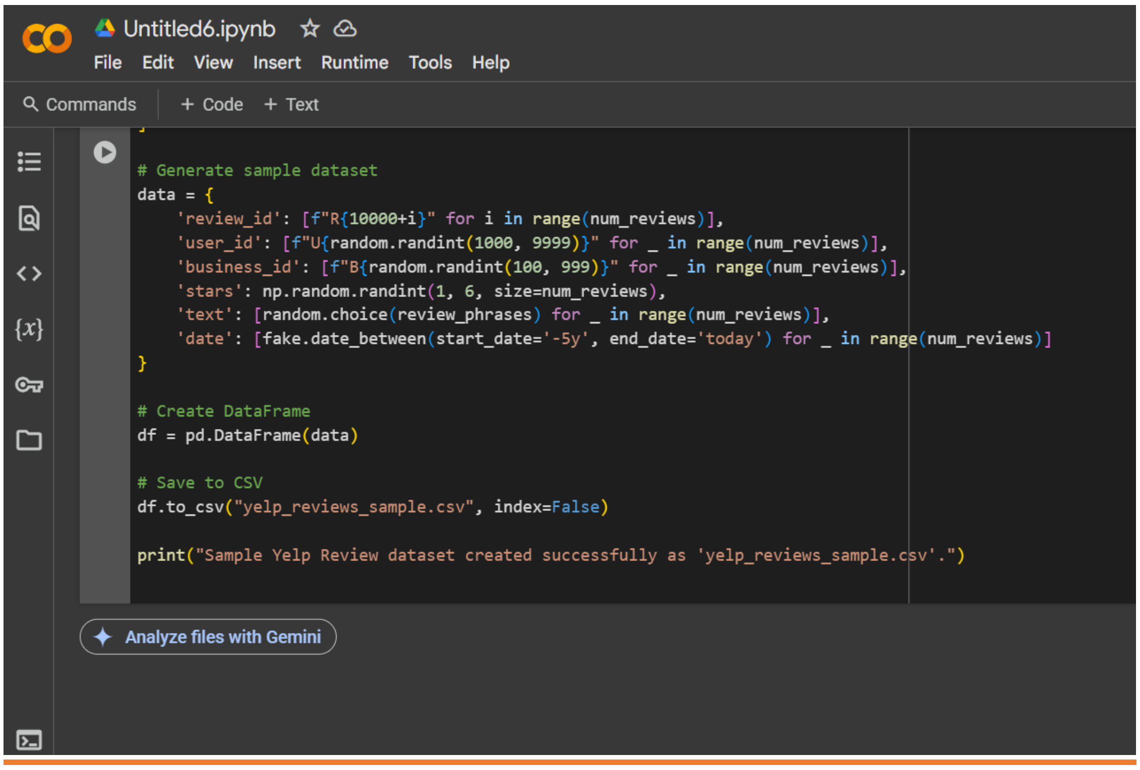
```python
import pandas as pd
import numpy as np
import random
from faker import Faker

# Initialize Faker for generating fake text and dates
fake = Faker()

# Set random seed for reproducibility
np.random.seed(42)

# Number of sample reviews
num_reviews = 1000

# Sample positive and negative review phrases
review_phrases = [
    "Amazing service and delicious food!",
    "Terrible experience, would not recommend.",
    "Great ambiance and friendly staff.",
    "Not worth the price.",
    "Absolutely loved it, will come back again.",
    "Disappointing food and rude service.",
    "Exceptional taste and cozy environment.",
    "The food was cold and tasteless.",
    "Highly recommend to everyone!",
    "Worst experience ever."
```


Untitled6.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Analyze files with Gemini

```python
# Generate sample dataset
data = {
    'review_id': [f"R{10000+i}" for i in range(num_reviews)],
    'user_id': [f"U{random.randint(1000, 9999)}" for _ in range(num_reviews)],
    'business_id': [f"B{random.randint(100, 999)}" for _ in range(num_reviews)],
    'stars': np.random.randint(1, 6, size=num_reviews),
    'text': [random.choice(review_phrases) for _ in range(num_reviews)],
    'date': [fake.date_between(start_date='-5y', end_date='today') for _ in range(num_reviews)]
}

# Create DataFrame
df = pd.DataFrame(data)

# Save to CSV
df.to_csv("yelp_reviews_sample.csv", index=False)

print("Sample Yelp Review dataset created successfully as 'yelp_reviews_sample.csv'.")
```

Questions On Numpy with Solution:

- **Q1.** Find the average review length.
  Solution: np.mean(review_lengths)

- **Q2.** Find the maximum review length.
  Solution: np.max(review_lengths)

- **Q3.** Find the minimum review length.
  Solution: np.min(review_lengths)

- **Q4.** Find unique star ratings.
  Solution: np.unique(df['stars'])

- **Q5.** Find standard deviation of review lengths.
  Solution: np.std(review_lengths)

- **Q6.** Find correlation between stars and review lengths.
  Solution: `np.corrcoef(df['stars'], review_lengths)`

- **Q7.** Find number of reviews longer than 100 characters.
  Solution: `np.sum(review_lengths > 100)`

- **Q8.** Calculate median review length.
  Solution: `np.median(review_lengths)`

- **Q9.** Calculate variance of star ratings.
  Solution: `np.var(df['stars'])`

- **Q10.** Find number of reviews with 5 stars.
  Solution: `np.count_nonzero(df['stars'] == 5)`

Questions On Pandas with Solution:

- Q1. Count reviews for each star rating.
  Solution: df['stars'].value_counts()

- Q2. Find top 5 users with most reviews.
  Solution:
  df['user_id'].value_counts().head(5)

- Q3. Find number of reviews per year.
  Solution:
  df['date'].dt.year.value_counts()

- Q4. Find average stars per business.
  Solution:
  df.groupby('business_id')['stars'].mea
  n()

- Q5. Find businesses with only 1 review.
  Solution:

```
df['business_id'].value_counts().eq(1).sum()
```

- **Q6.** Find reviews posted on weekends.
  Solution:
  ```
  df[df['date'].dt.weekday >= 5]
  ```

- **Q7.** Total words written by each user.
  Solution:
  ```
  df.groupby('user_id')['text'].apply(lambda x: x.str.split().str.len().sum())
  ```

- **Q8.** Review with the longest text.
  Solution:
  ```
  df['text'].apply(len).idxmax()
  ```

- **Q9.** Extract reviews containing 'amazing'.
  Solution:
  ```
  df[df['text'].str.contains('amazing', case=False)]
  ```

- **Q10.** Average review length per star rating.
  Solution:
  ```
  df.groupby('stars')['text'].
  apply(lambda x:
  x.str.len().mean()
  ```

THANK YOU