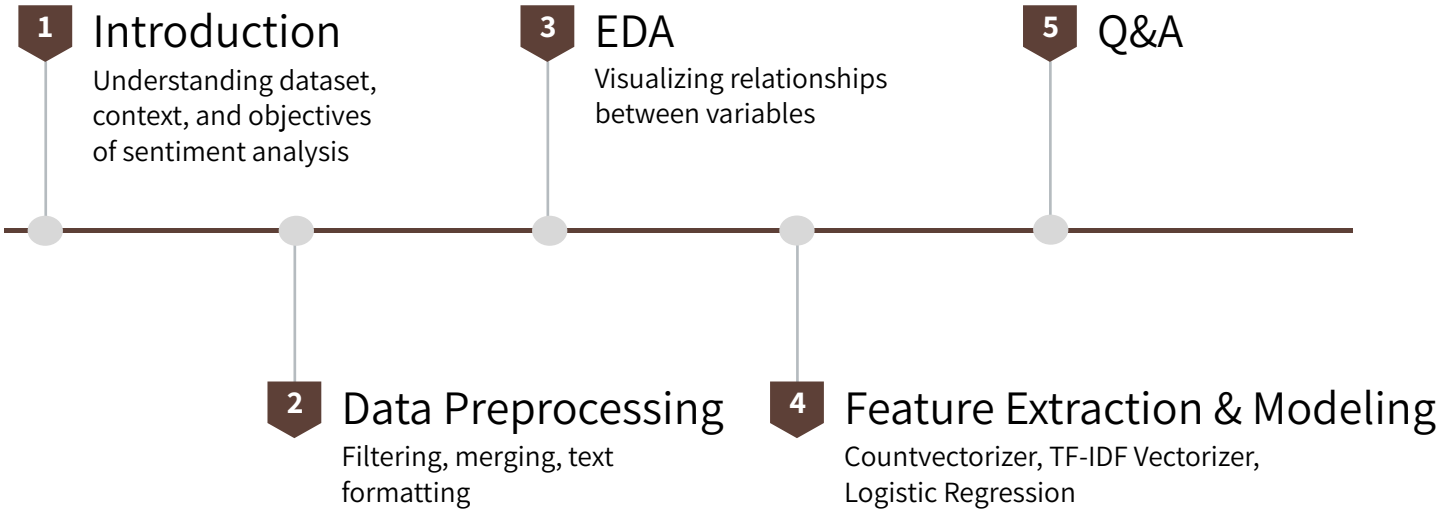
Yelp Sentiment Analysis

Group members : Duong, Priyanka, Ting-Yu

Presentation Roadmap



The Dataset

6,990,280 reviews

150,346 businesses

200,100 pictures

11 metropolitan areas

Sentiment Analysis

Natural language processing to determine positive and negative sentiments

Machine Learning

Automate sentiment labeling and create ability to ingest new data

Data Preprocessing

Step 1 : Business Data

- 14 columns
- selected only "restaurants" category

business_id	name	address	city	state	postal_code	latitude	longitude	stars	review_count	is_open	attributes	categories	hours
MTSW4McQd7CbVtyjqoe9mw	St Honore Pastries	935 Race St	Philadelphia	PA	19107	39.955505	-75.155564	4.0	80	1	{'RestaurantsDelivery': 'False' 'OutdoorSeati..	Restaurants, Food, Bubble Tea, Coffee & Tea, B...	{'Monday': '7:0-20:0', 'Tuesday': '7:0-20:0', ...
CF33F8-E6oudUQ46HnavjQ	Sonic Drive-In	615 S Main St	Ashland City	TN	37015	36.269593	-87.058943	2.0	6	1	{'BusinessParking': 'None' 'BusinessAcceptsCr..	Burgers, Fast Food, Sandwiches, Food, Ice Crea...	{'Monday': '0:0-0:0', 'Tuesday': '6:0-22:0', '...
k0hlBqXX-Bt0vf1op7Jr1w	Tsevi's Pub And Grill	8025 Mackenzie Rd	Affton	MO	63123	38.565165	-90.321087	3.0	19	0	{'Caters': 'True' 'Alcohol': 'u'full_bar'' '..	Pubs, Restaurants, Italian, Bars, American (Tr...	None

