Amazon
Review Analysis for
Helpfulness Rating Prediction

- “Is this review helpful?”

Soo Hyeon Kim

Problem?

Great, well balanced sound but with some Bluetooth issues

November 30, 2017

Color: Black | Verified Purchase

Starting with the good. The speaker has great sound with a well balanced mix of clear highs, full mid-tones and surprisingly good bass that provides punch but isn't boomy.

The speaker design is simple yet elegant with its tapered shape, cloth covering and angled top, which houses almost all of the speaker's controls (except for the power button that's on the back bottom right of the speaker).

Now about the not so good. The Bluetooth connection from my phone or tablet or echo dot to the speaker is choppy but only when streaming dialog. For example, when streaming people talking, it sounds like the speaker cuts off the stream when the talking pauses between sentences. This results in the first word or two of the following sentence getting chopped off before the speaker restarts.

The second is a minor nit, not a deal breaker by any means. The "buttons" are (I think) capacitative and not mechanical. When I touch a button rapidly, e.g., to turn up the volume, there seems to be a lag between when my finger touches the button and when the button lights up. So only a few of the finger presses actually register. But again a minor nit that is easily resolved by pressing the buttons more slowly.

If anyone else has experienced this issue can you please let me know if or how you were able to fix the issue?

Overall a great speaker, especially at the Lightening Deal's phenomenal price! I will definitely shop for more Cowin products with confidence.

4 people found this helpful

Helpful | Comment | Report abuse

Five Stars

January 9, 2018

Color: Black | Verified Purchase

Just a perfect little speaker. Works very well with my Echo Dot

4 people found this helpful

Helpful | Comment | Report abuse

Overview

- This project set off with one clear goal in mind:
  “How to predict the helpfulness of reviews”
- **Data Wrangling:**
  From data obtaining to feature selection
- **Exploratory Data Analysis:** Stories from data
- **Inferential Statistical Analysis:**
  Statistical investigation of relationship between variables
- **Machine Learning:**
  Searching for optimal model and predicting helpfulness

01 Data Wrangling

- Data was mined at the repository of Dr. McAuley with his permission (http://jmcauley.ucsd.edu/data/amazon/index.html)


```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the
piano.  He is having a wonderful time playing these old hymns.
The music  is at times hard to read because we think the book
was published for singing from more than playing from.  Great
purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```


(Raw review data example)

01 Data Wrangling

- Filtering Data: choose reviews that have helpful denominator between 10-10K

- Cleaning Data

```
## Investigate missing value percentage
df.isnull().sum() / df.shape[0] * 100
```

```
reviewerID           0.000000
asin                 0.000000
reviewerName         0.319143
helpful              0.000000
reviewText           0.000000
overall              0.000000
summary              0.000000
unixReviewTime       0.000000
reviewTime           0.000000
salesRank           15.842014
categories           0.576170
title               14.582458
description         15.586838
price               12.043028
brand               75.951619
dtype: float64
```

- Simplifying Categories

```
## See head and tail of counts of categories
cat_ser = df.categories.value_counts()

## show top, bottom 10
print(cat_ser.head(10))
print(cat_ser.tail(10))
```

```
Books                          558400
Movies & TV                    125968
CDs & Vinyl                    111716
Electronics                    102527
Home & Kitchen                  50224
Health & Personal Care          40612
Video Games                     34267
Sports & Outdoors               27100
Tools & Home Improvement        23793
Toys & Games                    20951
Name: categories, dtype: int64
Country                    9
New Age                    8
R&B                        7
Folk                       7
Blues                      5
Wine                       5
Amazon Coins               3
Broadway & Vocalists       3
Rap & Hip-Hop              2
Children's Music           1
Name: categories, dtype: int64
```

Data Wrangling 01

- Transforming and scaling helpfulness: [13, 20] — “3”

$$helpful\_num : helpful\_den = rating : 5$$

$$\therefore rating = 5 \times \frac{helpful\_num}{helpful\_den}$$

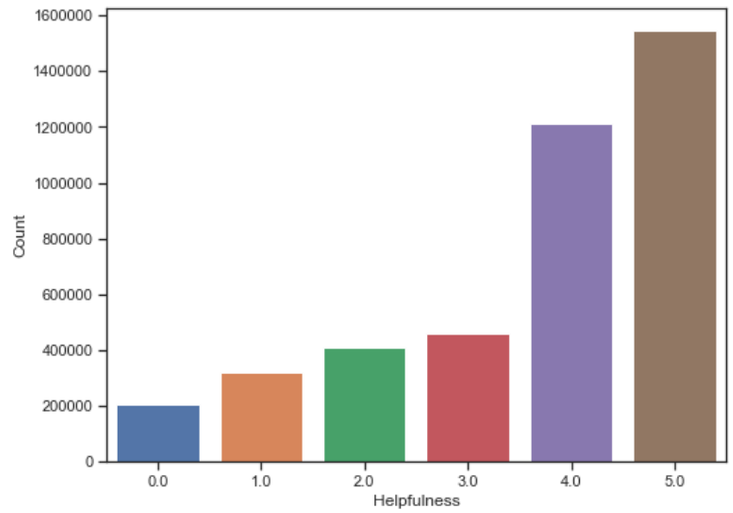
- Combining summary and reviewText

```
## combining 'summary' and 'reviewText'
df['review'] = df[['summary', 'reviewText']].apply(lambda x: '. '.join(x), axis=1)
```

