YELP RESTAURANT STARS PREDICTION

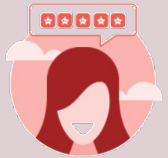
AIM & WORK FLOW

- **AIM:** Predict the stars of restaurants based on the accessible data
- **Work Flow:**
- Data cleaning
- Data visualization
- Model training
- Predication

DATA CLEANING

- Data Source
  https://www.yelp.com/dataset/documentation/main

The Dataset


6,990,280 reviews


150,346 businesses


200,100 pictures


11 metropolitan areas

908,915 tips by 1,987,897 users
Over 1.2 million business attributes like hours, parking, availability, and ambience
Aggregated check-ins over time for each of the 131,930 businesses

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150346 entries, 0 to 150345
Data columns (total 14 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   business_id   150346 non-null  object
 1   name          150346 non-null  object
 2   address       150346 non-null  object
 3   city          150346 non-null  object
 4   state         150346 non-null  object
 5   postal_code   150346 non-null  object
 6   latitude      150346 non-null  float64
 7   longitude     150346 non-null  float64
 8   stars         150346 non-null  float64
 9   review_count  150346 non-null  int64
 10  is_open       150346 non-null  int64
 11  attributes    136602 non-null  object
 12  categories    150243 non-null  object
 13  hours         127123 non-null  object
dtypes: float64(3), int64(2), object(9)
memory usage: 16.1+ MB
```

- Compressed Columns
- Redundant information
- Missing data
- Meaningless Dtype
- Business other than restaurant

Hours

```
{'Monday': '0:0-0:0',
 'Tuesday': '8:0-18:30',
 'Wednesday': '8:0-18:30',
 'Thursday': '8:0-18:30',
 'Friday': '8:0-18:30',
 'Saturday': '8:0-14:0'}
```

Attributes

```
{'BusinessParking': 'None',
 'BusinessAcceptsCreditCards': 'True',
 'RestaurantsAttire': "u'casual'",
 'OutdoorSeating': 'True',
 'RestaurantsReservations': 'False',
 'Caters': 'False',
 'RestaurantsTakeOut': 'True',
 'Alcohol': "u'none'",
 'Ambience': 'None',
 'GoodForKids': 'True',
 'RestaurantsPriceRange2': '1',
 'ByAppointmentOnly': 'False',
 'CoatCheck': 'False',
 'DogsAllowed': 'False',
 'RestaurantsTableService': 'False',
 'RestaurantsGoodForGroups': 'True',
 'RestaurantsDelivery': 'True',
 'WiFi': "u'no'",
 'WheelchairAccessible': 'True',
 'HasTV': 'True',
 'HappyHour': 'False',
 'DriveThru': 'True',
 'BikeParking': 'False'}
```

- Unfold daily open and close time into 4 columns :

Average weekday open time

Average weekday close time

Average weekend open time

Average weekend close time

- Unfold attributes into different columns and only keep those who are related to restaurant.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 34987 entries, 3 to 150339
Data columns (total 23 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   business_id              34987 non-null  object
 1   name                     34987 non-null  object
 2   address                  34987 non-null  object
 3   city                     34987 non-null  object
 4   state                    34987 non-null  object
 5   postal_code              34987 non-null  object
 6   latitude                 34987 non-null  float64
 7   longitude                34987 non-null  float64
 8   stars                    34987 non-null  float64
 9   review_count             34987 non-null  int64
 10  is_open                  34987 non-null  int64
 11  RestaurantsPriceRange2   34987 non-null  object
 12  RestaurantsTakeOut       34987 non-null  object
 13  RestaurantsDelivery      34987 non-null  object
 14  opendays_work            34987 non-null  int64
 15  opendays_weekend         34987 non-null  int64
 16  open_workday             34987 non-null  float64
 17  close_workday            34987 non-null  float64
 18  open_weekend             34987 non-null  float64
 19  close_weekend            34987 non-null  float64
 20  categories               34987 non-null  object
 21  Service                  34987 non-null  int64
 22  Food                     34987 non-null  bool
dtypes: bool(1), float64(7), int64(5), object(10)
memory usage: 6.2+ MB
```

- Is_open = True
- Is restaurant = True
- Drop meaningless and duplicate columns
- Fill in missing data

For service, if the service column is None, suppose the restaurant does no have the certain service.

For opening time, if none, suppose open at 24 and close at 0

text	topics	sentiment
This is nice little Chinese bakery in the hear...	[bubble, def, look, disrespected, slowly, deci...	0.224
I have always liked Sonic as it is good fresh ...	[chance, made, close, ones, numbers, 10, liked...	-0.011
Waited several minutes waiting to order. I was...	[chance, look, show, times, man, close, coupon...	0.014
I eat pho about 4 times a week and from a spec...	[chance, made, times, folks, either, fish, fan...	0.218
Went there at 4am and there was only one waitr...	[look, starving, understand, muscle, nets, one...	0.035

	text
0	This is nice little Chinese bakery in the hear...
1	I have always liked Sonic as it is good fresh ...
2	Waited several minutes waiting to order. I was...
3	I eat pho about 4 times a week and from a spec...
4	Went there at 4am and there was only one waitr...
...	...
34982	I must say that this place is amazing. Comfort...
34983	I only stop at this WaWa during off hours as g...
34984	I've been a Starbucks queen ever since I can r...
34985	I do not know why Adelita is not packed. There...
34986	Every single item on their menu is delicious! ...