Step 2 : Reducing Data Size

- the top five cities with the highest number of reviews

	state	city	review_count
800	PA	Philadelphia	665749
313	LA	New Orleans	465988
927	TN	Nashville	318560
163	FL	Tampa	293130
266	IN	Indianapolis	242024

Step 3 : Review Data

- 9 columns
- retaining only 'user id,' 'business id,' 'stars,' 'text,' and 'date.'

	review_id	user_id	business_id	stars	useful	funny	cool	text	date
0	KU_O5udG6zpxOg-VcAEodg	mh_-eMZ6K5RLWhZyISBhwA	XQfwVwDr-v0ZS3_CbbE5Xw	3	0	0	0	If you decide to eat here, just be aware it is...	2018-07-07 22:09:11
1	BiTunyQ73aT9WBnpR9DZGw	OyoGAe7OKpv6SyGZT5g77Q	7ATYjTIgM3jUlt4UM3IypQ	5	1	0	1	I've taken a lot of spin classes over the year...	2012-01-03 15:28:18
2	saUsX_uimxRlCVr67Z4Jig	8g_iMtfSiwikVnbP2etR0A	YjUWPpI6HXG530lwP-fb2A	3	0	0	0	Family diner. Had the buffet. Eclectic assortm...	2014-02-05 20:30:30
3	AqPFMleE6RsU23_auESxiA	_7bHUi9Uuf5__HHc_Q8guQ	kxX2SOes4o-D3ZQBkiMRfA	5	1	0	1	Wow! Yummy, different, delicious. Our favo...	2015-01-04 00:01:03
4	Sx8TMOWLNuJBWer-0pcmoA	bcjbaE6dDog4jkNY91ncLQ	e4Vwtrqf-wpJfwesgvdgxQ	4	1	0	1	Cute interior and owner (?) gave us tour of up...	2017-01-14 20:54:15

Step 4 : Data Merging

- merged the review data and business data by the "business_id"

Step 5 : Sentiment Labeling

- 'stars' column to a sentiment indicator
- removed reviews with 3 stars
- labeled reviews with 4 or 5 stars as 1 (positive)
- labeled reviews with 1 or 2 stars as 0 (negative).

restaurant_reviews	categories	sentiment
This is the second time we tried turning point...	Restaurants, Breakfast & Brunch, Food, Juice B...	0.0
The place is cute and the staff was very frien...	Restaurants, Breakfast & Brunch, Food, Juice B...	1.0
Mediocre at best. The decor is very nice, and ...	Restaurants, Breakfast & Brunch, Food, Juice B...	0.0
When I was shown to my seat of was still wet s...	Restaurants, Breakfast & Brunch, Food, Juice B...	0.0
I went on a Thursday morning for breakfast. St...	Restaurants, Breakfast & Brunch, Food, Juice B...	1.0