```
## drop 'summary' and 'reviewText' columns
df = df.drop(columns=['summary', 'reviewText'])
```

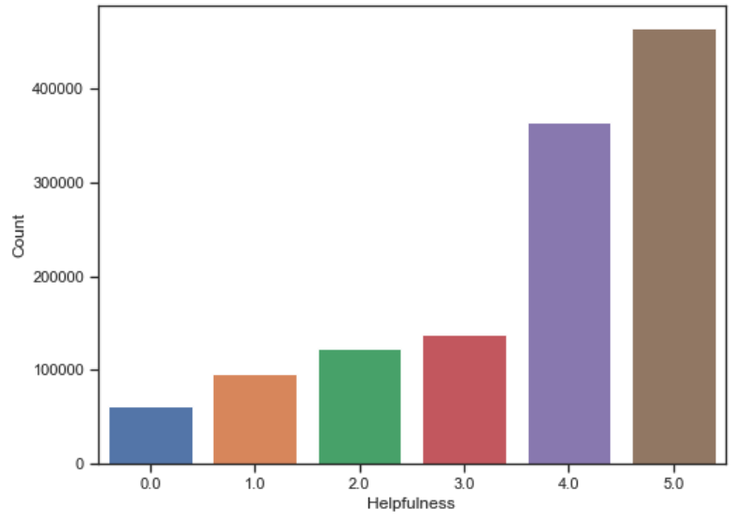
01 Data Wrangling

- Stratified Sampling:



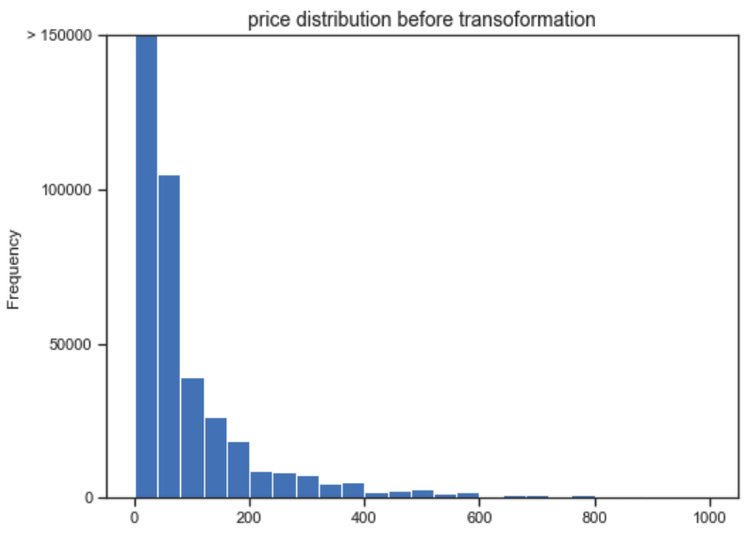
4.15M

30%



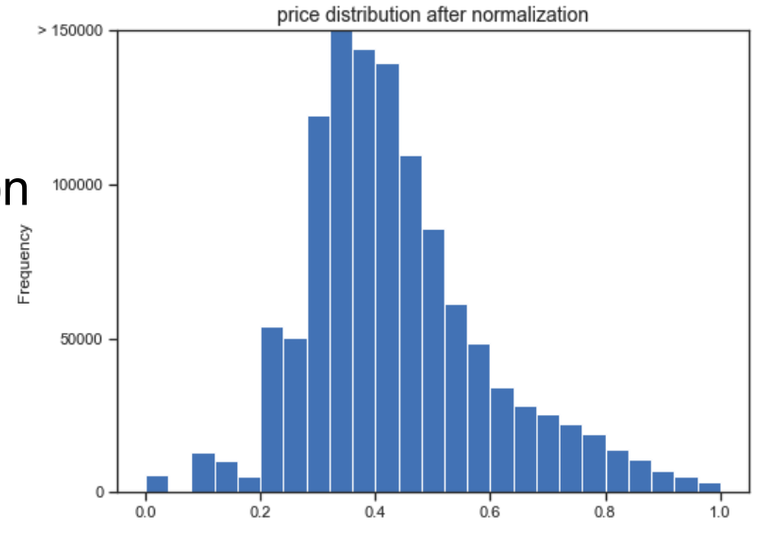
1.24M

- Appeasing skewness: Price feature



Log transformation

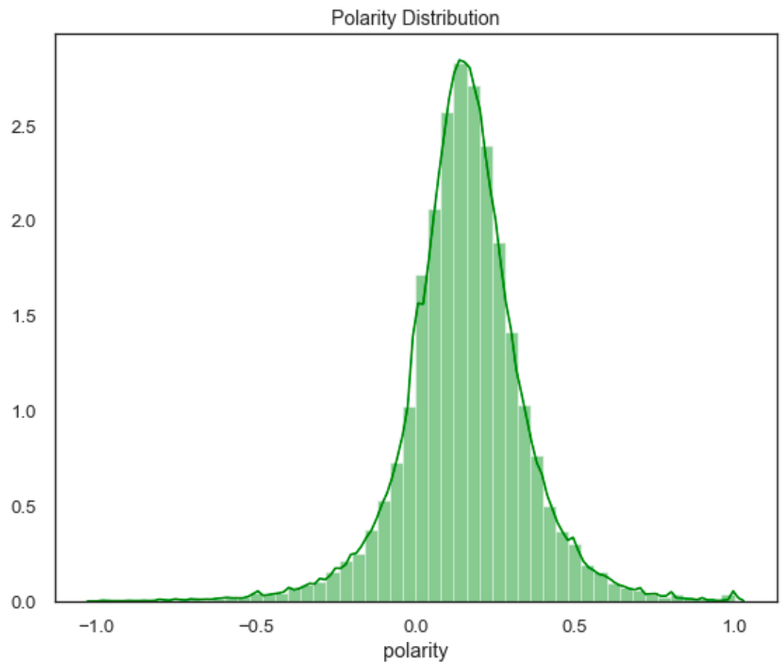
Normalization

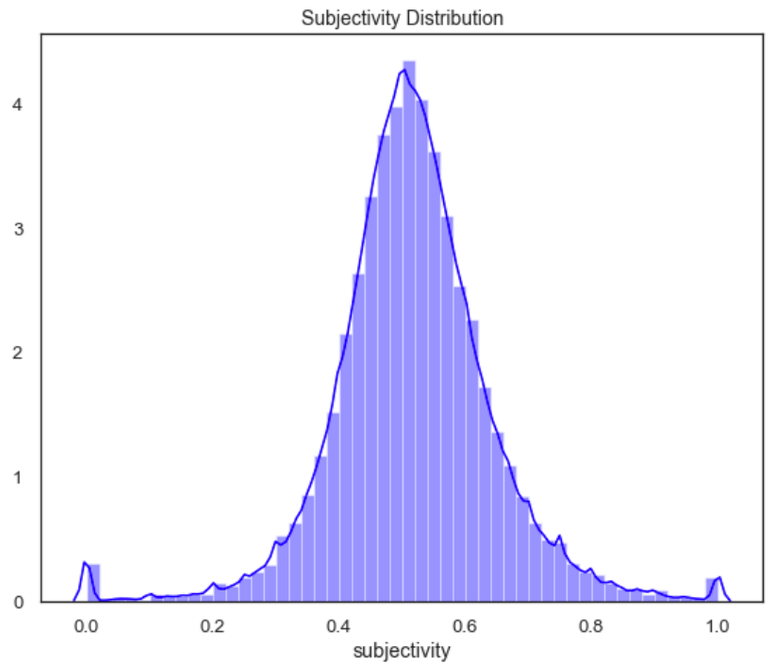


Feature Engineering

01 Data Wrangling

1) Sentiment Features: polarity & subjectivity





2) Textual features:

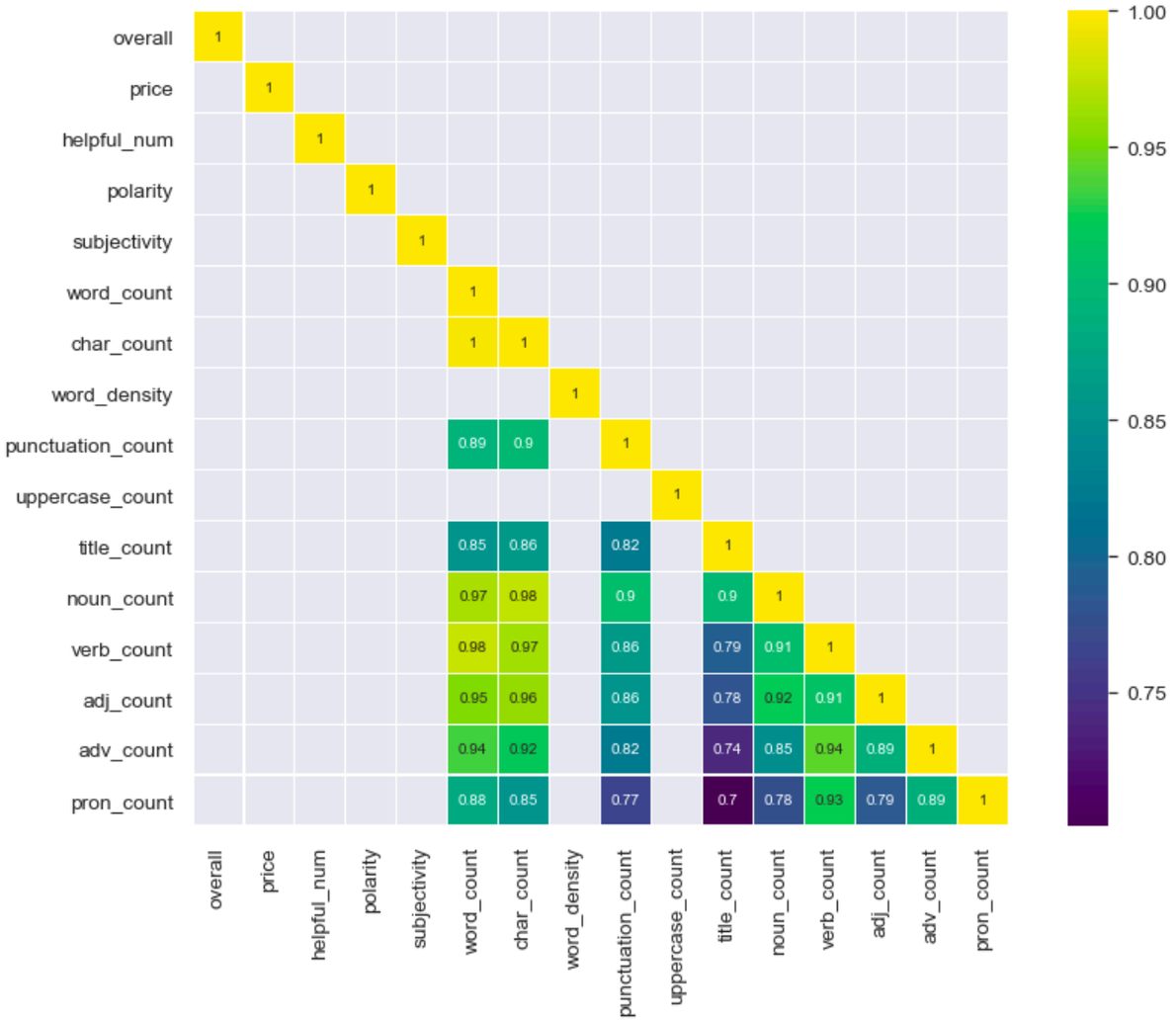
Word Count, Character Count, Average Word Density, Punctuation Count, Uppercase Count, Title Word Count, Noun Count, Verb Count, Adjective Count, Adverb Count, Pronoun Count