- **Sentiment score**

-1 Most Negative

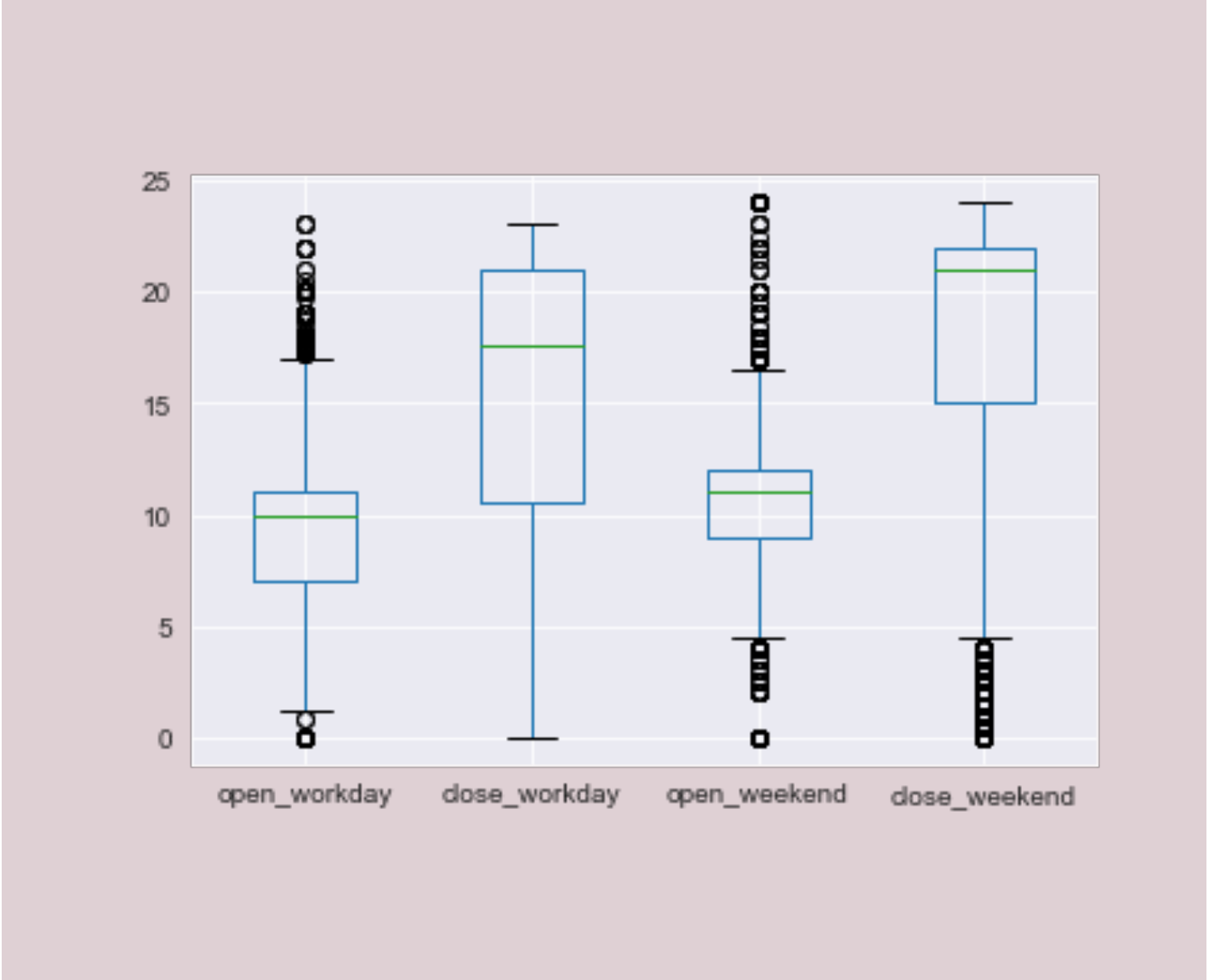
+1 Most Positive

- **“Topics”**

Most frequent 5 words in each review used for tfidf

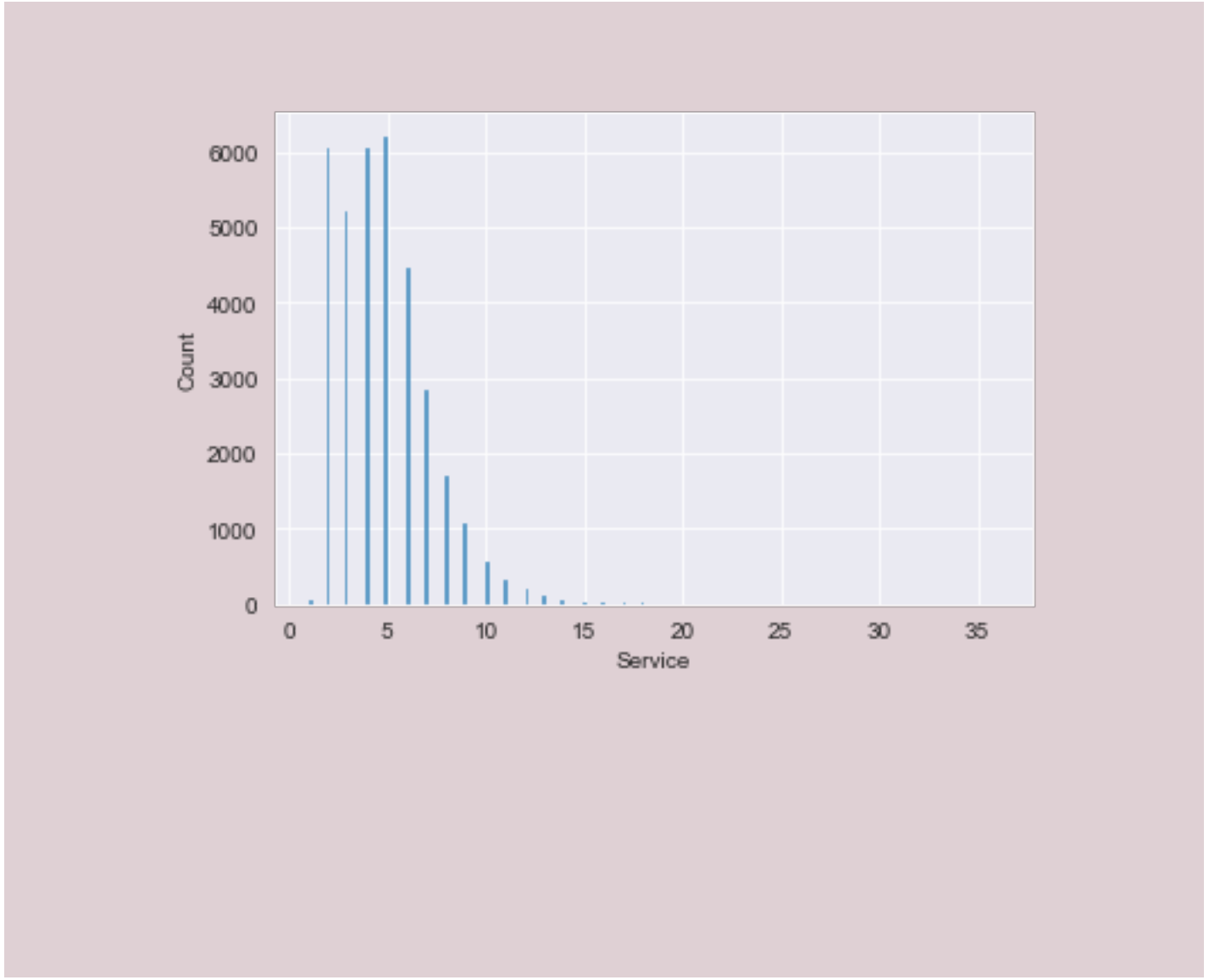
DATA VISUALIZATION

SERVICE TIME



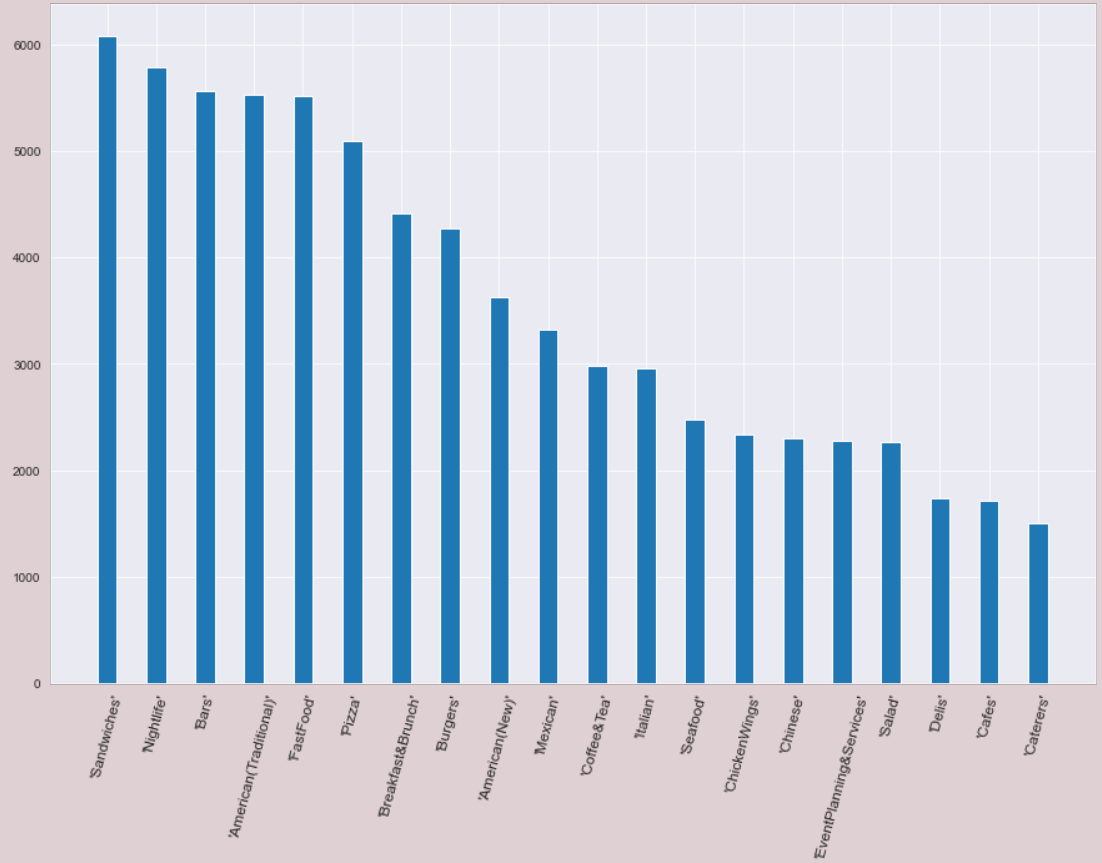
DATA VISUALIZATION