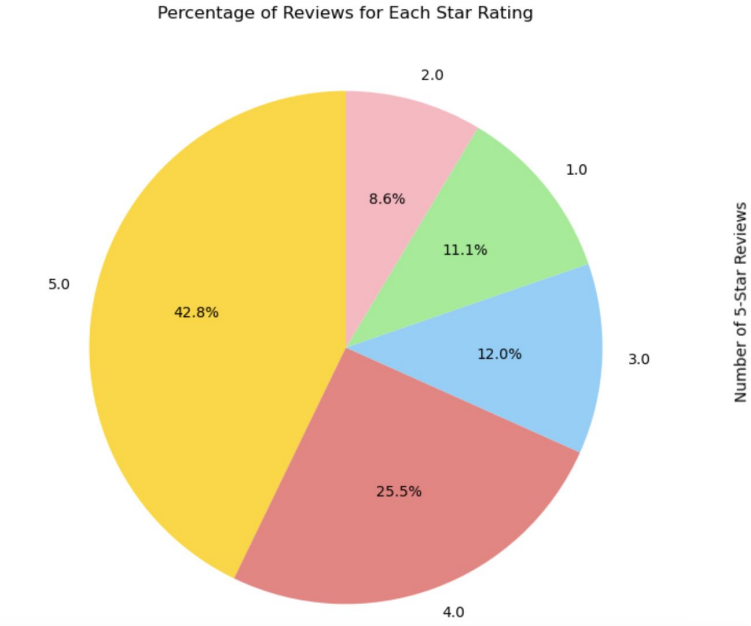
Step 6 : Text Formatting

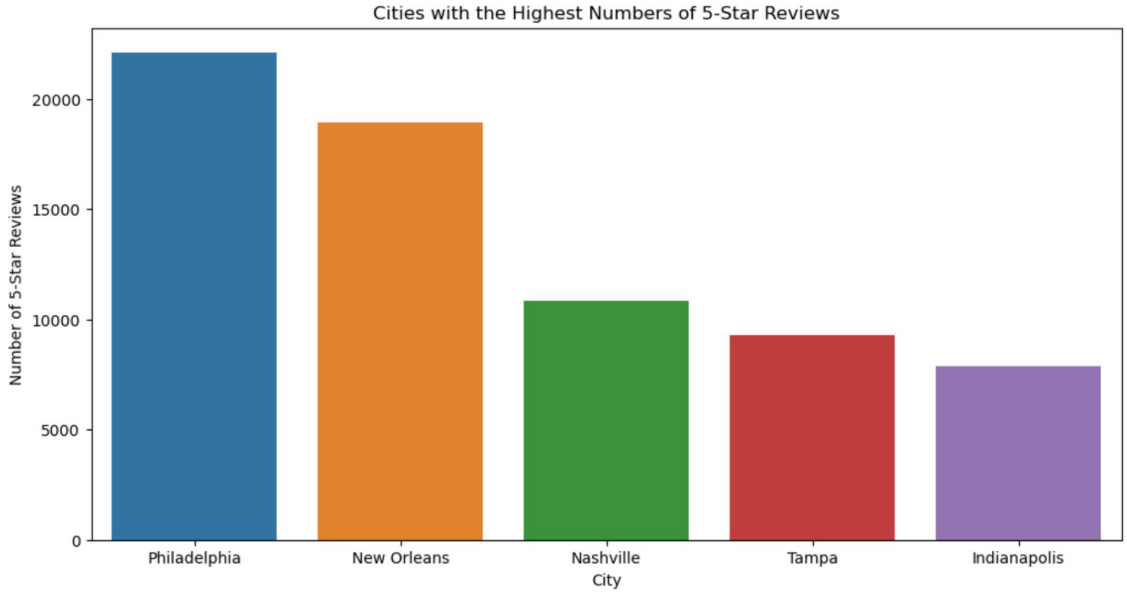
- regular expressions : eliminating characters like punctuation, numbers, and whitespaces.
- NLTK : including the removal of stopwords, stemming, and lemmatization.

