NAME: ANUSHKA NAMDEV ROMAN

CLASS: CS6

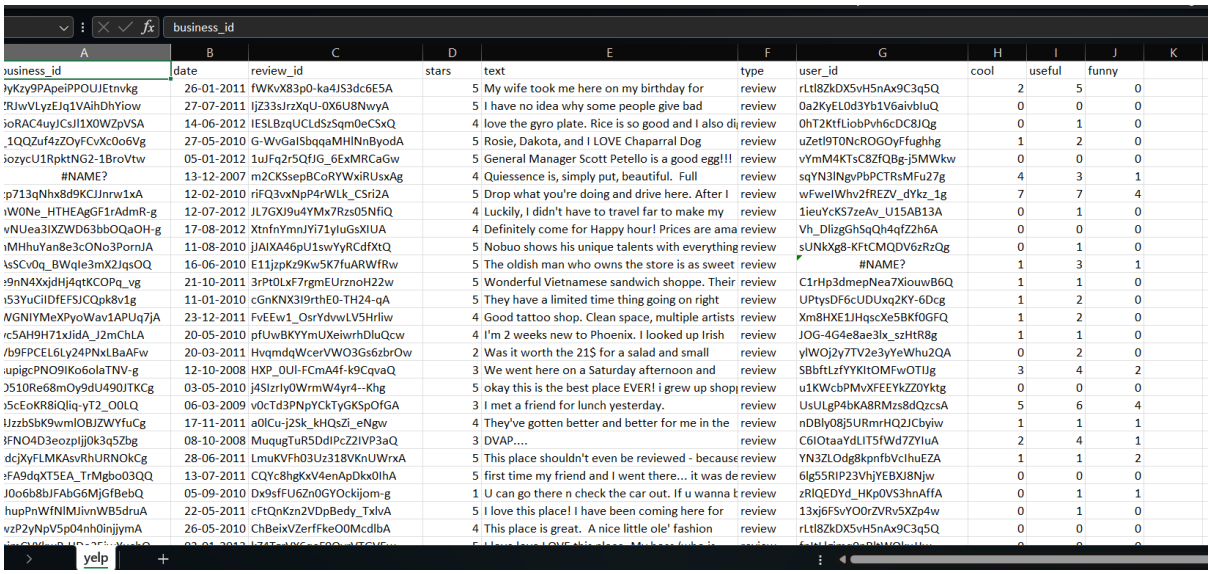
ROLL NO: CS6-39

PRN: 202401100035

GOOGLE COLAB LINK:

https://colab.research.google.com/drive/1OpeEBu1UGW_YNlJJ6ZMdDDbYHuhvmmVB

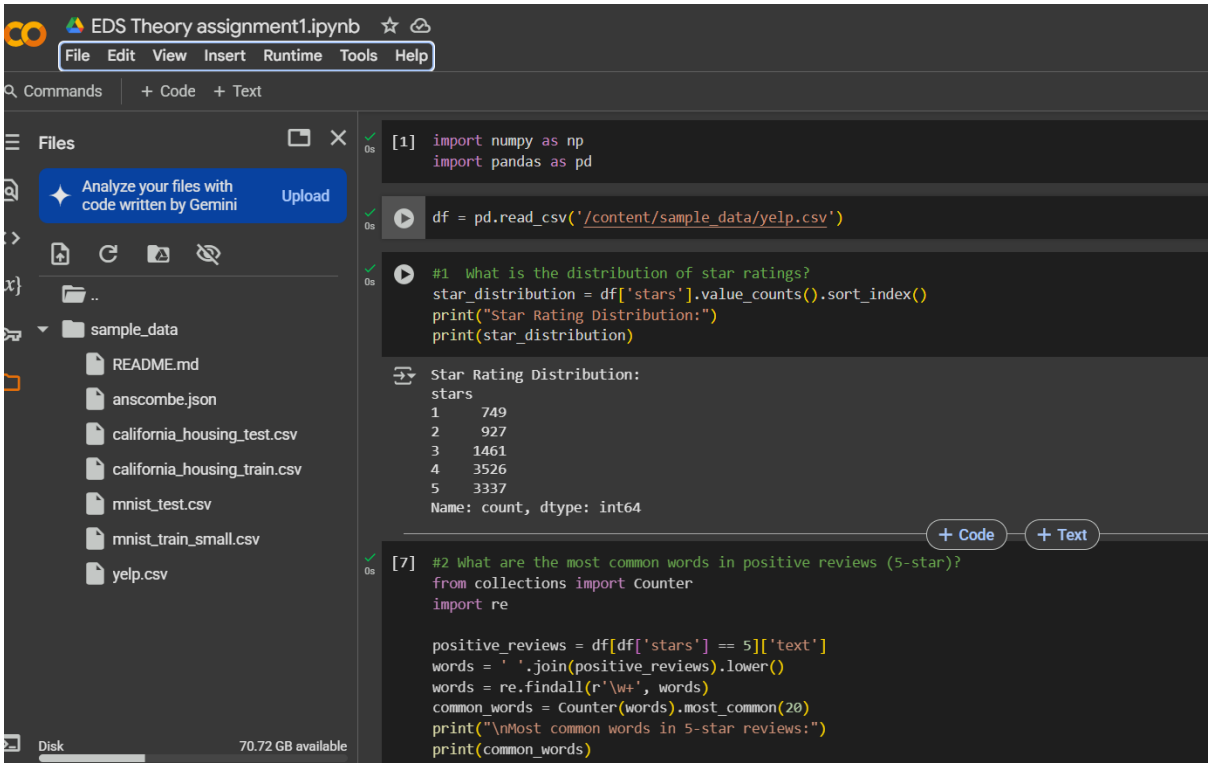
DATASET:

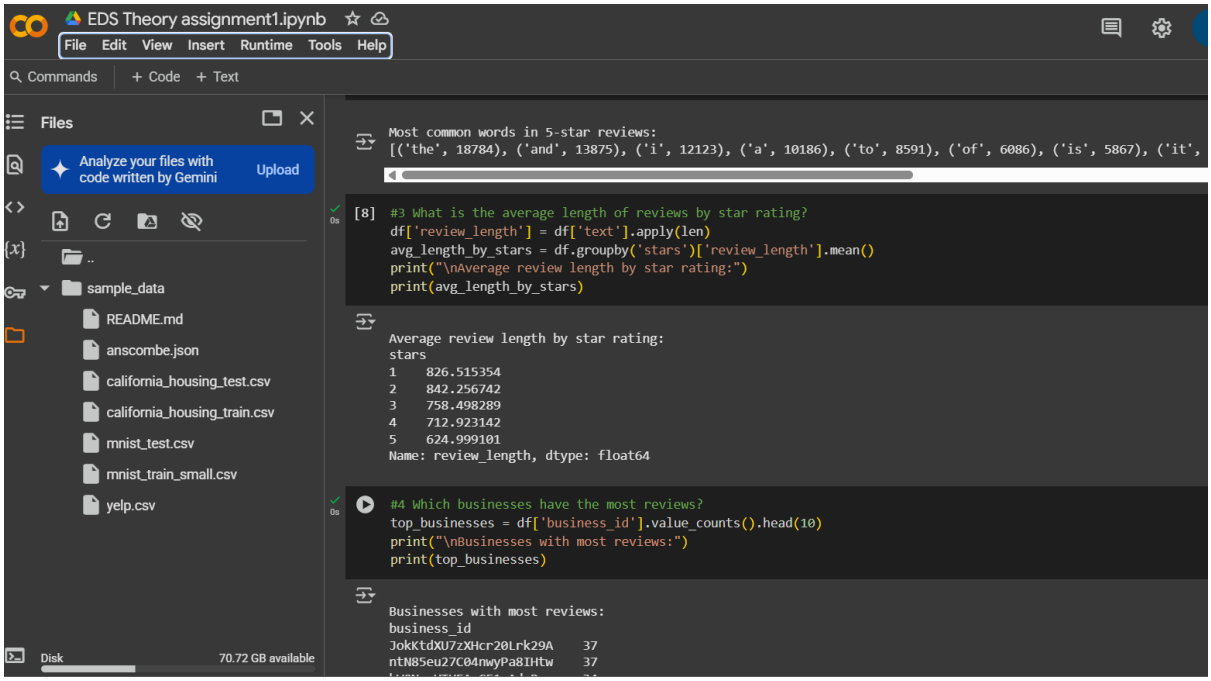


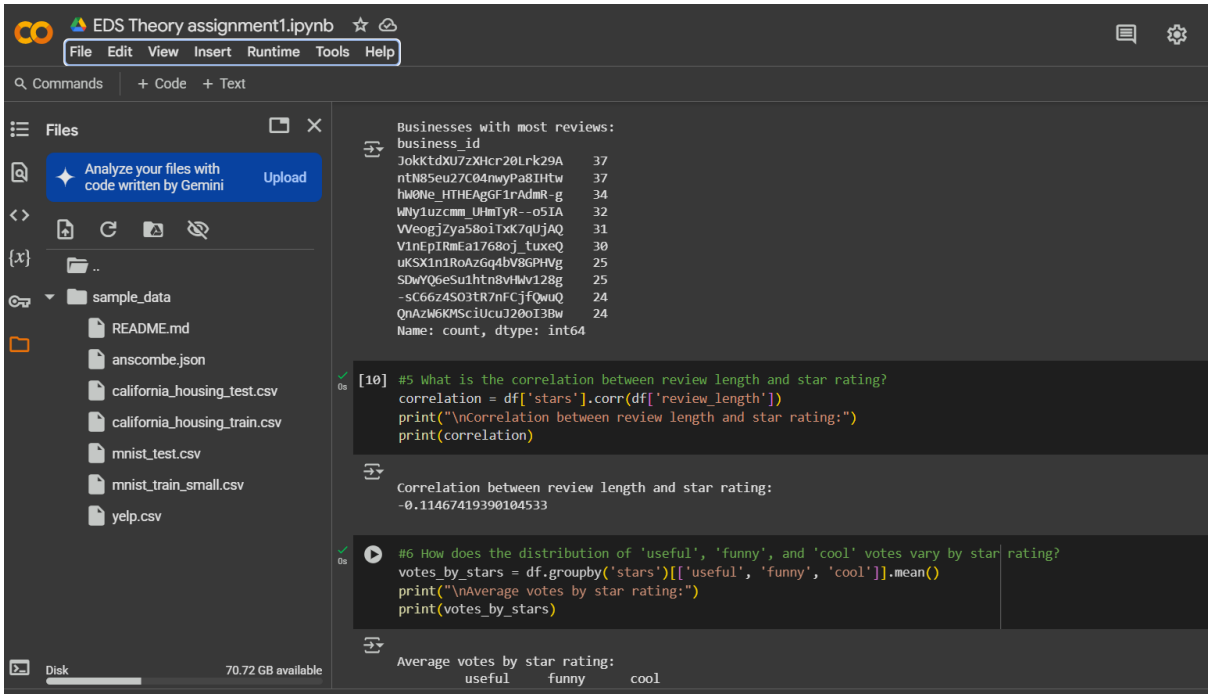
business_id

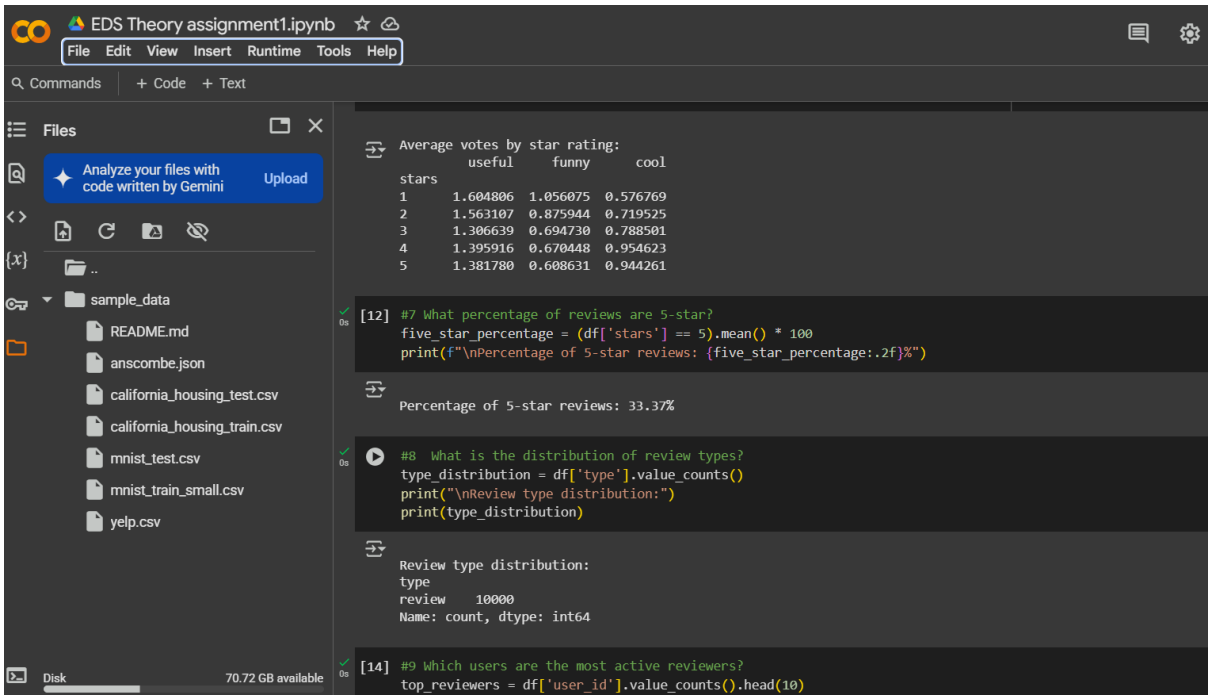
A	B	C	D	E	F	G	H	I	J	K
usiness_id	date	review_id	stars	text	type	user_id	cool	useful	funny	
yKzy9PApeiPPOUJEtnvkg	26-01-2011	fWKvX83p0-ka4JS3dc6E5A	5	My wife took me here on my birthday for	review	rLtl8ZkDX5vH5nAx9C3q5Q	2	5	0	
RJwVLyzEJq1VAihDhYiow	27-07-2011	IjZ33sJrzXqU-0X6U8NwyA	5	I have no idea why some people give bad	review	0a2KyEL0d3Yb1V6aivbIuQ	0	0	0	
oRAC4uyJCsJl1X0WZpVSA	14-06-2012	IESLBzqUCLdSzSqm0eCSxQ	4	love the gyro plate. Rice is so good and I also di	review	0hT2KtfLiobPvh6cDC8JQg	0	1	0	
1QQZuf4zZOyFCvXc0o6Vg	27-05-2010	G-WvGaISbqqaMHlNnByodA	5	Rosie, Dakota, and I LOVE Chaparral Dog	review	uZetl9T0NcROGOyFfughhg	1	2	0	
ozycU1RpktNG2-1BroVtw	05-01-2012	1uJFq2r5QfJG_6ExMRCaGw	5	General Manager Scott Petello is a good egg!!!	review	vYmM4KTsC8ZfQBg-j5MWkw	0	0	0	
#NAME?	13-12-2007	m2CKSsepBCoRYWxiRUsxAg	4	Quiessence is, simply put, beautiful. Full	review	sqYN3lNgvPbPCTRsMFu27g	4	3	1	
p713qNhx8d9KCJJnrw1xA	12-02-2010	riFQ3vxNpP4rWLk_CSri2A	5	Drop what you're doing and drive here. After I	review	wFweIWhv2fREZV_dYkz_1g	7	7	4	
W0Ne_HTHEAgGF1rAdmR-g	12-07-2012	JL7GXJ9u4YMx7Rzs05NfiQ	4	Luckily, I didn't have to travel far to make my	review	1ieuYcKS7zeAv_U15AB13A	0	1	0	
NUea3IXZWD63bbOQaOH-g	17-08-2012	XtnfnYmnJYi71yIuGsXIUA	4	Definitely come for Happy hour! Prices are ama	review	Vh_DlizgGhSqQh4qfZ2h6A	0	0	0	
MHhuYan8e3cONo3PornJA	11-08-2010	jJAIXA46pU1swYyRCdfXtQ	5	Nobuo shows his unique talents with everything	review	sUNkXg8-KFtCMQDV6zRzQg	0	1	0	
sSCv0q_BWqIe3mX2JqsOQ	16-06-2010	E11jzpKz9Kw5K7fuARWfRw	5	The oldish man who owns the store is as sweet	review	#NAME?	1	3	1	
9nN4XxjdHj4qtKCOPq_vg	21-10-2011	3rPt0LxF7rgmEUrznoH22w	5	Wonderful Vietnamese sandwich shoppe. Their	review	C1rHp3dmepNea7XiouwB6Q	1	1	0	