SERVICE TYPE I



DATA VISUALIZATION

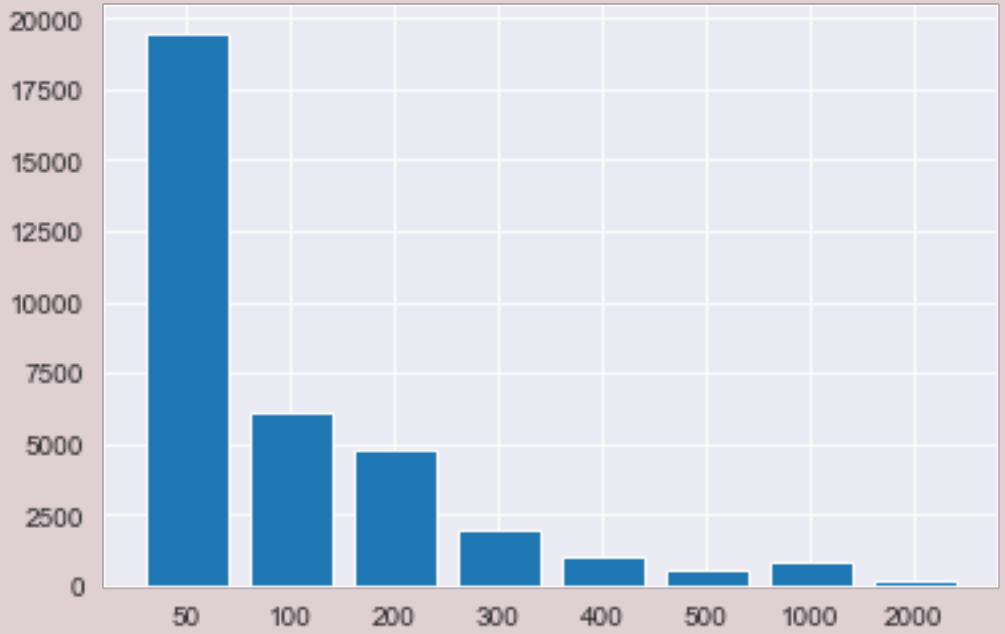
SERVICE TYPE II



Sandwiches	Nightlife	Bars	American(Traditional)	FastFood
Pizza	Breakfast&Brunch	Burgers	American(New)	Mexican
Coffee&Tea	Italian	Seafood	ChickenWings	Chinese
EventPlanning&Services		Salad	Delis	Cafes