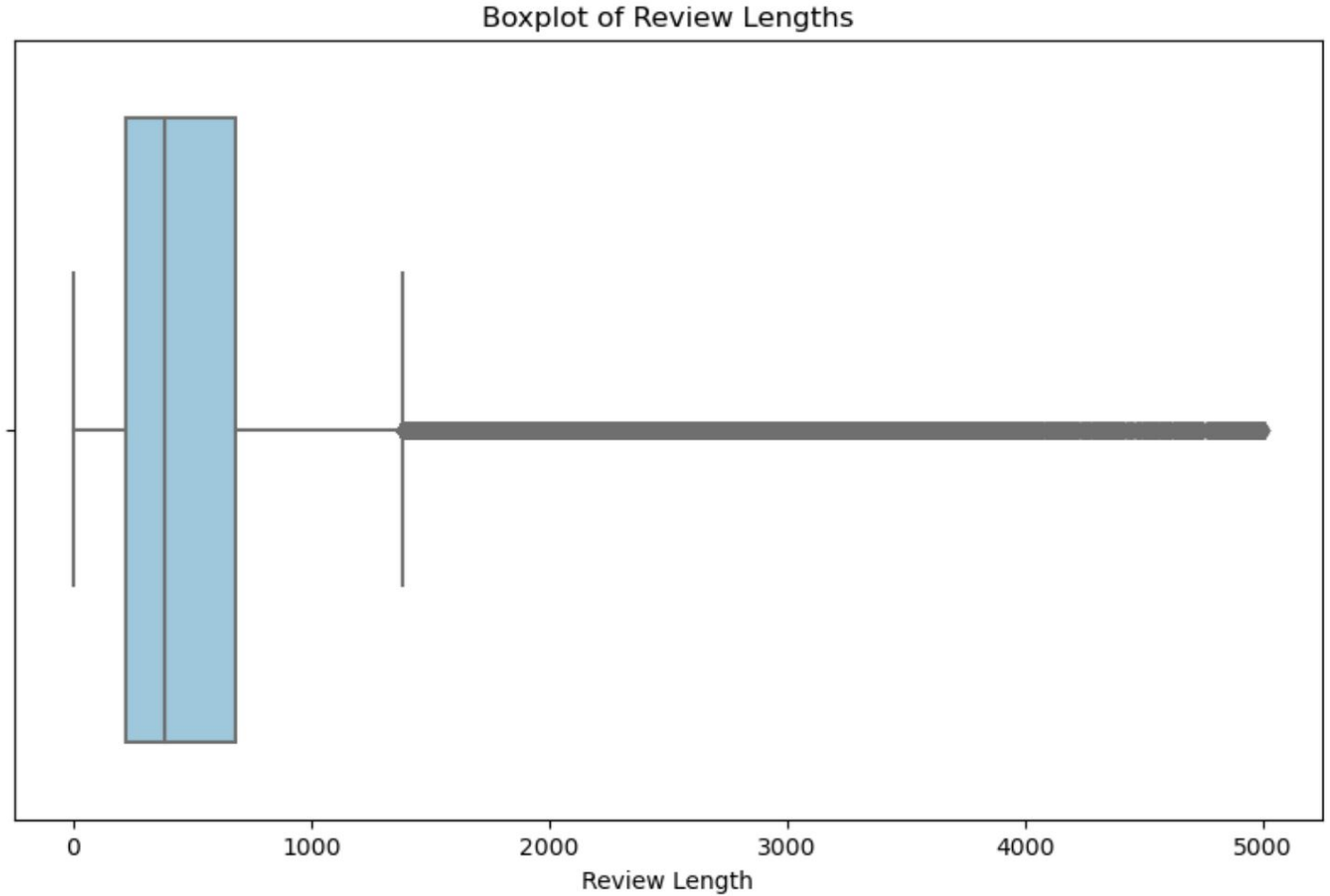
reviews_text	cleaned_reviews
This is nice little Chinese bakery in the hear...	[thi, nice, littl, chine, bakeri, heart, phila...
This is the bakery I usually go to in Chinatow...	[thi, bakeri, i, usual, go, chinatown, they, d...
A delightful find in Chinatown Very clean and ...	[a, delight, find, chinatown, veri, clean, kin...
I ordered a graduation cake for my niece and i...	[i, order, graduat, cake, niec, came, absolut,...
HKSTYLE MILK TEA FOUR STARS\n\nNot quite sure...	[hkstyle, milk, tea, four, star, not, quit, su...
...	...
Ordered delivery for some tacos on a Saturday ...	[order, deliveri, taco, saturday, night, food,...
First time trying this restaurant and I had a ...	[first, time, tri, restaur, i, good, experi, i...
This restaurant is truly amazing The owner is ...	[thi, restaur, truli, amaz, the, owner, nice, ...
Recently got take out from adelita they were g...	[recent, got, take, adelita, great, super, fas...
Another pretty good Mexican place to add into ...	[anoth, pretti, good, mexican, place, add, mix...