3) Tf-idf features: 500 vectors that has high tf-idf score.

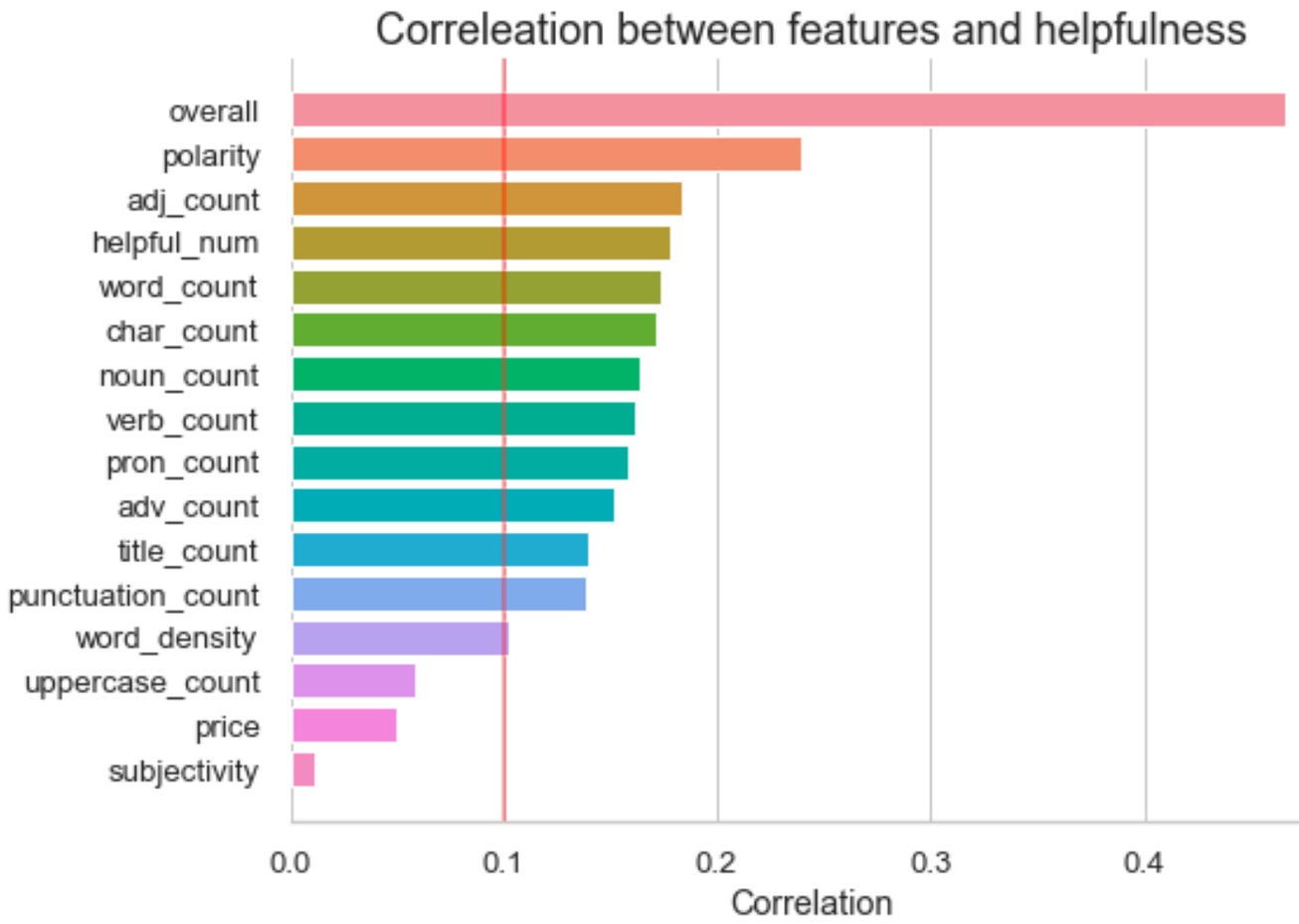
Feature Selection

01 Data Wrangling

- Collinearity and Correlation

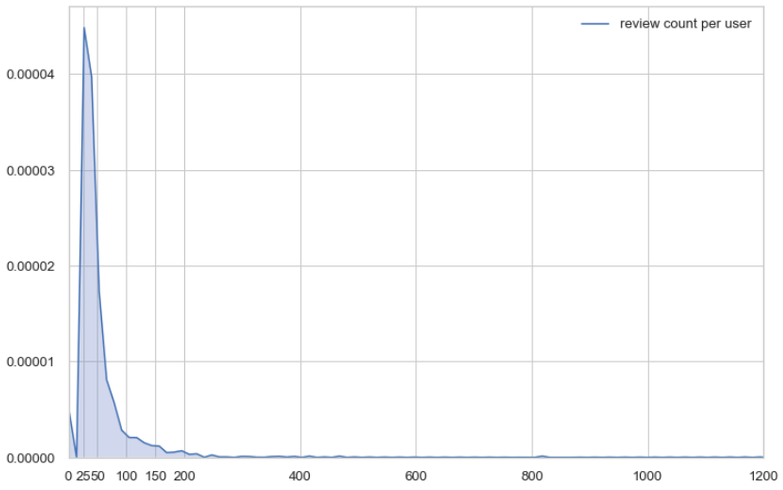


(Strong correlation among features)

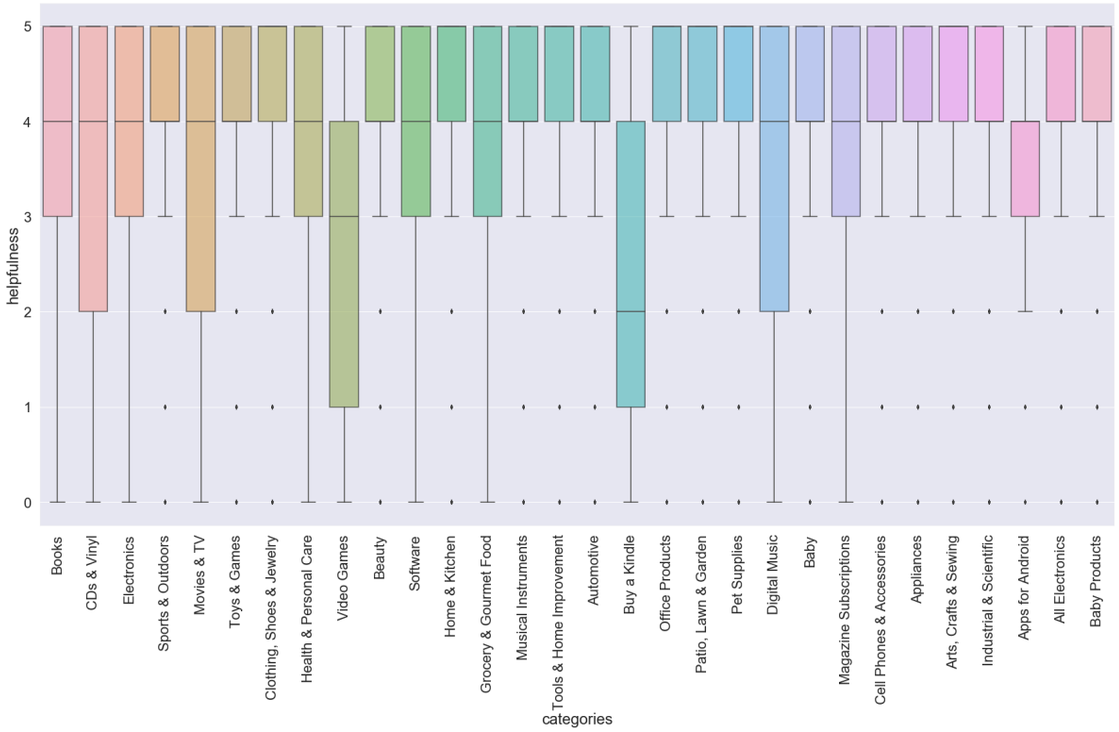


(Correlation to helpfulness)

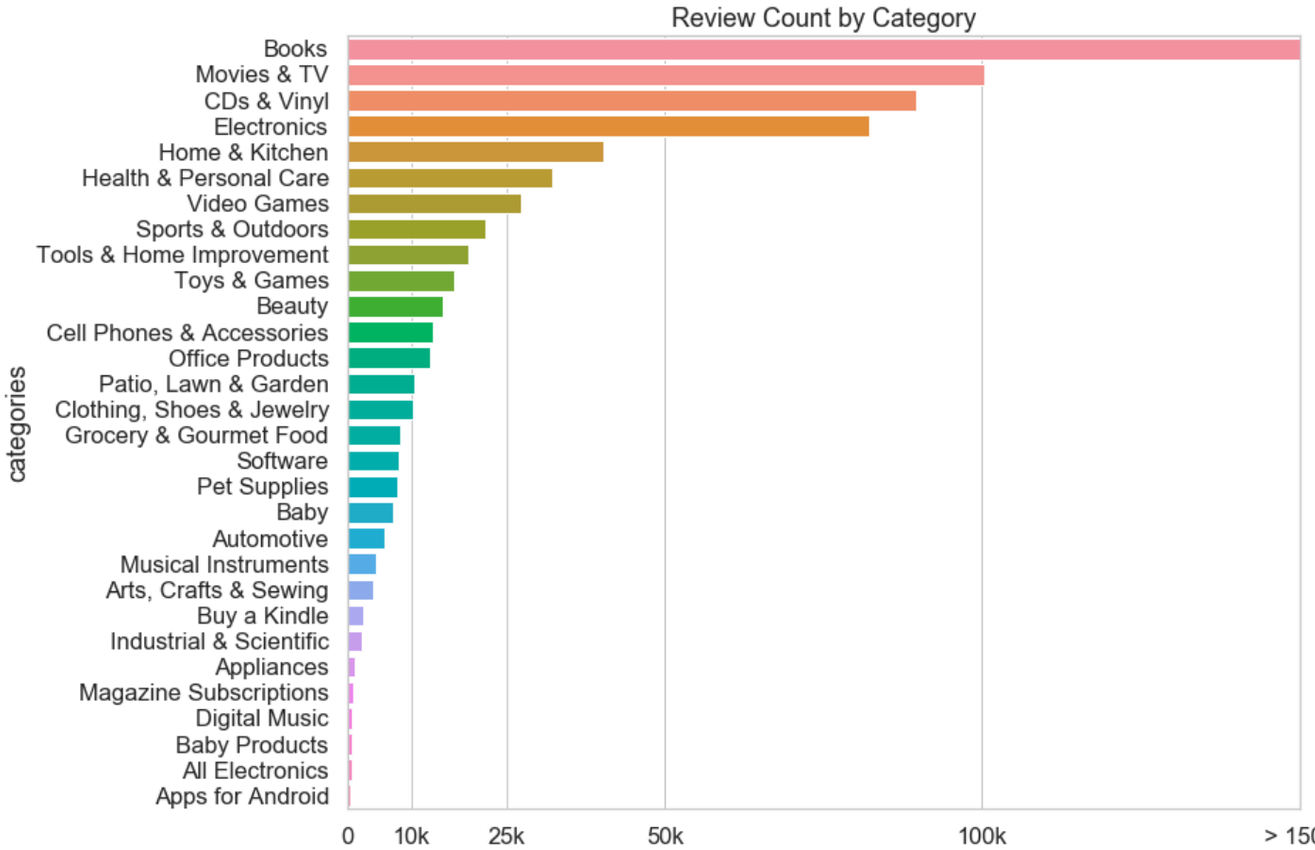
02 Exploratory Data Analysis



(Dist. of the # of reviews per user)



(Helpfulness dist. By categories)



(# of reviews per categories)

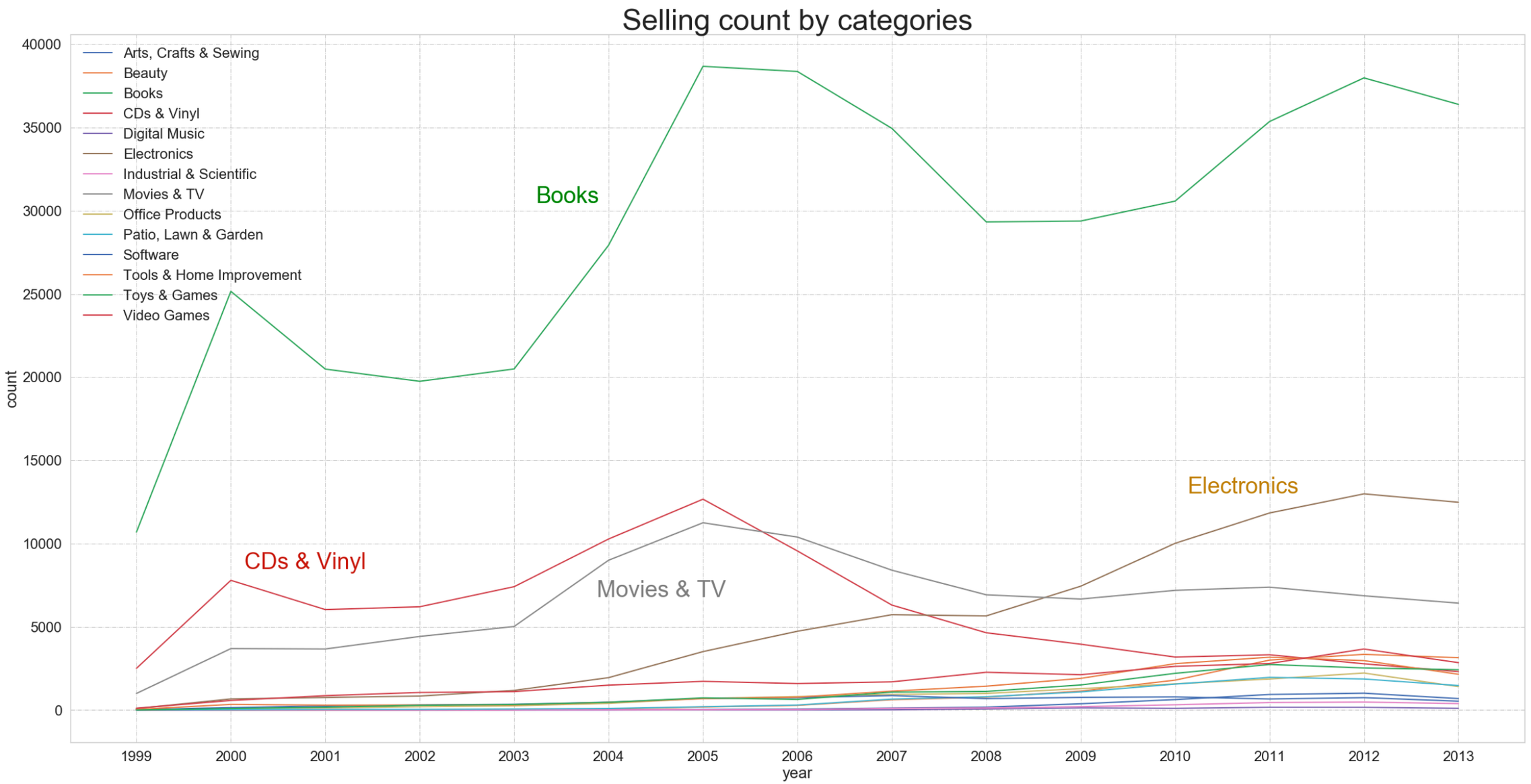
02 Exploratory Data Analysis

Common 500 words in 90's

Common 500 words in '14

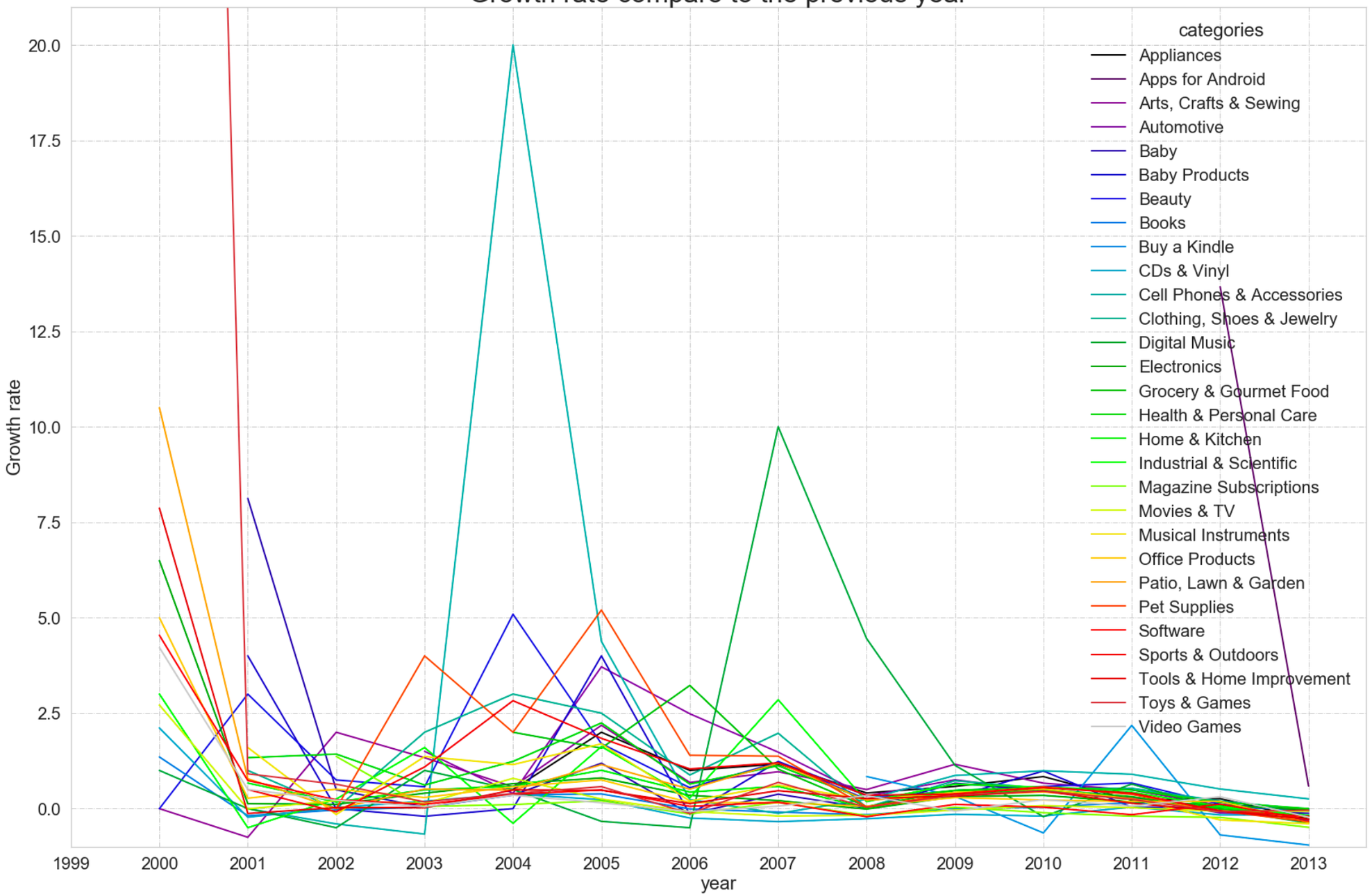
(Common 500 words in the late 1990s and 2014)