DATA VISUALIZATION

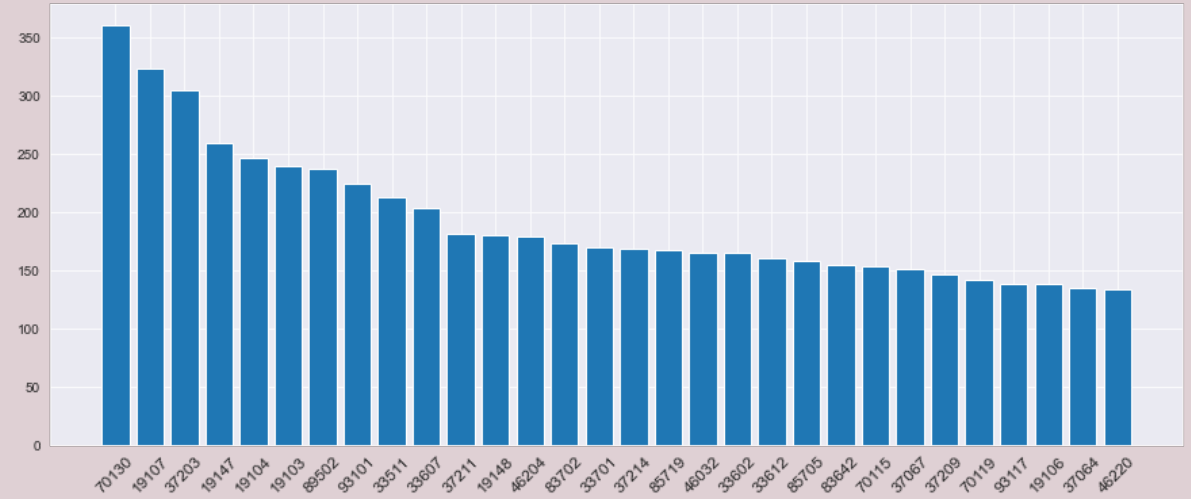
REVIEWS



	name	review_count
26439	Acme Oyster House	7568
26163	Oceana Grill	7400
21327	Hattie B’s Hot Chicken - Nashville	6093
33302	Reading Terminal Market	5721
34209	Ruby Slipper - New Orleans	5193

DATA VISUALIZATION

LOCATIONS



70130	New Orleans, LA,	California
19107	Philadelphia,	Pennsylvania
37203	Nashville,	Tennessee
19147	Philadelphia,	Pennsylvania
19104	Philadelphia,	Pennsylvania
19103	Philadelphia,	Pennsylvania
89502	Reno,	NV
93101	Santa Barbara,	California
33511	Hillsborough County,	Florida
33607	Tampa,	Florida
37211	Nashville,	Tennessee
19148	Philadelphia,	Pennsylvania
46204	Indianapolis,	Indiana
83702	Ada County,	Idaho
33701	St. Petersburg,	Florida
37214	Nashville,	Tennessee
85719	Tucson,	Arizona
46032	Hamilton County,	Indiana
33602	Tampa,	Florida
33612	Hillsborough County,	Florida
85705	Pima County,	Arizona
83642	Ada County,	Idaho
70115	New Orleans, LA,	California
37067	Williamson County,	Tennessee
37209	Nashville,	Tennessee
70119	New Orleans,	Louisiana
93117	Santa Barbara County,	California
19106	Philadelphia,	Pennsylvania
37064	Williamson County,	Tennessee
46220	Marion County,	Indiana