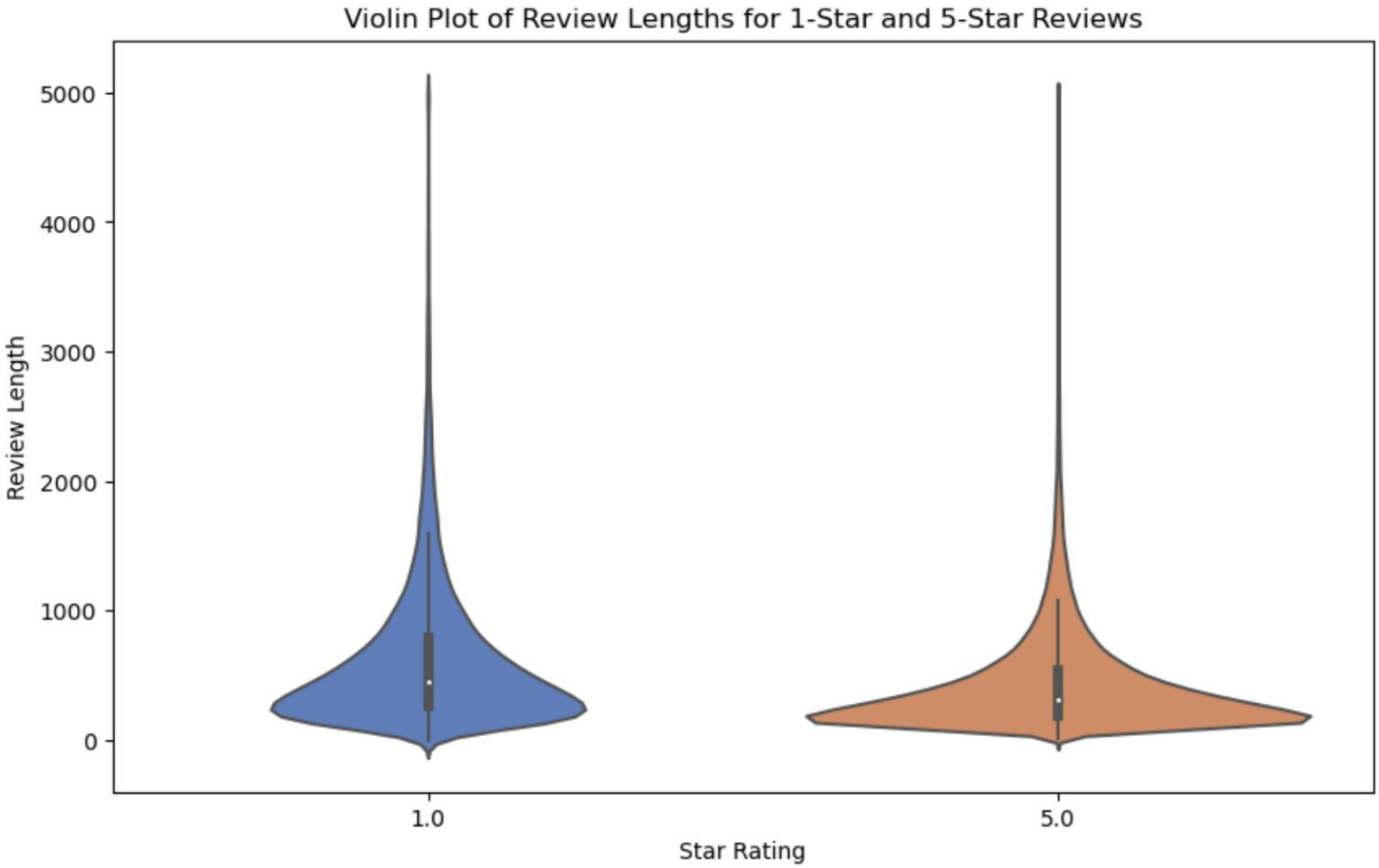
Step 7 : Data Storage