02 Exploratory Data Analysis



(Selling count by categories over the time)

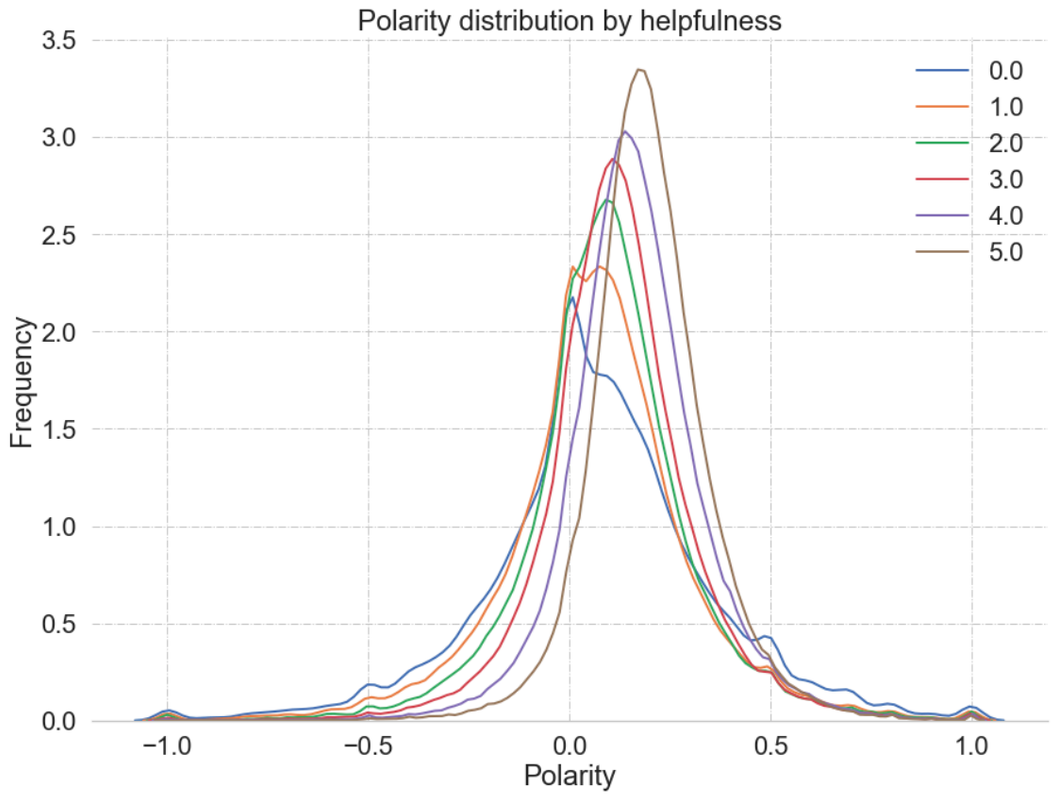
02 Exploratory Data Analysis



(Growth rate compared to the previous year)

03 Inferential Statistical Analysis

- Polarity & Helpfulness
  - Ho: Polarity and helpfulness rating are independent
  - Ha: Polarity and helpfulness rating are NOT independent



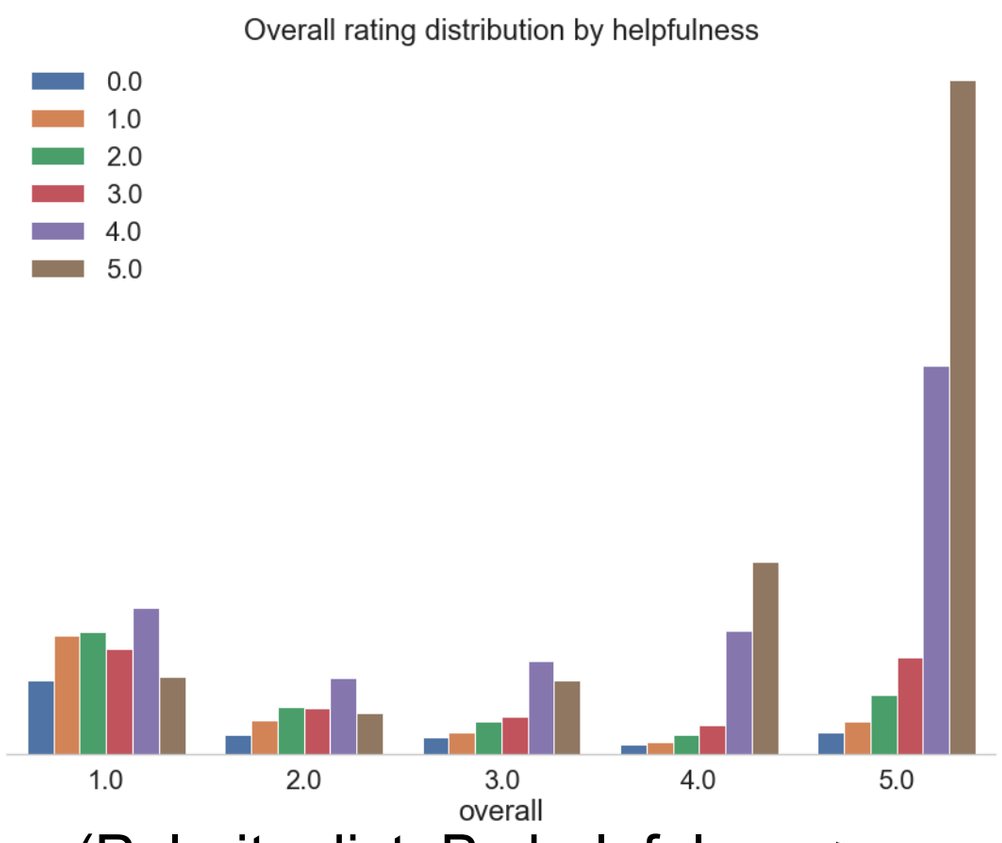
(Polarity dist. By helpfulness)

helpfulness	0.0	1.0	2.0	3.0	4.0	5.0	total
pol_bin							
[-0.5, 0)	15573	23516	25084	22045	34907	23550	144675
[-1, -0.5)	1256	1202	795	493	586	357	4689
[0, 0.5)	29093	48699	68506	84319	244011	334403	809031
[0.5, 1]	3496	3085	3306	3385	10128	12241	35641
total	49418	76502	97691	110242	289632	370551	994036

(Contingency table of polarity and helpfulness)

03 Inferential Statistical Analysis

- Overall & Helpfulness
  - Ho: Overall rating and helpfulness rating are independent
  - Ha: Overall rating and helpfulness rating are NOT independent



(Polarity dist. By helpfulness>

helpfulness overall	0.0	1.0	2.0	3.0	4.0	5.0	total
1.0	25946	41436	42973	36879	51139	27245	225618
2.0	6910	11916	16469	16185	26619	14287	92386
3.0	6035	7759	11179	13226	32715	25629	96543
4.0	3141	4015	6531	10294	43082	67361	134424
5.0	7386	11376	20539	33658	136077	236029	445065
total	49418	76502	97691	110242	289632	370551	994036

(Contingency table of polarity and helpfulness)

04 Machine Learning

Final Touch on Data

- Dimension reduction on tf-idf features with truncatedSVD