53YuCiIDfEFSJCQpk8v1g	11-01-2010	cGnKNX3I9rthE0-TH24-qA	5	They have a limited time thing going on right	review	UPtysDF6cUDUxq2KY-6Dcg	1	2	0	
GNIYMeXPyoWav1APUq7jA	23-12-2011	FvEEw1_OsrYdvwLV5Hrliw	4	Good tattoo shop. Clean space, multiple artists	review	Xm8HXE1JHqscXe5BKf0GFQ	1	2	0	
c5AH9H71xJidA_J2mChLA	20-05-2010	pfUwBKYYmUXeiwrhDluQcw	4	I'm 2 weeks new to Phoenix. I looked up Irish	review	JOG-4G4e8ae3lx_szHtR8g	1	1	0	
b9FPCEL6Ly24PNxLBaAFw	20-03-2011	HvqmdqWcerVWO3Gs6zbrOw	2	Was it worth the 21$ for a salad and small	review	yIWOj2y7TV2e3yYeWhu2QA	0	2	0	
upigcPNO9IKo6olaTNV-g	12-10-2008	HXP_0Ul-FCmA4f-k9CqvaQ	3	We went here on a Saturday afternoon and	review	SBbftLzfYYKItOMFwOTIJg	3	4	2	
510Re68mOy9dU490JTKCg	03-05-2010	j4SIzrIy0WrmW4yr4--Khg	5	okay this is the best place EVER! i grew up shop	review	u1KWcbPMvXFEEYkZZ0Yktg	0	0	0	
5cEoKR8iQliq-yT2_O0LQ	06-03-2009	v0cTd3PNpYCkTyGKSpOfGA	3	I met a friend for lunch yesterday.	review	UsULgP4bKA8RMzs8dQzcsA	5	6	4	
JzzbSbK9wmlOBJZWYfuCg	17-11-2011	a0lCu-j2Sk_kHQsZi_eNgw	4	They've gotten better and better for me in the	review	nDBly08j5URmrHQ2JCbyiw	1	1	1	
FNO4D3eozpIjj0k3q5Zbg	08-10-2008	MuqugTuR5DdIPcZ2IVP3aQ	3	DVAP....	review	C6IOtaaYdLIT5fWd7ZYIuA	2	4	1	
dcjXyFLMKAsvRhURNOkCg	28-06-2011	LmuKVFh03Uz318VKnUWrxA	5	This place shouldn't even be reviewed - because	review	YN3ZLOdg8kpnfbVcIhuEZA	1	1	2	
FA9dqXT5EA_TrMgbo03QQ	13-07-2011	CQYc8hgKxV4enApDkx0IhA	5	first time my friend and I went there... it was de	review	6lg55RIP23VhjYEBXJ8Njw	0	0	0	
J0o6b8bJFAbG6MjGfBebQ	05-09-2010	Dx9sfFU6Zn0GYOckijom-g	1	U can go there n check the car out. If u wanna b	review	zRlQEDYd_HKp0VS3hnAffA	0	1	1	
hupPnWfNlMJivnWB5druA	22-05-2011	cFtQnKzn2VDpBedy_TxlvA	5	I love this place! I have been coming here for	review	13xj6FSvYO0rZVRv5XZp4w	0	1	0	
vzP2yNpV5p04nh0injjymA	26-05-2010	ChBeixVZerfFkeO0McdlbA	4	This place is great. A nice little ole' fashion	review	rLtl8ZkDX5vH5nAx9C3q5Q	0	0	0	