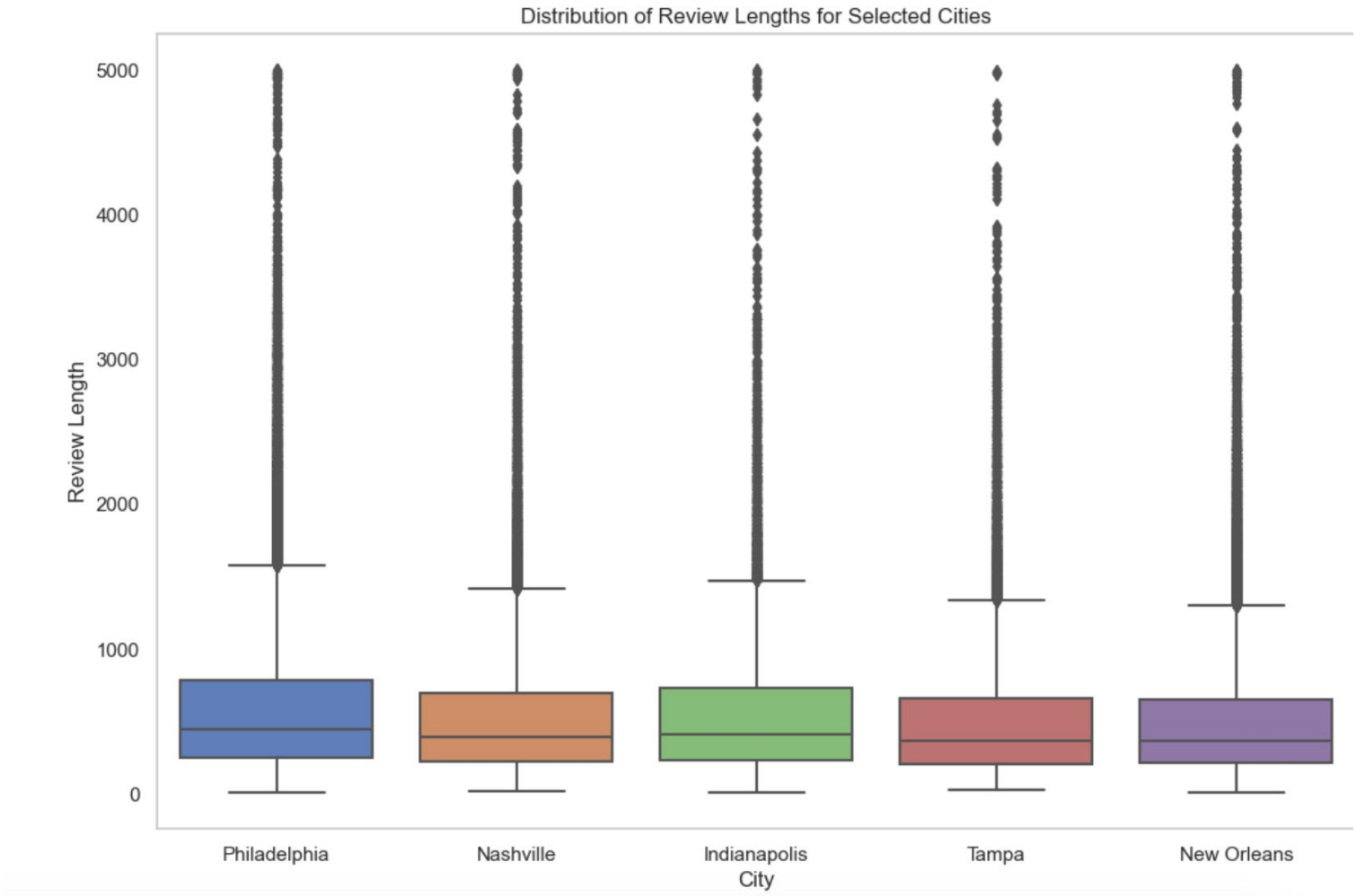
- saved the organized data as a CSV file