```
## Dimension reduction to 300
svd = TruncatedSVD(n_components=300, n_iter=7, random_state=7979)
svd.fit(V_train)

print("Reduced 300-feature data holds {:.2f} % variance of the original data"

## transform arrays
V_train_new = svd.transform(V_train)
V_test_new = svd.transform(V_test)

## change to data frames
columns = ['pc'+str(i) for i in range(1, 300+1)]
V_train_new = pd.DataFrame(V_train_new, columns=columns)
V_test_new = pd.DataFrame(V_test_new, columns=columns)

## dimension check
V_train_new.shape, V_test_new.shape
```

```
Reduced 300-feature data holds 71.40 % variance of the original data
```

- Standardizing all features

	helpful_num	polarity	word_count	word_density	cat_Appliances	cat_Apps for Android	cat_Arts, Crafts & Sewing	cat_Automotive	cat_Baby	cat_Bab Product
mean	5.178760e-16	-2.166002e-16	-5.021578e-17	-3.793256e-15	-1.448696e-14	8.757220e-15	1.027800e-14	7.586651e-15	-5.933334e-15	-1.120757e 1
std	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+0
...	pc291	pc292	pc293	pc294	pc295	pc296	pc297	pc298	pc299	pc300
...	3.026219e-16	5.180992e-16	4.679954e-17	-2.088562e-17	2.880885e-16	-1.476404e-17	7.776454e-16	-2.265700e-17	3.130718e-16	-1.905817e-16
...	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00

04 Machine Learning

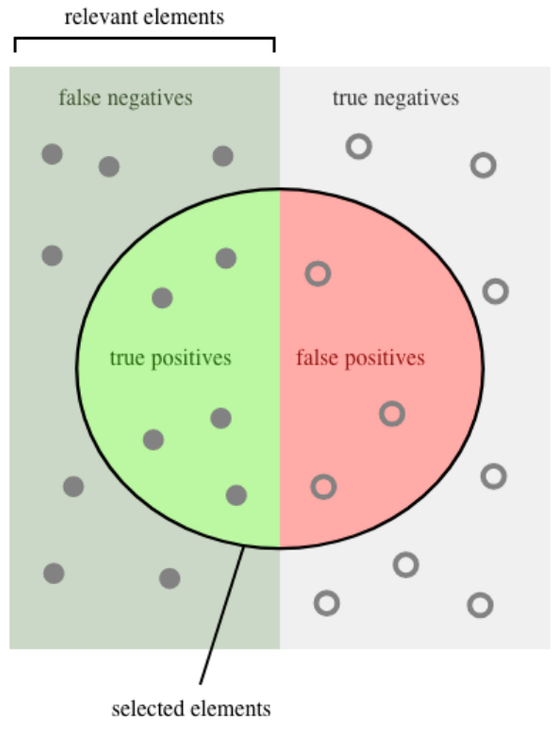
ML frameworks

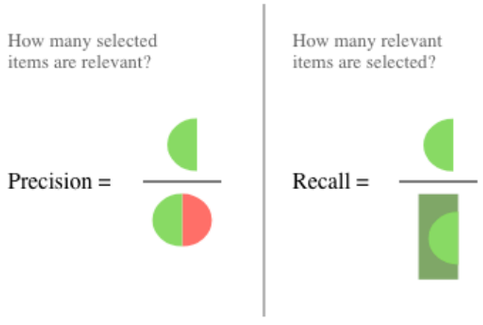
- Logistic regression and gradient boosting frameworks as candidate models
- Evaluated with AP (average precision)

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

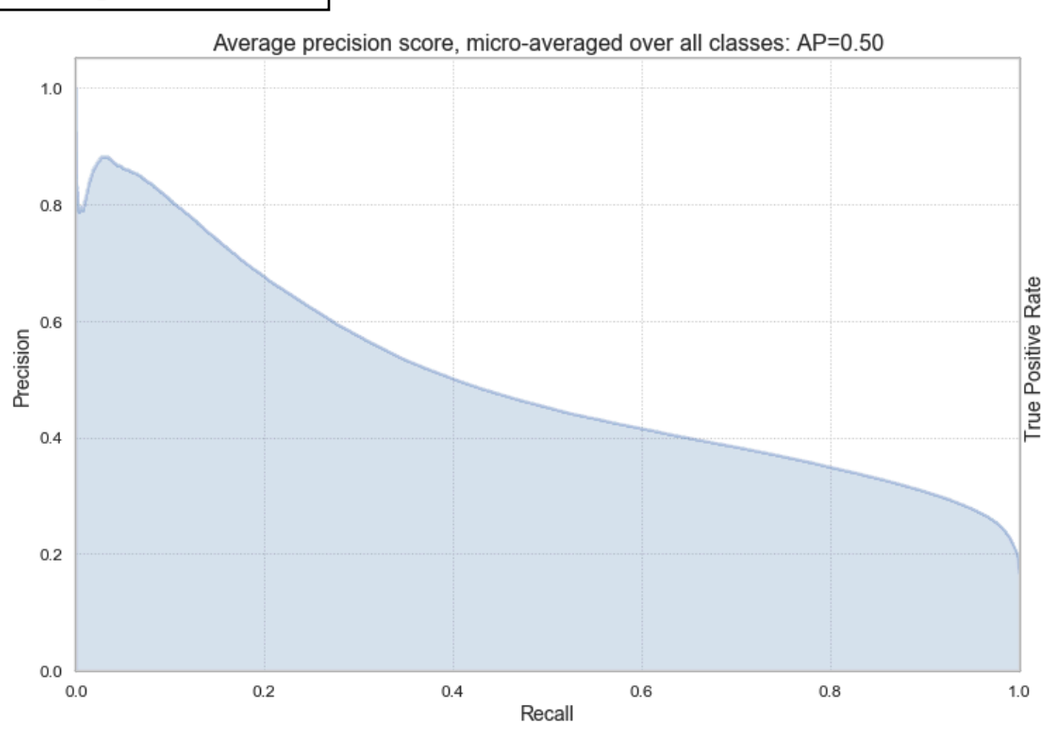
$$AP = \sum_{n} (R_n - R_{n-1}) P_n$$