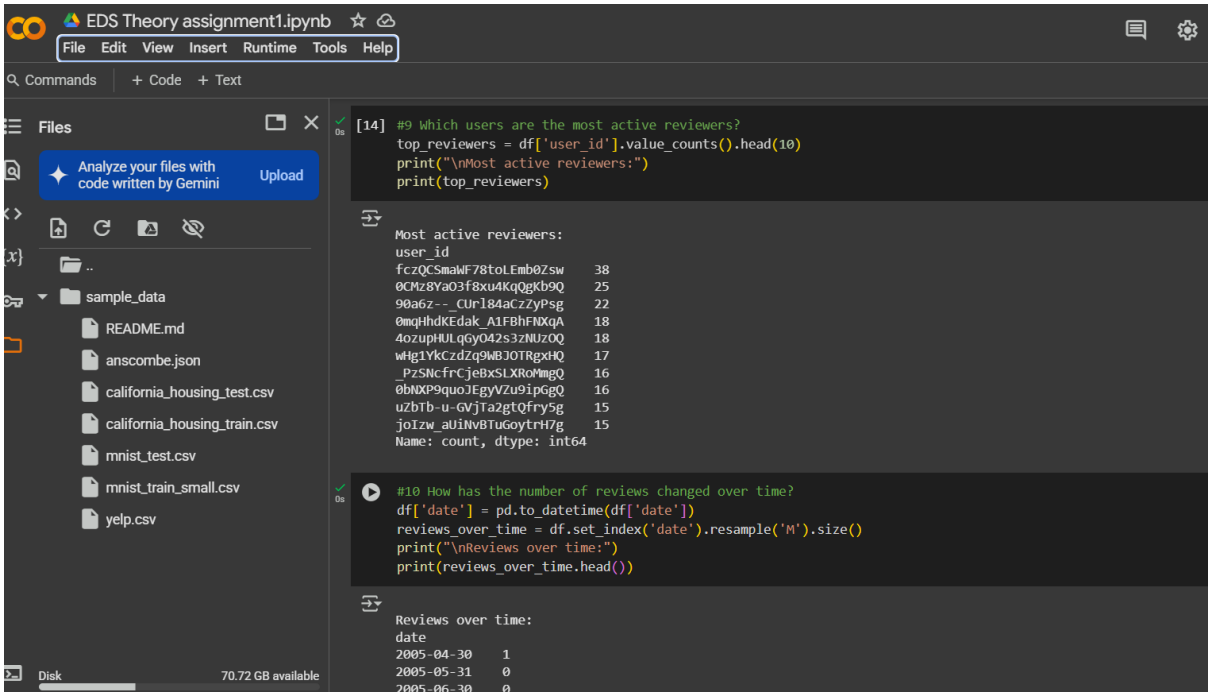
yelp

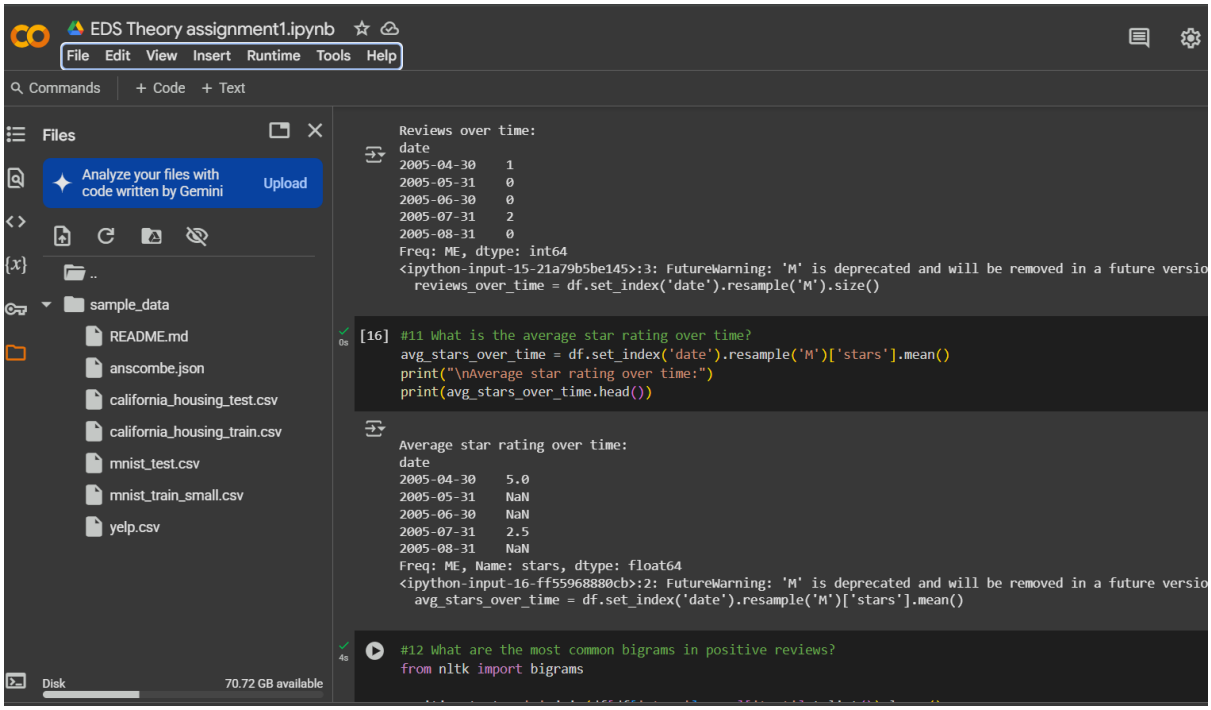

EDS Theory assignment1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Files
Analyze your files with code written by Gemini
Upload
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
yelp.csv
Disk 70.72 GB available
[1] import numpy as np
import pandas as pd
df = pd.read_csv('/content/sample_data/yelp.csv')
#1 What is the distribution of star ratings?
star_distribution = df['stars'].value_counts().sort_index()
print("Star Rating Distribution:")
print(star_distribution)
Star Rating Distribution:
stars
1 749
2 927
3 1461
4 3526
5 3337
Name: count, dtype: int64
+ Code + Text
[7] #2 What are the most common words in positive reviews (5-star)?
from collections import Counter
import re
positive_reviews = df[df['stars'] == 5]['text']
words = ' '.join(positive_reviews).lower()
words = re.findall(r'\w+', words)
common_words = Counter(words).most_common(20)
print("\nMost common words in 5-star reviews:")
print(common_words)