Percentage of Reviews for Each Star Rating
2.0
1.0
3.0
4.0
5.0
8.6%
11.1%
12.0%
25.5%
42.8%


Cities with the Highest Numbers of 5-Star Reviews
Number of 5-Star Reviews
0
5000
10000
15000
20000
Philadelphia
New Orleans
Nashville
Tampa
Indianapolis
City


Boxplot of Review Lengths
0
1000
2000
3000
4000
5000
Review Length


Violin Plot of Review Lengths for 1-Star and 5-Star Reviews
Review Length
5000
4000
3000
2000
1000
0
1.0
5.0
Star Rating


Distribution of Review Lengths for Selected Cities
Review Length
5000
4000
3000
2000
1000
0
Philadelphia
Nashville
Indianapolis
Tampa
New Orleans
City

Feature extraction, model training and prediction

Count Vectorizer

Vectorization is the process of turning a collection of text documents into numerical feature vectors.

CountVectorizer tokenizes the text and counts the frequency of each token. The result is a sparse matrix where each cell contains the count of the word in that particular review.

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(max_features=17000)
X_train_counts = vectorizer.fit_transform(X_train)
```

```
print(X_train_counts)
```

```
  (0, 15109)    1
  (0, 7345)     1
  (0, 8566)     1
  (0, 15380)    1
  (0, 5238)     2
```

```
feature_names = vectorizer.get_feature_names_out()
print(feature_names)
```

```
['aa' 'aaa' 'aah' ... 'zuppa' 'zuzu' 'zydeco']
```

```
print(vectorizer.vocabulary_)
```

```
{'thoroughli': 15109, 'impress': 7345, 'littl': 8566, 'tow
7666, 'refresh': 12269, 'polit': 11482, 'courteou': 3392,
ood': 6232, 'thing': 15087, 'would': 16739, 'highli': 6917
'gener': 6032, 'thi': 15072, 'must': 9839, 'nola': 10135,
'got': 6276, 'oyster': 10738, 'po': 11444, 'boy': 1694, 'm
```

TF-IDF Vectorizer

- Term Frequency (TF): calculated as the number of times a word appears in the document divided by the total number of words in the document.
- Inverse Document Frequency (IDF): This measures the significance of a word across a set of documents. It's calculated as the logarithm of the total number of documents divided by the number of documents containing the word.
- TF-IDF Score: This is the product of TF and IDF.
- The more frequently a word appears in a document, the higher its TF value, indicating its importance in that document.Words that appear frequently across many documents (like "this", "what", "if") are considered less important, hence they have lower IDF scores.

```
tfidfVectorizer=TfidfVectorizer(max_features=17000)
X_train_counts=tfidfVectorizer.fit_transform(X_train)
```

Example of TF-IDF

```
corpus = ['second time try turn point location time long wait food order time long wait minute omelette skillet hardly egg f
          'place cute staff friendly nice menu good brunch lunch seat right away enjoy avocado toast bacon nice brunch place
```