DATA VISUALIZATION

WORD CLOUD

4 stars

2 stars

3 stars

5 stars

1 star

TOPICS IN DIFFERENT RESTAURANTS

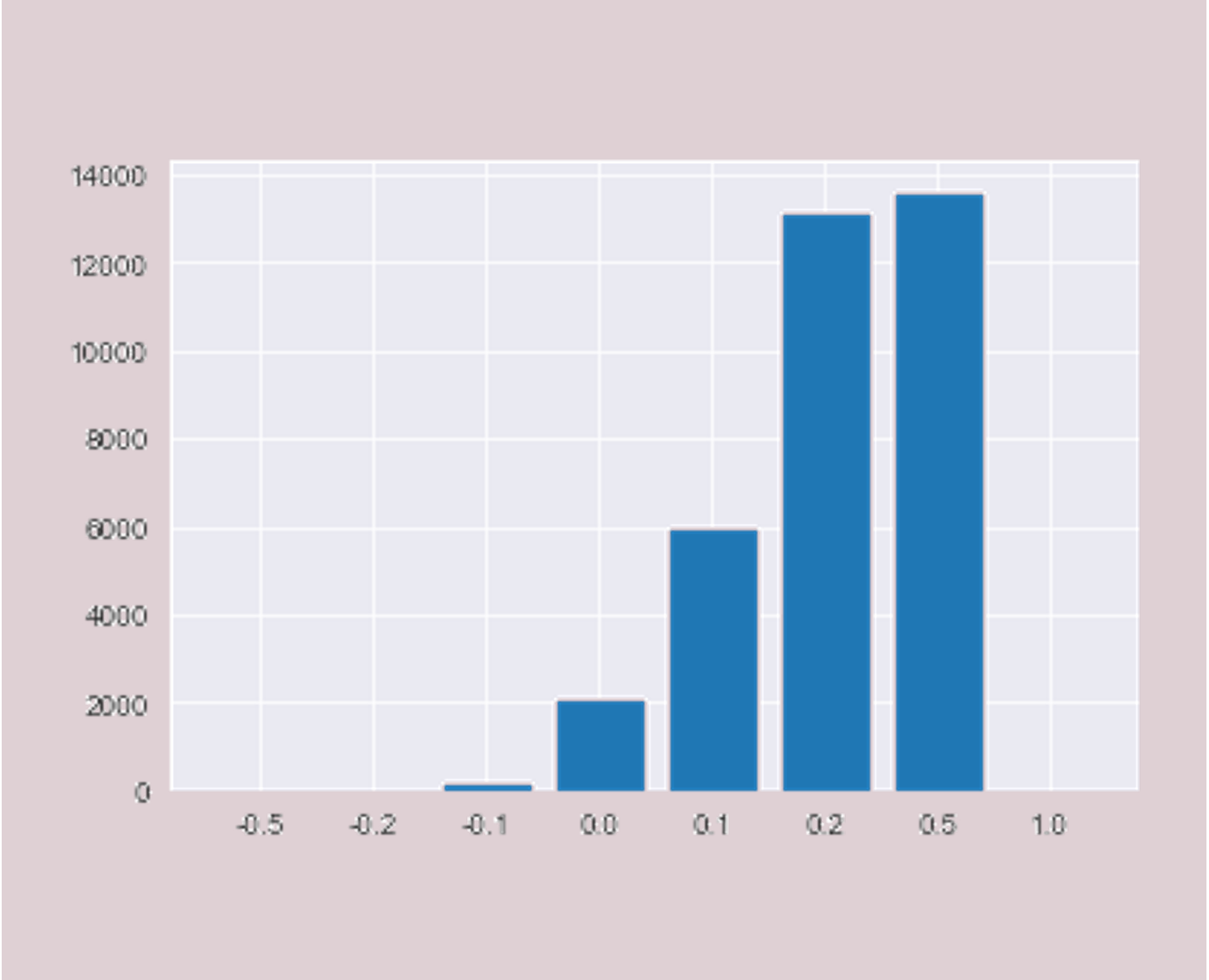
Fast Food

Chinese

French

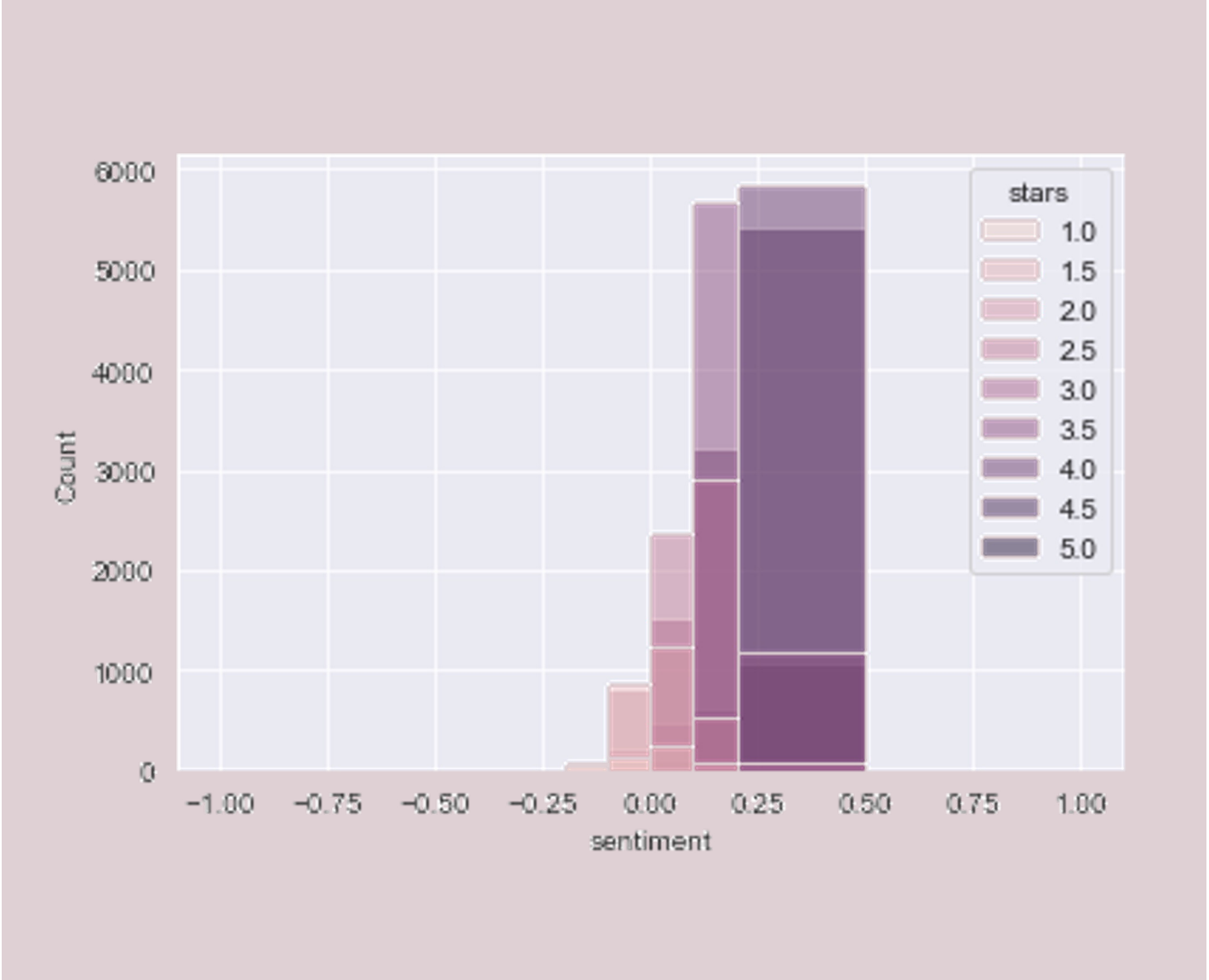
DATA VISUALIZATION

SENTIMENT



DATA VISUALIZATION

SENTIMENT



	Features
0	review_count
1	Service
2	RestaurantsDelivery
3	RestaurantsPriceRange2
4	RestaurantsReservations
5	Alcohol
6	HappyHour
7	OutdoorSeating
8	opendays_weekend
9	open_workday
10	close_workday
11	open_weekend
12	close_weekend

METHODS(WITHOUT NLP METHODS)

- Linear Regression
- Decision Tree Regression
- Gradient Boosting Regressor
- Stacking Regressors
- Gradient Boosting Classifiers
- Random Forest Classifiers

CONCLUSION

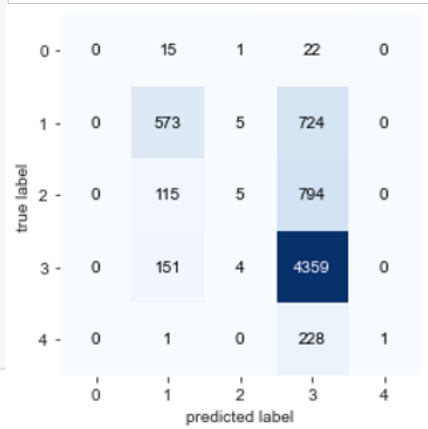
CLASSIFIERS HAVE BETTER PERFORMANCE WITH 5 CLASSES

WITH 5 CLASSES
1.0
2.0
3.0
4.0
5.0

RandomForestClassifier

```
from sklearn.ensemble import RandomForestClassifier
param_grid = {
              'n_estimators': [10,50,100],
               'max_depth':[3,5,8,10]}
rfc_gscv = GridSearchCV(estimator=RandomForestClassifier(),
                        param_grid=param_grid,
                        cv=3,
                        n_jobs=-1).fit(X_train,y_train)

print(f'best parameter setting found: {rfc_gscv.best_params_}')
print(f'best training score found   : {rfc_gscv.best_score_.round(3)}')

rfc_gscv_test_score = rfc_gscv.score(X_test,y_test)

print(f'rfc_gscv test score  : {rfc_gscv_test_score.round(3)}')
```

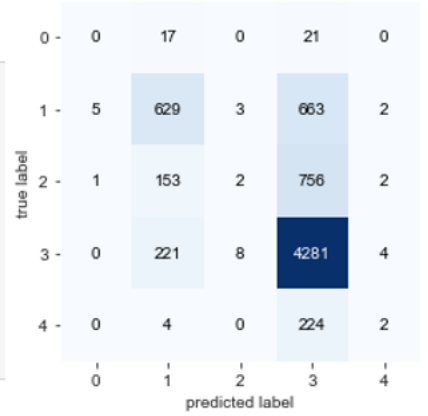
```
best parameter setting found: {'max_depth': 10, 'n_estimators': 100}
best training score found   : 0.706
rfc_gscv test score  : 0.705
```