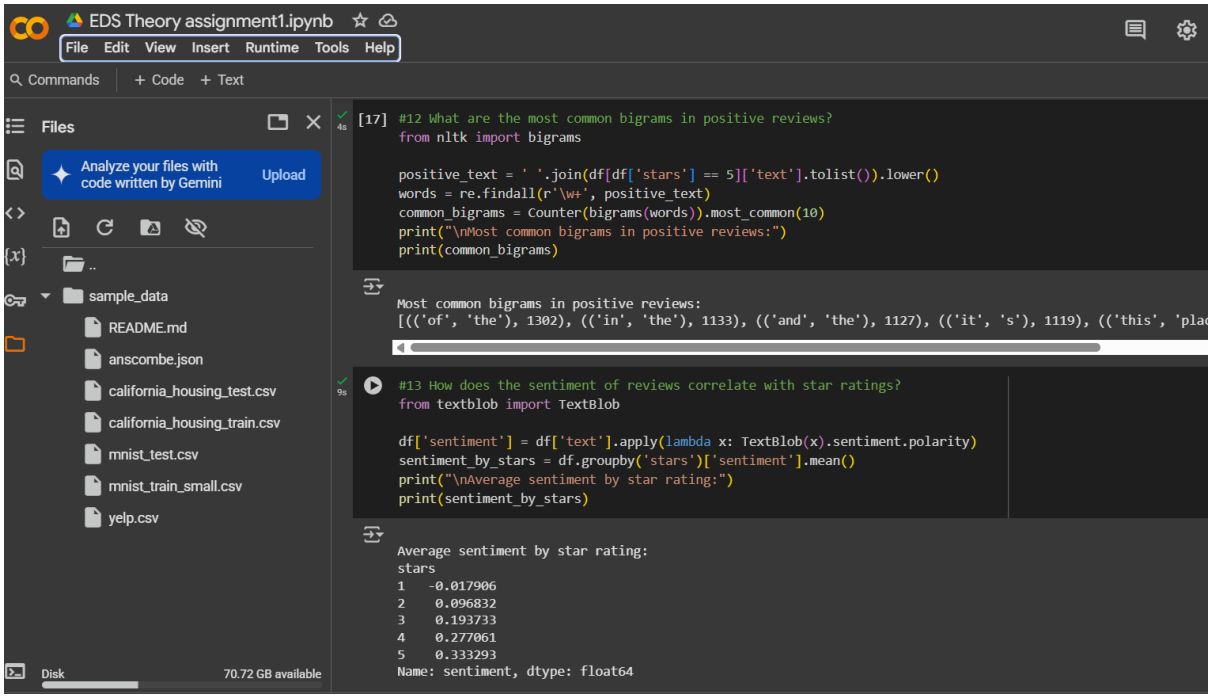

EDS Theory assignment1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Files
Analyze your files with code written by Gemini
Upload
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
yelp.csv
Disk 70.72 GB available
Most common words in 5-star reviews:
[('the', 18784), ('and', 13875), ('i', 12123), ('a', 10186), ('to', 8591), ('of', 6086), ('is', 5867), ('it',
[8] #3 What is the average length of reviews by star rating?
df['review_length'] = df['text'].apply(len)
avg_length_by_stars = df.groupby('stars')['review_length'].mean()
print("\nAverage review length by star rating:")
print(avg_length_by_stars)
Average review length by star rating:
stars
1 826.515354
2 842.256742
3 758.498289
4 712.923142
5 624.999101
Name: review_length, dtype: float64
#4 Which businesses have the most reviews?
top_businesses = df['business_id'].value_counts().head(10)
print("\nBusinesses with most reviews:")
print(top_businesses)
Businesses with most reviews:
business_id
JokKtdXU7zXHcr20Lrk29A 37
ntN85eu27C04nwyPa8IHtw 37