```
['avocado' 'away' 'bacon' 'blt' 'brunch' 'chop' 'cute' 'eat' 'egg' 'enjoy'
 'experience' 'feel' 'find' 'food' 'friendly' 'good' 'hard' 'hardly'
 'like' 'location' 'long' 'lunch' 'mainly' 'menu' 'minute' 'nearby' 'nice'
 'omelette' 'onion' 'order' 'overall' 'place' 'point' 'right' 'seat'
 'second' 'skillet' 'staff' 'stressful' 'suppose' 'time' 'toast' 'tomato'
 'try' 'turn' 'wait' 'wife']
[[0.08659802 0.         0.         0.12171049 0.         0.24342099
  0.         0.12171049 0.12171049 0.         0.12171049 0.12171049
  0.12171049 0.12171049 0.         0.         0.12171049 0.12171049
  0.12171049 0.12171049 0.36513148 0.         0.12171049 0.
  0.12171049 0.         0.         0.12171049 0.12171049 0.12171049
  0.12171049 0.         0.12171049 0.         0.         0.12171049
  0.12171049 0.         0.12171049 0.12171049 0.48684198 0.
  0.24342099 0.12171049 0.12171049 0.36513148 0.12171049]
 [0.14088238 0.19800527 0.19800527 0.         0.39601054 0.
  0.19800527 0.         0.         0.19800527 0.         0.
  0.         0.         0.19800527 0.19800527 0.         0.
  0.         0.         0.         0.19800527 0.         0.19800527
  0.         0.19800527 0.39601054 0.         0.         0.
  0.         0.39601054 0.         0.19800527 0.19800527 0.
  0.         0.19800527 0.         0.         0.         0.19800527
  0.         0.         0.         0.         0.        ]]
```

food
good
the
place
great
order
time
best
back
thi
definit
recommend
alway
staff
servic
tri
come
wait
nice
restaur
fri
also
go
get
like
it
love
amaz
drink
would
we
delici
chicken
realli
one
got
chees
meal
well
price
sandwich
look
make
menu
bar
flavor
friendli
pizza
they
coffe
tast
could
didnt
eat
side
lunch
perfect
ask
ive
friend
atmospher
went
night
first
dish
visit
hour
peopl
made
came
worth
experi
dinner
littl
dont
fresh
awesom
seat
beer
say
my
server
want
burger
even
thing
shrimp
hot
custom
excel
salad
take
much
enjoy
spot
sauc
new
never
better
breakfast
disappoint
dessert
rice
sweet
bread
reason
sure
wasnt
help
you
everi
tasti
option
check
next
said
though
find
work
cook
sushi
someth
that
brunch
our
two
see
star
know
nashvil
walk
area
bad
still
there
lot
sinc
wonder
need
maniday
way
line
waitress
orlean
happi
fantast
stop
so
highli
think
portion
veri
took
oyster
pretti
give
around
busi
select
last
meat
your
taco
qualiti
live
steak
expect
small
locat
ever
roll
super
review
and
not
soup
feel
long
pork
special
top
favorit
im
serv
differ
right
minut
absolut
philli
egg
dine
must
bit
open
but
cant
if

ML Modeling

- Used Count Vectorizer and TF-IDF Vectorizer to convert the yelp reviews into feature vectors.
- Model Training: Trained the machine learning model on the features extracted from the reviews using Logistic Regression and XGBoost using a 80/20 split of train and test data
- Prediction & Evaluation: Used the trained model to predict the sentiment of test reviews and evaluated the model's performance.

Logistic Regression

```
print(classification_report(y_test, predictions, digits=4))
```

	precision	recall	f1-score	support
0.0	0.9247	0.8859	0.9049	73671
1.0	0.9709	0.9814	0.9761	286003
accuracy			0.9619	359674
macro avg	0.9478	0.9337	0.9405	359674
weighted avg	0.9615	0.9619	0.9615	359674

```
Confusion Matrix for XGBoost Classifier:
[[ 59310  14361]
 [  5218 280785]]
Score:  94.56
Classification Report:
```

	precision	recall	f1-score	support
0.0	0.92	0.81	0.86	73671
1.0	0.95	0.98	0.97	286003
accuracy			0.95	359674
macro avg	0.94	0.89	0.91	359674
weighted avg	0.94	0.95	0.94	359674

Questions?