GradientBoostingClassifier

```
from sklearn.ensemble import GradientBoostingClassifier
param_grid = {
              'n_estimators': [10,50,100,200]}
gbc_gscv = GridSearchCV(estimator=GradientBoostingClassifier(),
                        param_grid=param_grid,
                        cv=3,
                        n_jobs=-1).fit(X_train,y_train)

print(f'best parameter setting found: {gbc_gscv.best_params_}')
print(f'best training score found   : {gbc_gscv.best_score_.round(3)}')

gbc_gscv_test_score = gbc_gscv.score(X_test,y_test)
```

```
best parameter setting found: {'n_estimators': 200}
best training score found   : 0.704
```

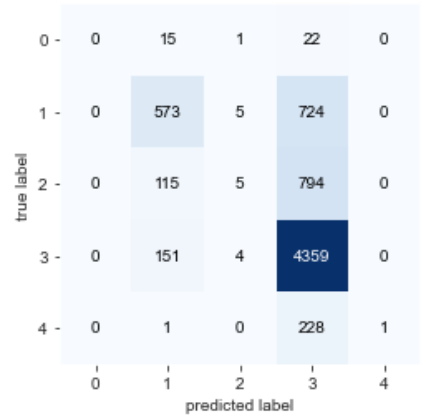


CONCLUSION

CLASSIFIERS HAVE BETTER PERFORMANCE WITH 5 CLASSES

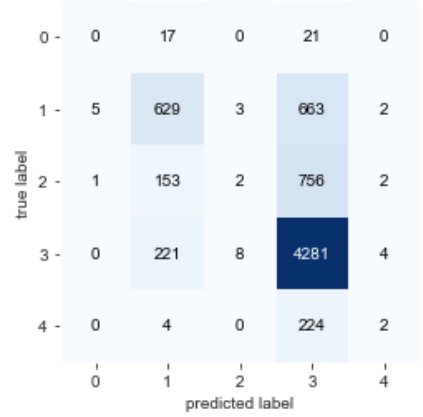
- Random Forest Classifiers

	factors	importance
0	review_count	0.318554
1	RestaurantsDelivery	0.089516
2	RestaurantsPriceRange2	0.114135
3	open_weekend	0.170404
4	close_weekend	0.214273
5	Service	0.093119



- Gradient Boosting Classifiers

	factors	importance
0	review_count	0.354094
1	RestaurantsDelivery	0.120427
2	RestaurantsPriceRange2	0.101749
3	open_weekend	0.154940
4	close_weekend	0.215799
5	Service	0.052991

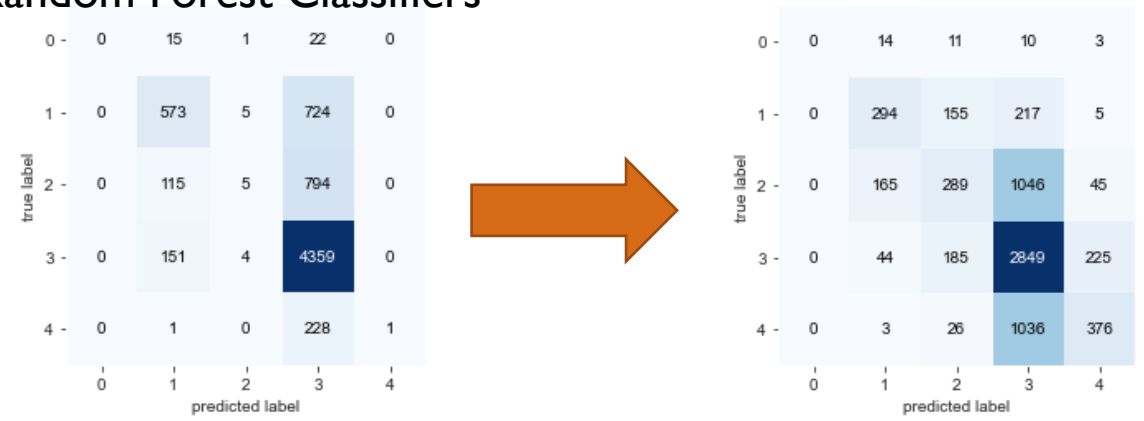


Restaurants should pay more attention to the above factors when they are trying to build reputation and get higher score. The most important factor is review counts followed by close time at weekend and open time at weekend.