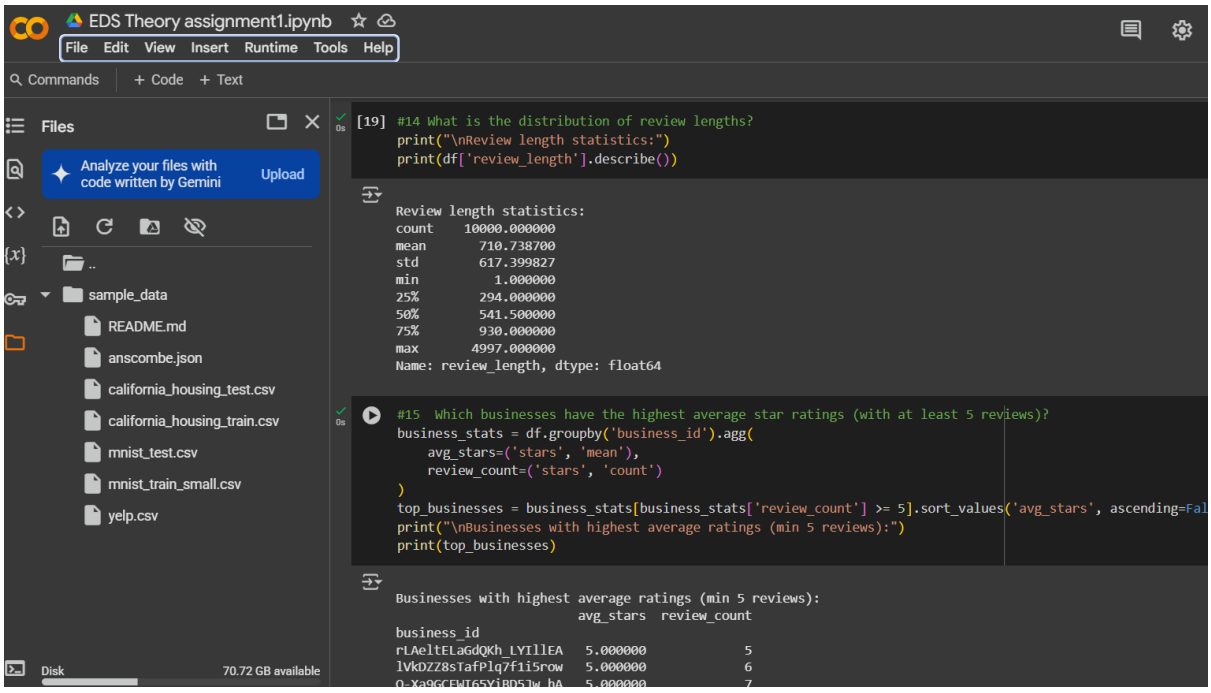

EDS Theory assignment1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Files
Analyze your files with code written by Gemini
Upload
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
yelp.csv
Disk 70.72 GB available
Businesses with most reviews:
business_id
JokKtdXU7zXHcr20Lrk29A 37
ntN85eu27C04nwyPa8IHtw 37
hW0Ne_HTHEAgGF1rAdmR-g 34
WNy1uzcmm_UHmTyR--o5IA 32
VVeogjZya58oiTxK7qUjAQ 31
V1nEpIRmEa1768oj_tuxeQ 30
uKSX1n1RoAzGq4bV8GPHVg 25
SDwYQ6eSu1htn8vHWw128g 25
-sC66z4SO3tR7nFCjfQwuQ 24
QnAzW6KMSciUcuJ2OoI3Bw 24
Name: count, dtype: int64
[10] #5 What is the correlation between review length and star rating?
correlation = df['stars'].corr(df['review_length'])
print("\nCorrelation between review length and star rating:")
print(correlation)
Correlation between review length and star rating:
-0.11467419390104533
#6 How does the distribution of 'useful', 'funny', and 'cool' votes vary by star rating?
votes_by_stars = df.groupby('stars')[['useful', 'funny', 'cool']].mean()
print("\nAverage votes by star rating:")
print(votes_by_stars)
Average votes by star rating:
useful funny cool

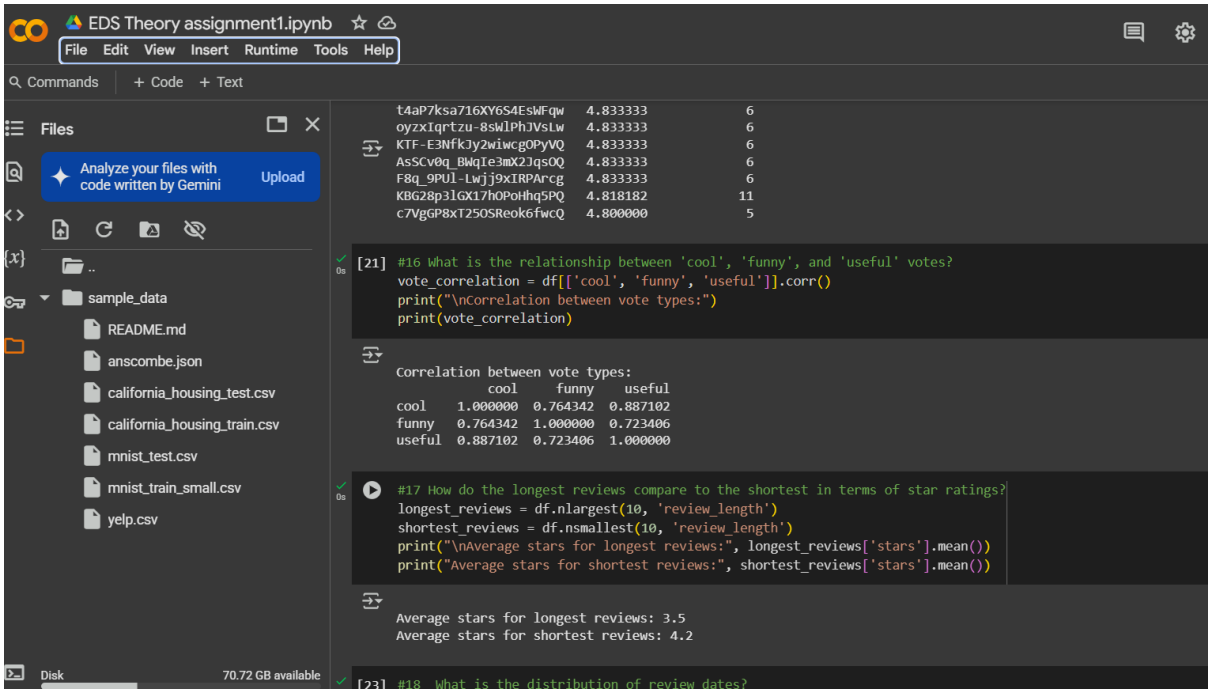

EDS Theory assignment1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Files
Analyze your files with code written by Gemini
Upload
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
yelp.csv
Disk 70.72 GB available
Average votes by star rating:
useful funny cool
stars
1 1.604806 1.056075 0.576769
2 1.563107 0.875944 0.719525
3 1.306639 0.694730 0.788501
4 1.395916 0.670448 0.954623
5 1.381780 0.608631 0.944261
[12] #7 What percentage of reviews are 5-star?
five_star_percentage = (df['stars'] == 5).mean() * 100
print(f"\nPercentage of 5-star reviews: {five_star_percentage:.2f}%")
Percentage of 5-star reviews: 33.37%
#8 What is the distribution of review types?
type_distribution = df['type'].value_counts()
print("\nReview type distribution:")
print(type_distribution)
Review type distribution:
type
review 10000
Name: count, dtype: int64
[14] #9 Which users are the most active reviewers?
top_reviewers = df['user_id'].value_counts().head(10)