ROUND-HALF-UP

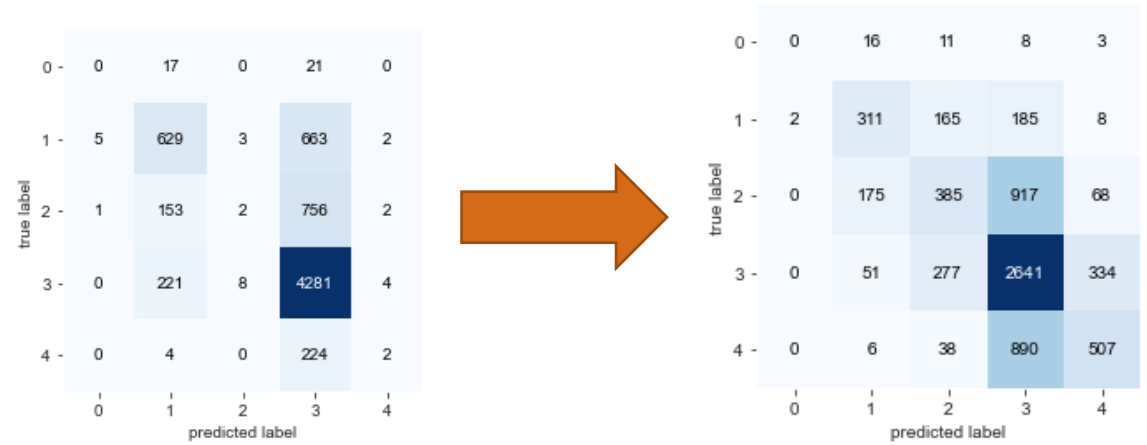
CLASSIFIERS HAVE BETTER PERFORMANCE WITH 5 CLASSES

- Random Forest Classifiers



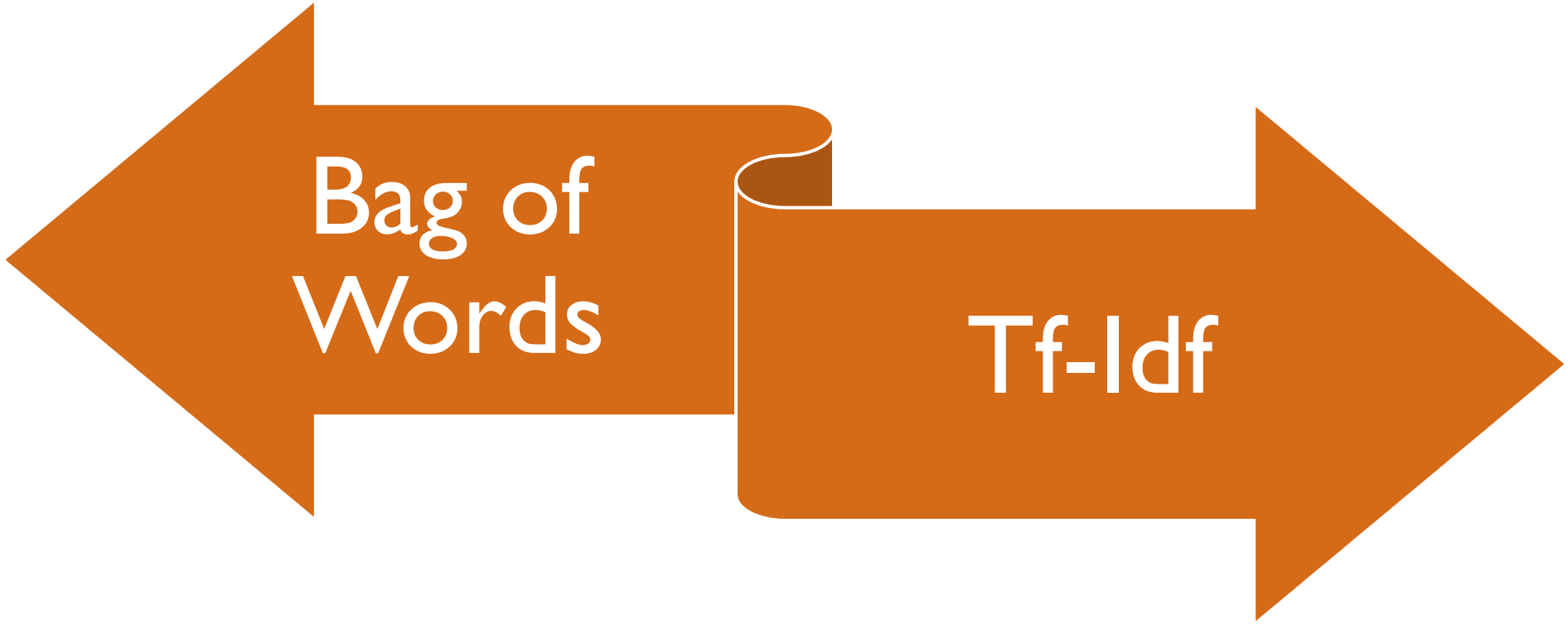

```
best parameter setting found: {'max_depth': 10, 'n_estimators': 50}
best training score found   : 0.545
rfc_gscv test score  : 0.545
```

- Gradient Boosting Classifiers

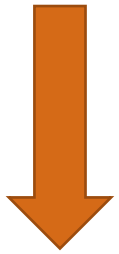



```
best parameter setting found: {'n_estimators': 200}
best training score found   : 0.543
gbc_gscv test score  : 0.549
```

NLP
USING REVIEWS OF RESTAURANTS



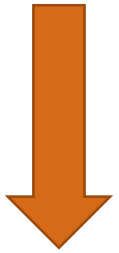
Reviews for each restaurant



Tokenization

Remove Stopwords

Tokens for each restaurant



Lemmatization

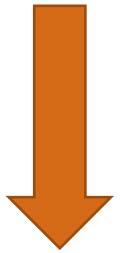
Stemming

Bag of Words for each restaurant

NLP

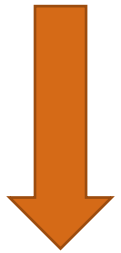
DATA PROCESSING

Bag of words for each restaurant



Calculate Top 5 frequent words for all restaurants

Bag of words for each restaurant



Filter by the set of frequent words (4723)

Bag of words for each restaurant

NLP

- Use Bag of Words for each restaurant filtered by the most frequent used words and calculated the word frequency for each restaurant.
- Use the transformed term frequency matrix for classification

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf= TfidfVectorizer(use_idf=False)
X_tf = tfidf.fit_transform(filter_s)
```

```
X_tf.shape
```

```
(34987, 4723)
```

ROUND-HALF-UP

CLASSIFIERS HAVE BETTER PERFORMANCE WITH 5 CLASSES

- Random Forest Classifiers

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
param_grid = {'n_estimators': [10,50,100],
                'max_depth':[3,5,8,10]}
rfc_gscv = GridSearchCV(estimator=RandomForestClassifier(),
                        param_grid=param_grid,
                        cv=3,
                        n_jobs=-1).fit(X_train,y_train)

print(f'best parameter setting found: {rfc_gscv.best_params_}')
print(f'best training score found   : {rfc_gscv.best_score_.round(3)}')

rfc_gscv_test_score = rfc_gscv.score(X_test,y_test)

print(f'rfc_gscv test score  : {rfc_gscv_test_score.round(3)}')
```

```
best parameter setting found: {'max_depth': 10, 'n_estimators': 100}
best training score found   : 0.688
rfc_gscv test score  : 0.691
```

- Gradient Boosting Classifiers

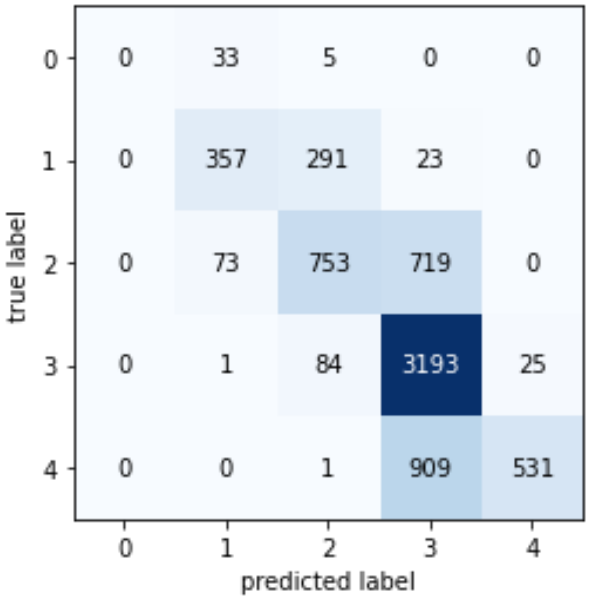
```
from sklearn.ensemble import GradientBoostingClassifier
param_grid = {'n_estimators': [10,50,100,200]}
gbc_gscv = GridSearchCV(estimator=GradientBoostingClassifier(),
                        param_grid=param_grid,
                        cv=3,
                        n_jobs=-1).fit(X_train,y_train)

print(f'best parameter setting found: {gbc_gscv.best_params_}')
print(f'best training score found   : {gbc_gscv.best_score_.round(3)}')

gbc_gscv_test_score = gbc_gscv.score(X_test,y_test)
y_pre_gbc = gbc_gscv.predict(X_test)
print(f'gbc_gscv test score  : {gbc_gscv_test_score.round(3)}')
```

```
best parameter setting found: {'n_estimators': 200}
best training score found   : 0.778
gbc_gscv test score  : 0.789
```

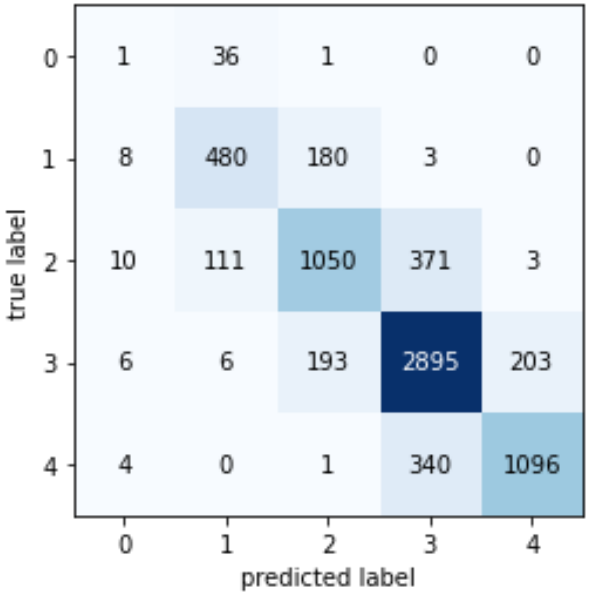
ROUND-HALF-UP

CLASSIFIERS HAVE BETTER PERFORMANCE WITH 5 CLASSES

- Random Forest Classifiers



- Gradient Boosting Classifiers



NLP

- Apply Tf-Idf measurement to all the text reviews for each restaurant. We keep words occur in at least 5 documents and at most 80% of documents, use tokens of at least 2 characters and use only unigrams. So, we create a vector with 109100 columns to each restaurant. And apply:
  - Linear Regression
  - Decision Tree Regressor
  - Decision Tree Classifier
  - Gradient Boosting Classifier
  - Gradient Boosting Regressor

  using rating as response variable, and make pipeline for each.

NLP

Regressor	Accuracy
Linear Regression	0.66
Decision Tree Regressor	0.51
Decision Tree Classifier	0.72
Gradient Boosting Classifier	0.81
Gradient Boosting Regressor	0.81

SENTIMENT ANALYSIS

- Based on the sentiment score, we can also train models to find the relation between it and ratings and predict ratings using sentiment scores.
- The correlation between the two variables are 0.88, so we can build models on them.
- We apply same methods on NLP

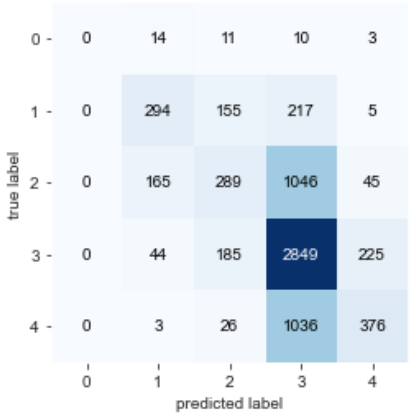
SENTIMENT ANALYSIS

Regressor	Accuracy
Linear Regression	0.78
Decision Tree Regressor	0.79
Decision Tree Classifier	0.81
Gradient Boosting Classifier	0.82
Gradient Boosting Regressor	0.79

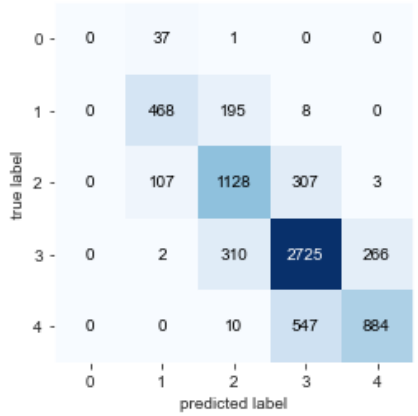
WITH SENTIMENT

CLASSIFIERS HAVE BETTER PERFORMANCE WITH 5 CLASSES

- Random Forest Classifiers

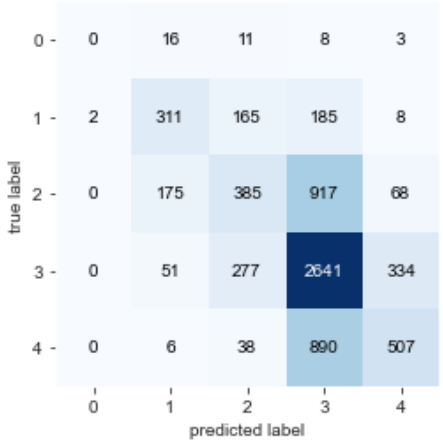


Test score: 0.55

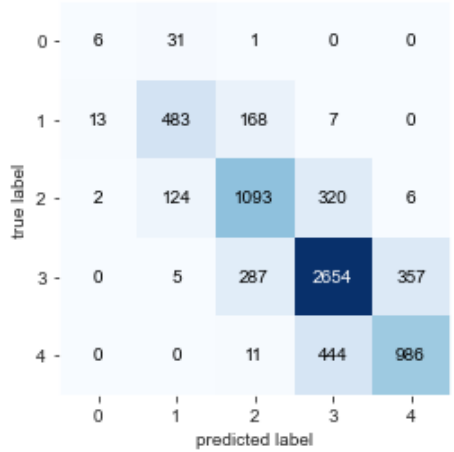


Test score: 0.74

- Gradient Boosting Classifiers



Test score: 0.55



Test score: 0.75

CONCLUSION

- NLP and sentiment analysis works better on predicting the rating of a restaurant.
- But the two methods need some customer reviews. If the restaurant does not get certain amount of the reviews, the owner can adjust the service based on the feature importance from the Gradient Boosting Classifier restaurant attributes to get better ratings.

	factors	importance
0	review_count	0.354094
1	RestaurantsDelivery	0.120427
2	RestaurantsPriceRange2	0.101749
3	open_weekend	0.154940
4	close_weekend	0.215799
5	Service	0.052991

THANKS FOR YOUR LISTENING