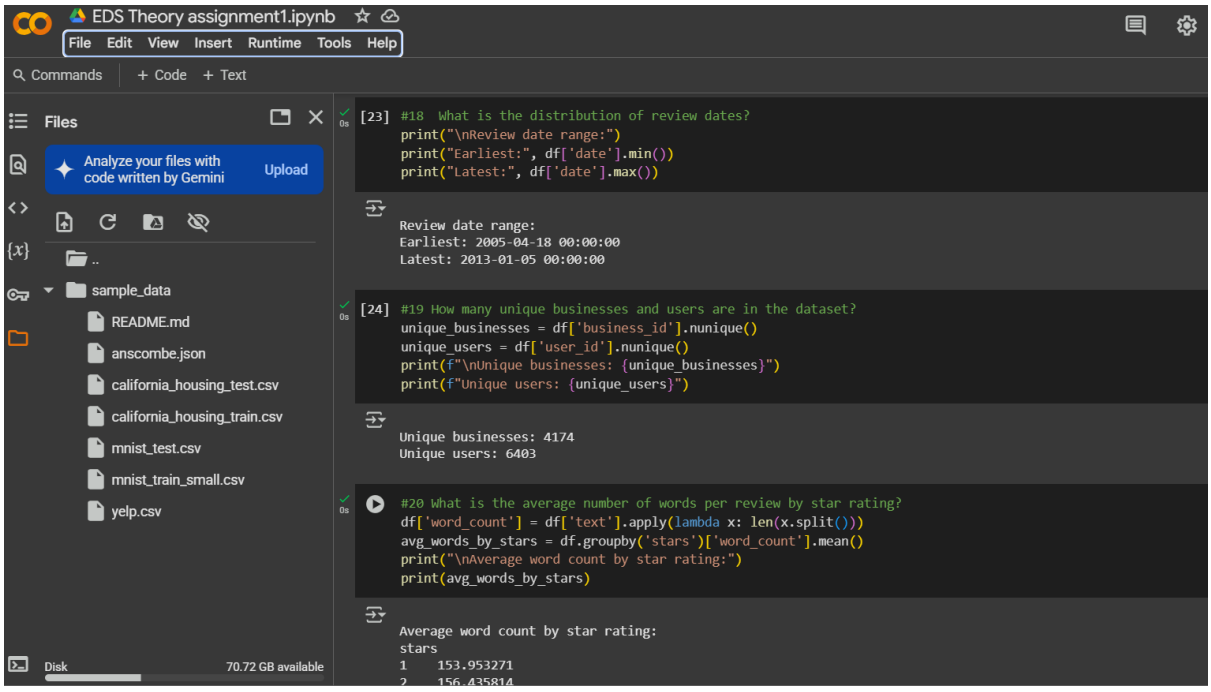

EDS Theory assignment1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Files
Analyze your files with code written by Gemini
Upload
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
yelp.csv
Disk 70.72 GB available
[14] #9 Which users are the most active reviewers?
top_reviewers = df['user_id'].value_counts().head(10)
print("\nMost active reviewers:")
print(top_reviewers)
Most active reviewers:
user_id
fczQCSmaWF78toLEmb0Zsw 38
0CMz8YaO3f8xu4KqQgKb9Q 25
90a6z--_CUrl84aCzZyPsg 22
0mqHhdKEdak_A1FBhFNXqA 18
4ozupHULqGyO42s3zNUzOQ 18
wHg1YkCzdZq9WBJOTRgxHQ 17
_PzSNcfrCjeBxSLXRoMmgQ 16
0bNXP9quoJEgyVZu9ipGgQ 16
uZbTb-u-GVjTa2gtQfry5g 15
joIzw_aUiNvBTuGoytrH7g 15
Name: count, dtype: int64
#10 How has the number of reviews changed over time?
df['date'] = pd.to_datetime(df['date'])
reviews_over_time = df.set_index('date').resample('M').size()
print("\nReviews over time:")
print(reviews_over_time.head())
Reviews over time:
date
2005-04-30 1
2005-05-31 0
2005-06-30 0

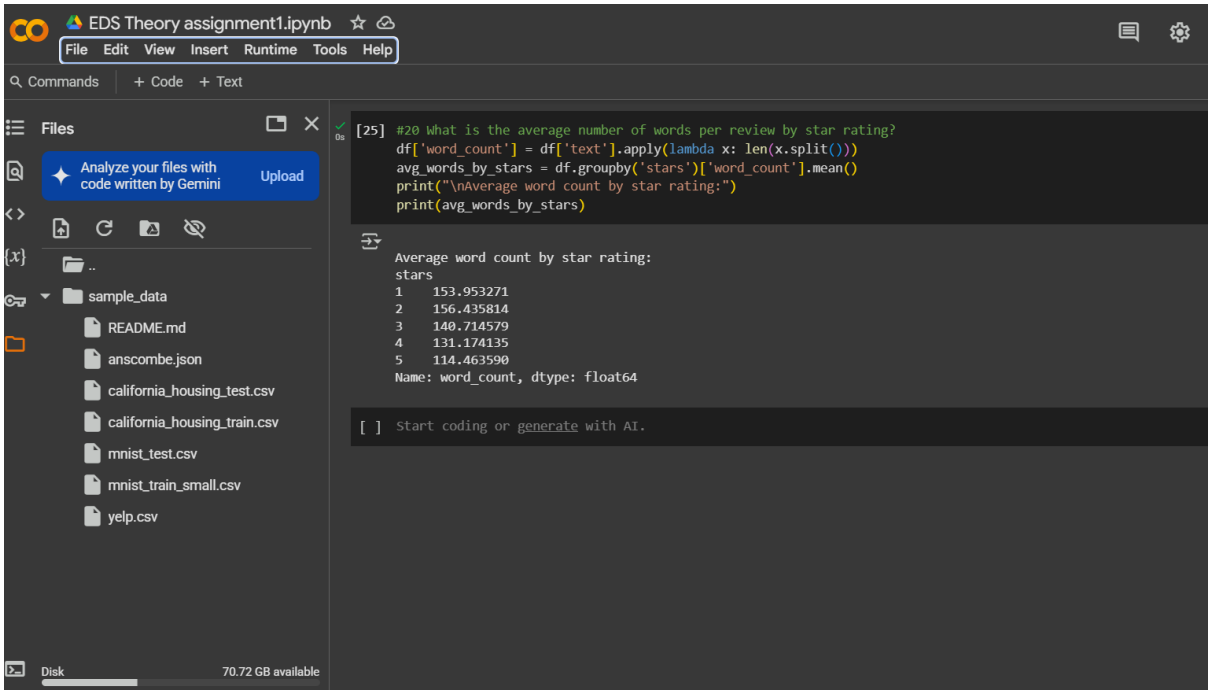

EDS Theory assignment1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Files
Analyze your files with code written by Gemini
Upload
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
yelp.csv
Disk 70.72 GB available
Reviews over time:
date
2005-04-30 1
2005-05-31 0
2005-06-30 0
2005-07-31 2
2005-08-31 0
Freq: ME, dtype: int64
<ipython-input-15-21a79b5be145>:3: FutureWarning: 'M' is deprecated and will be removed in a future versio
reviews_over_time = df.set_index('date').resample('M').size()
[16] #11 What is the average star rating over time?
avg_stars_over_time = df.set_index('date').resample('M')['stars'].mean()
print("\nAverage star rating over time:")
print(avg_stars_over_time.head())
Average star rating over time:
date
2005-04-30 5.0
2005-05-31 NaN
2005-06-30 NaN
2005-07-31 2.5
2005-08-31 NaN
Freq: ME, Name: stars, dtype: float64
<ipython-input-16-ff55968880cb>:2: FutureWarning: 'M' is deprecated and will be removed in a future versio
avg_stars_over_time = df.set_index('date').resample('M')['stars'].mean()
#12 What are the most common bigrams in positive reviews?
from nltk import bigrams


EDS Theory assignment1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Files
Analyze your files with code written by Gemini
Upload
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
yelp.csv
Disk 70.72 GB available
[17] #12 What are the most common bigrams in positive reviews?
from nltk import bigrams
positive_text = ' '.join(df[df['stars'] == 5]['text'].tolist()).lower()
words = re.findall(r'\w+', positive_text)
common_bigrams = Counter(bigrams(words)).most_common(10)
print("\nMost common bigrams in positive reviews:")
print(common_bigrams)
Most common bigrams in positive reviews:
[(('of', 'the'), 1302), (('in', 'the'), 1133), (('and', 'the'), 1127), (('it', 's'), 1119), (('this', 'pla
#13 How does the sentiment of reviews correlate with star ratings?
from textblob import TextBlob
df['sentiment'] = df['text'].apply(lambda x: TextBlob(x).sentiment.polarity)
sentiment_by_stars = df.groupby('stars')['sentiment'].mean()
print("\nAverage sentiment by star rating:")
print(sentiment_by_stars)
Average sentiment by star rating:
stars
1 -0.017906
2 0.096832
3 0.193733
4 0.277061
5 0.333293
Name: sentiment, dtype: float64


EDS Theory assignment1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Files
Analyze your files with code written by Gemini
Upload
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
yelp.csv
Disk 70.72 GB available
[19] #14 What is the distribution of review lengths?
print("\nReview length statistics:")
print(df['review_length'].describe())
Review length statistics:
count 10000.000000
mean 710.738700
std 617.399827
min 1.000000
25% 294.000000
50% 541.500000
75% 930.000000
max 4997.000000
Name: review_length, dtype: float64
#15 Which businesses have the highest average star ratings (with at least 5 reviews)?
business_stats = df.groupby('business_id').agg(
avg_stars=('stars', 'mean'),
review_count=('stars', 'count')
)
top_businesses = business_stats[business_stats['review_count'] >= 5].sort_values('avg_stars', ascending=Fal
print("\nBusinesses with highest average ratings (min 5 reviews):")
print(top_businesses)
Businesses with highest average ratings (min 5 reviews):
avg_stars review_count
business_id
rLAeltELaGdQKh_LYIllEA 5.000000 5
lVkDZZ8sTafPlq7f1i5row 5.000000 6


EDS Theory assignment1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Files
Analyze your files with code written by Gemini
Upload
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
yelp.csv
Disk 70.72 GB available
t4aP7ksa716XY6S4EsWFqw 4.833333 6
oyzxIqrtzu-8sWlPhJVsLw 4.833333 6
KTF-E3NfkJy2wiwcgOPyVQ 4.833333 6
AsSCv0q_BWqIe3mX2JqsOQ 4.833333 6
F8q_9PUl-Lwjj9xIRPArcg 4.833333 6
KBG28p3lGX17hOPoHhq5PQ 4.818182 11
c7VgGP8xT25OSReok6fwcQ 4.800000 5
[21] #16 What is the relationship between 'cool', 'funny', and 'useful' votes?
vote_correlation = df[['cool', 'funny', 'useful']].corr()
print("\nCorrelation between vote types:")
print(vote_correlation)
Correlation between vote types:
cool funny useful
cool 1.000000 0.764342 0.887102
funny 0.764342 1.000000 0.723406
useful 0.887102 0.723406 1.000000
#17 How do the longest reviews compare to the shortest in terms of star ratings?
longest_reviews = df.nlargest(10, 'review_length')
shortest_reviews = df.nsmallest(10, 'review_length')
print("\nAverage stars for longest reviews:", longest_reviews['stars'].mean())
print("Average stars for shortest reviews:", shortest_reviews['stars'].mean())
Average stars for longest reviews: 3.5
Average stars for shortest reviews: 4.2
[23] #18 What is the distribution of review dates?


EDS Theory assignment1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Files
Analyze your files with code written by Gemini
Upload
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
yelp.csv
Disk 70.72 GB available
[23] #18 What is the distribution of review dates?
print("\nReview date range:")
print("Earliest:", df['date'].min())
print("Latest:", df['date'].max())
Review date range:
Earliest: 2005-04-18 00:00:00
Latest: 2013-01-05 00:00:00
[24] #19 How many unique businesses and users are in the dataset?
unique_businesses = df['business_id'].nunique()
unique_users = df['user_id'].nunique()
print(f"\nUnique businesses: {unique_businesses}")
print(f"Unique users: {unique_users}")
Unique businesses: 4174
Unique users: 6403
#20 What is the average number of words per review by star rating?
df['word_count'] = df['text'].apply(lambda x: len(x.split()))
avg_words_by_stars = df.groupby('stars')['word_count'].mean()
print("\nAverage word count by star rating:")
print(avg_words_by_stars)
Average word count by star rating:
stars
1 153.953271
2 156.435814


EDS Theory assignment1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Files
Analyze your files with code written by Gemini
Upload
..
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
yelp.csv
Disk
70.72 GB available
[25] #20 What is the average number of words per review by star rating?
df['word_count'] = df['text'].apply(lambda x: len(x.split()))
avg_words_by_stars = df.groupby('stars')['word_count'].mean()
print("\nAverage word count by star rating:")
print(avg_words_by_stars)
Average word count by star rating:
stars
1 153.953271
2 156.435814
3 140.714579
4 131.174135
5 114.463590
Name: word_count, dtype: float64
[ ] Start coding or generate with AI.