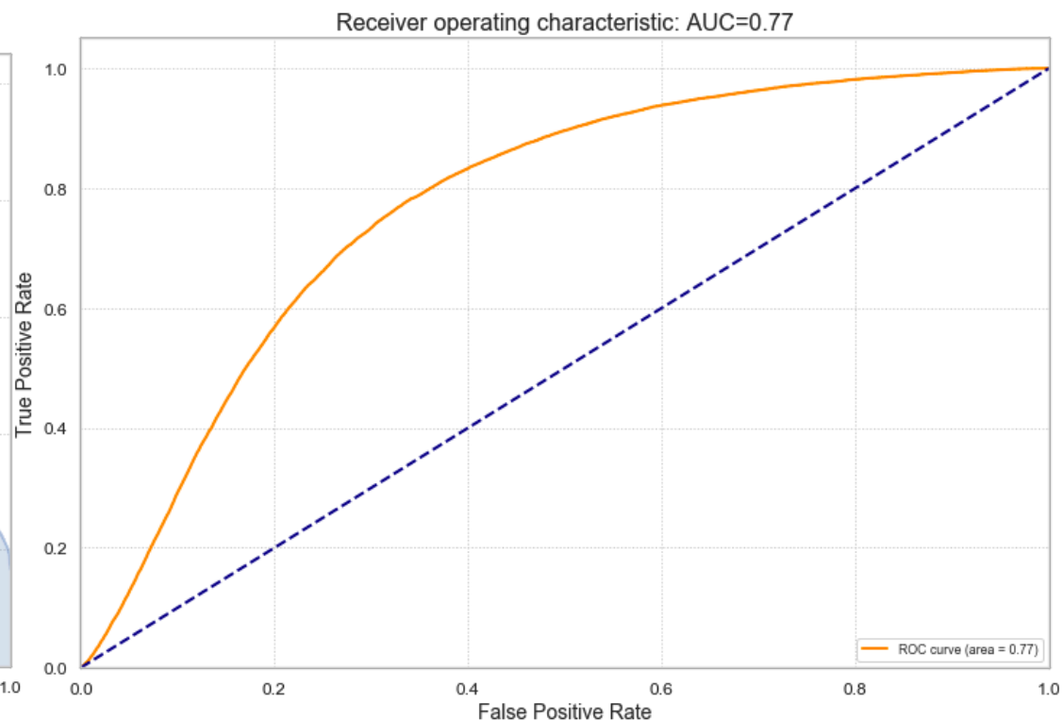


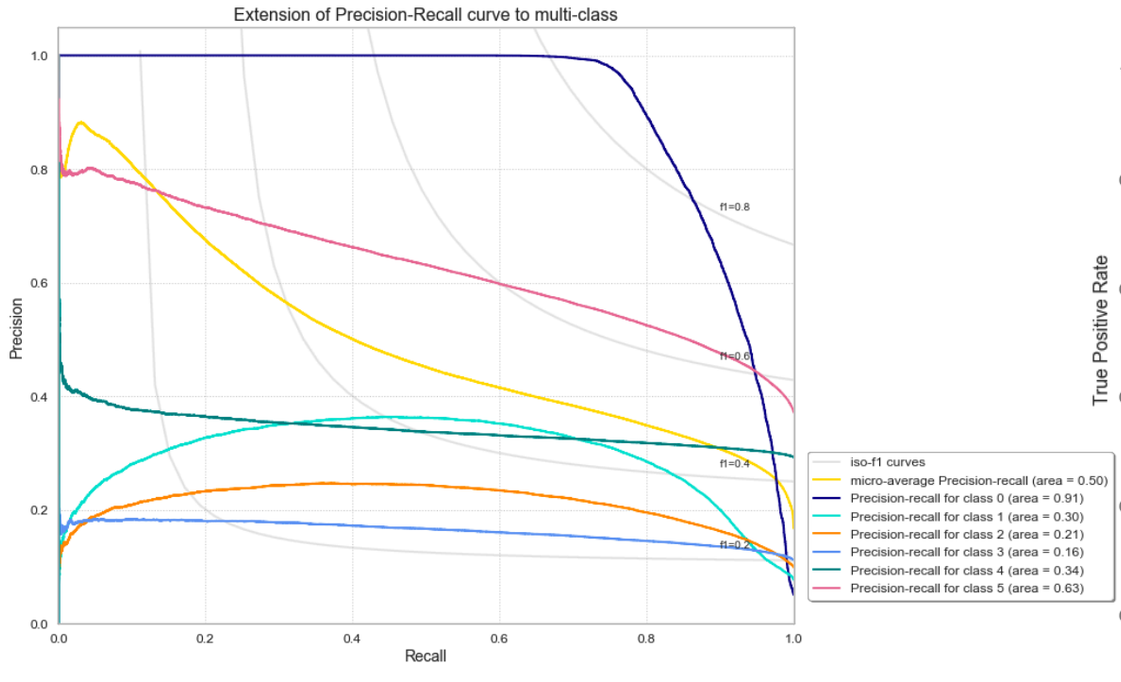


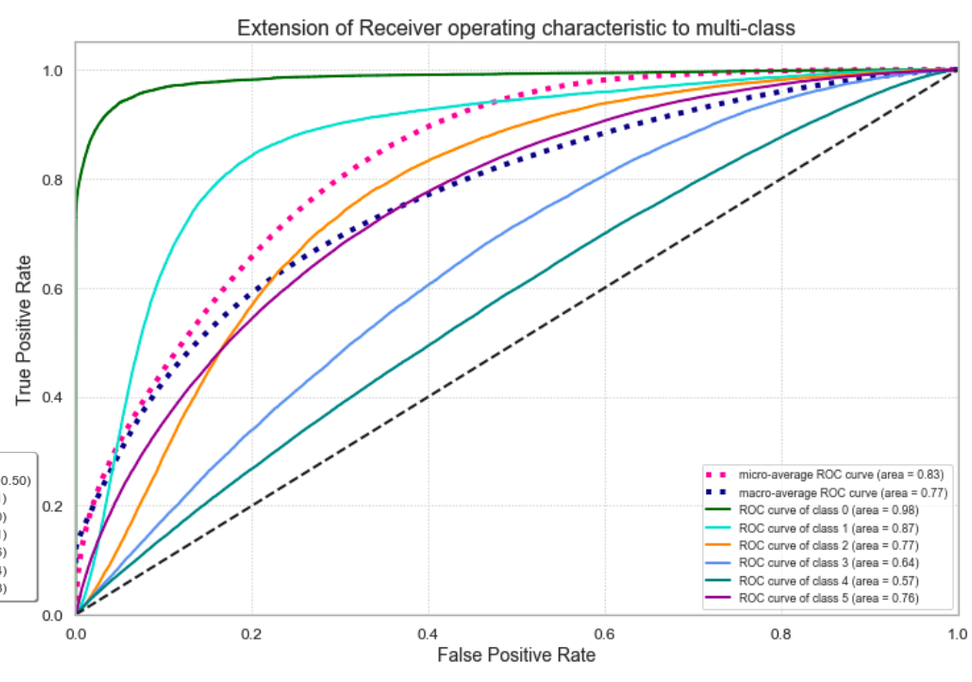
04 Machine Learning

Logistic Regression





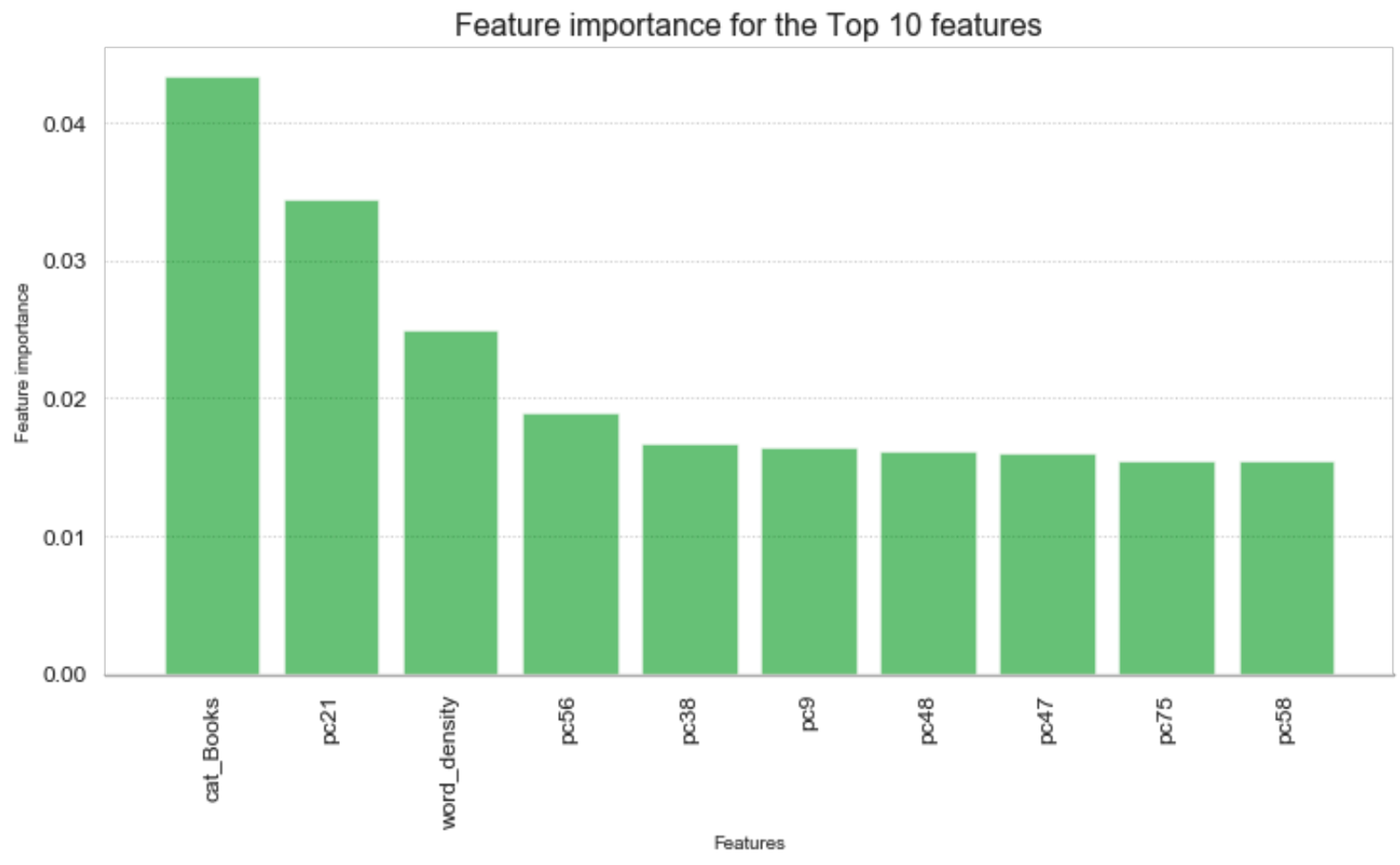




04 Machine Learning

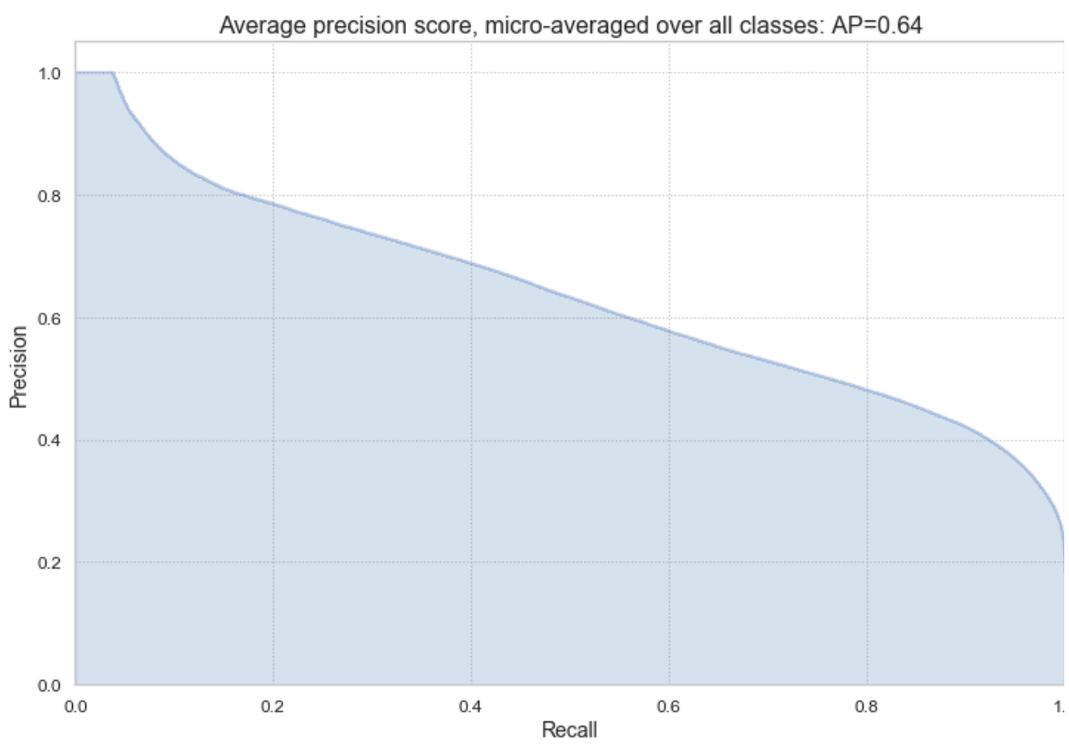
Logistic Regression

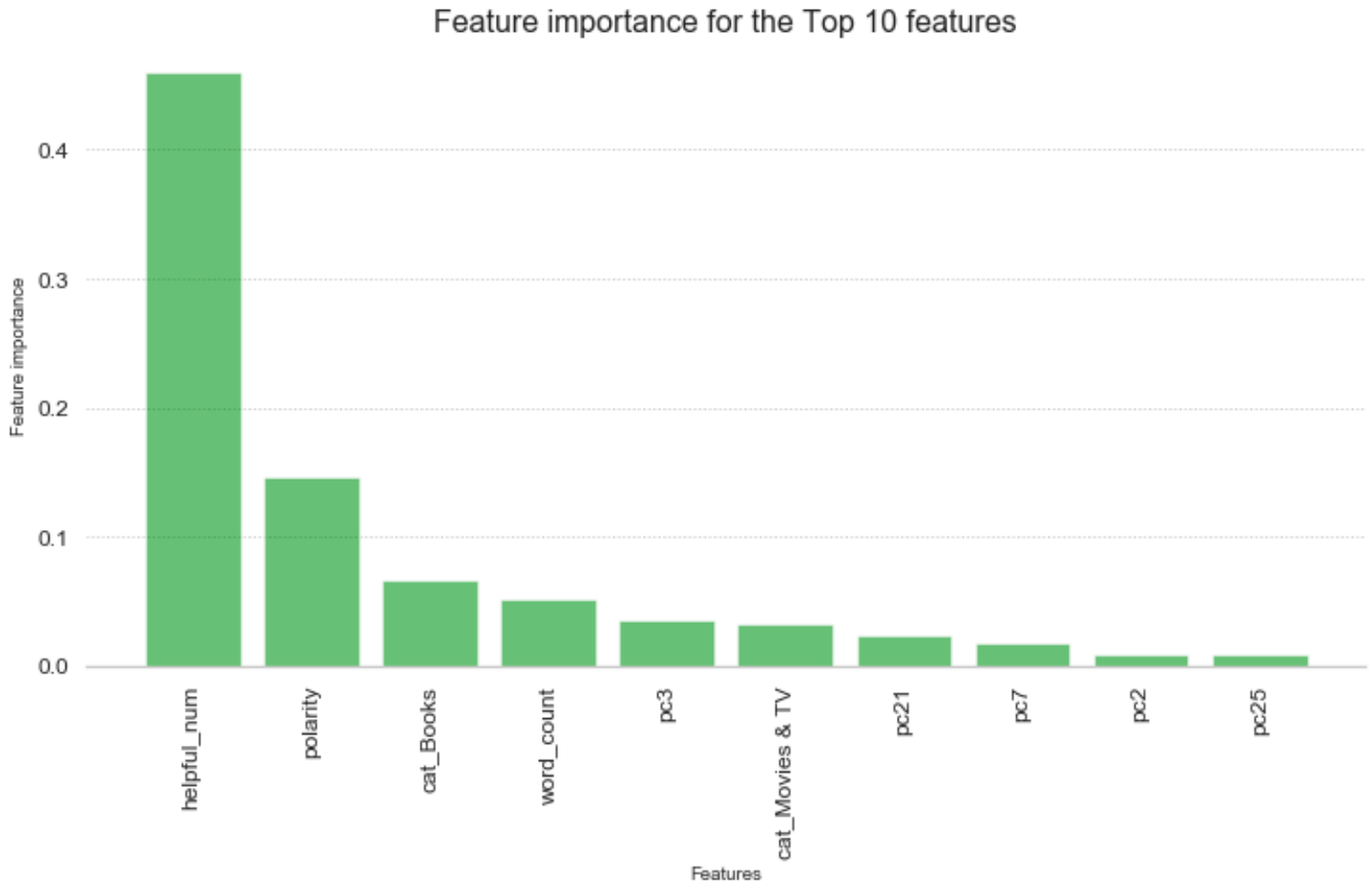
```
Best logistic regression estimator:
OneVsRestClassifier(estimator=LogisticRegression(C=10, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=1004, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False),
          n_jobs=1)
```



04 Machine Learning

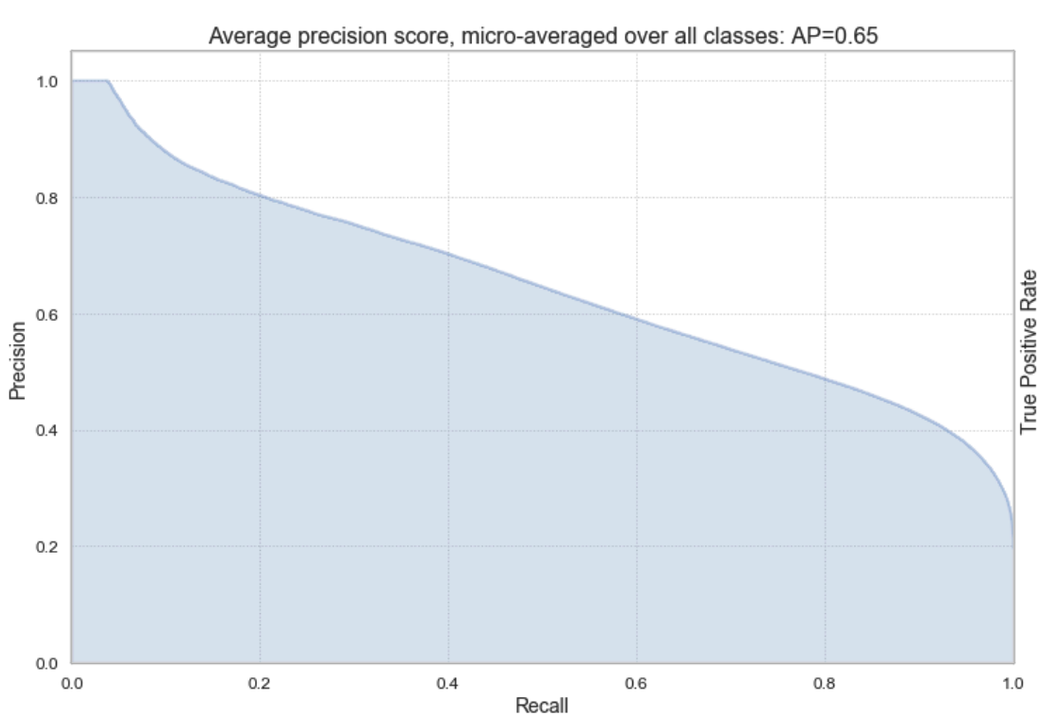
XGB classifier - default

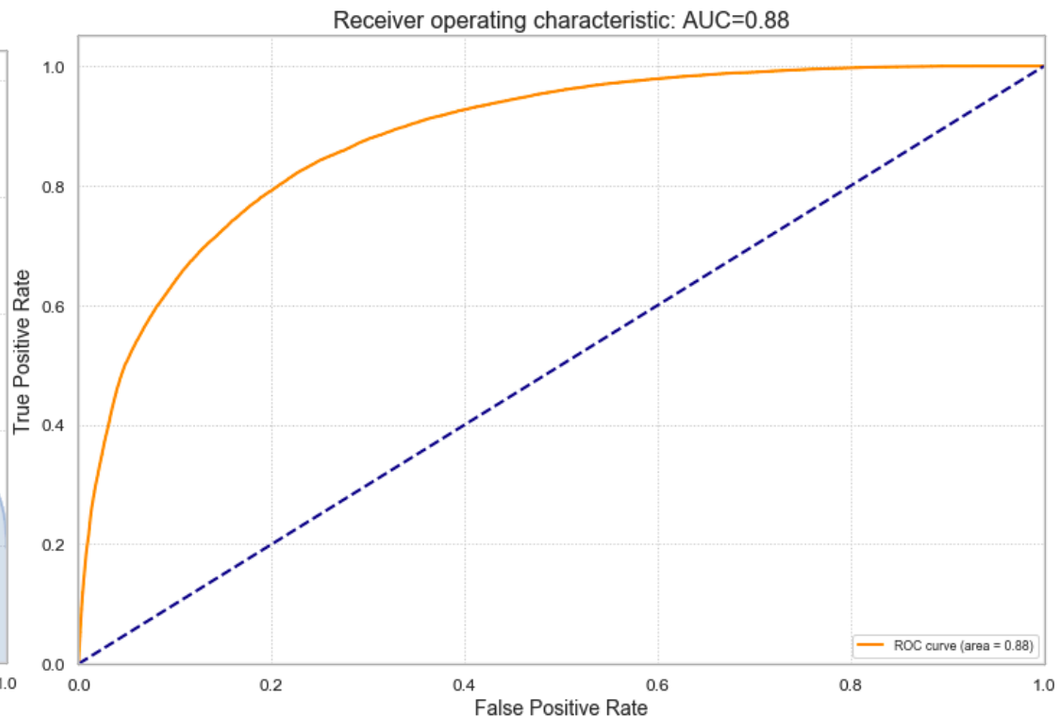


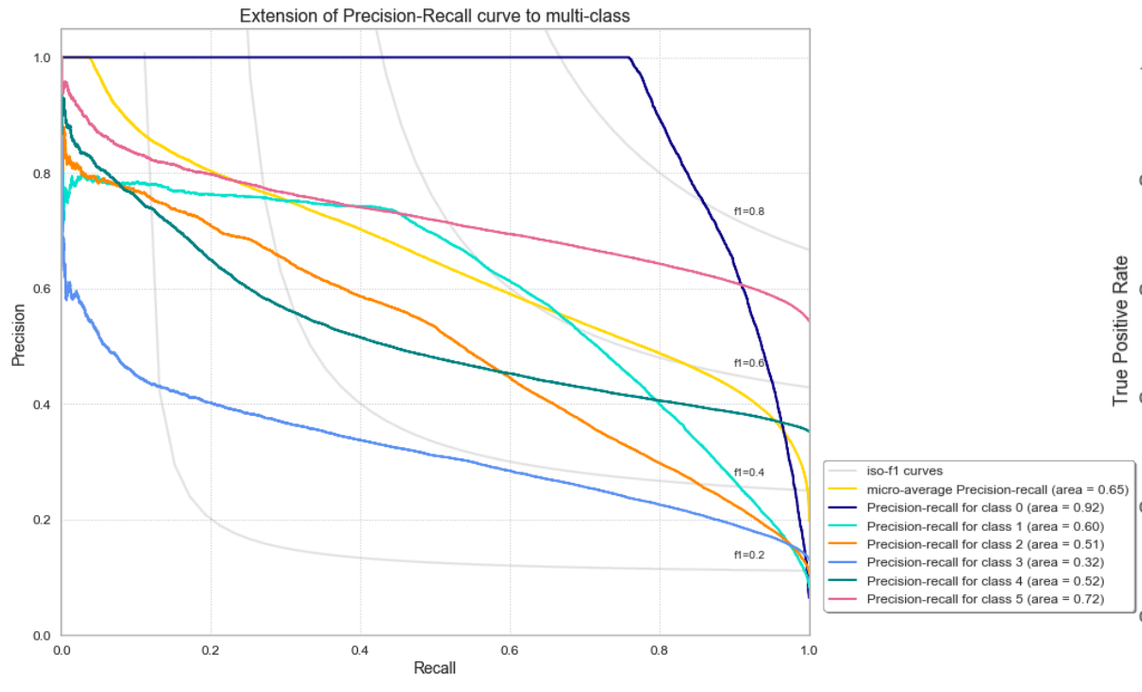


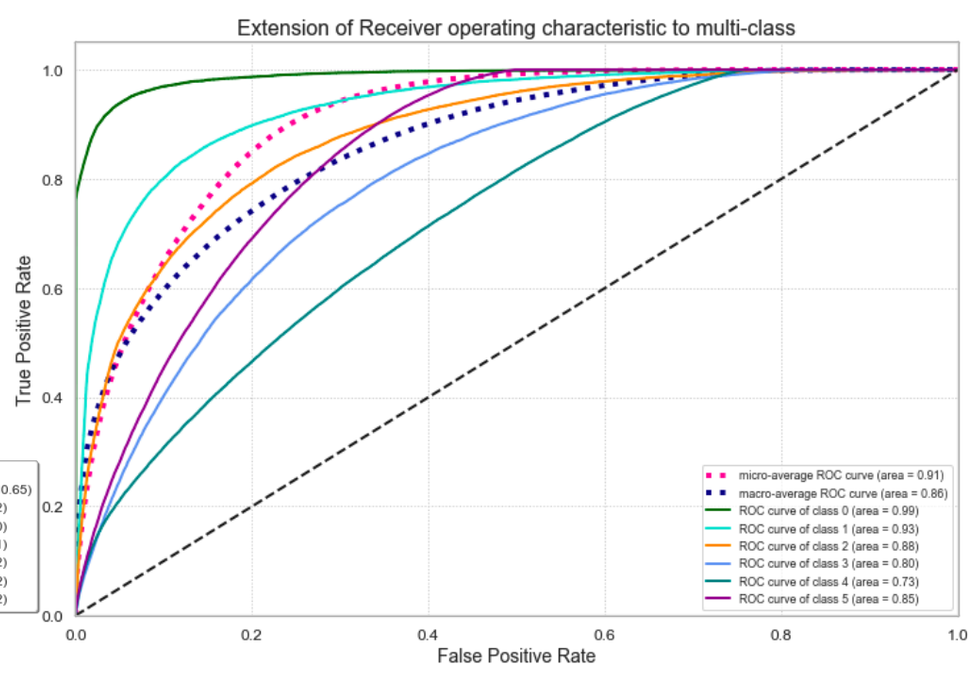
04 Machine Learning

XGB classifier







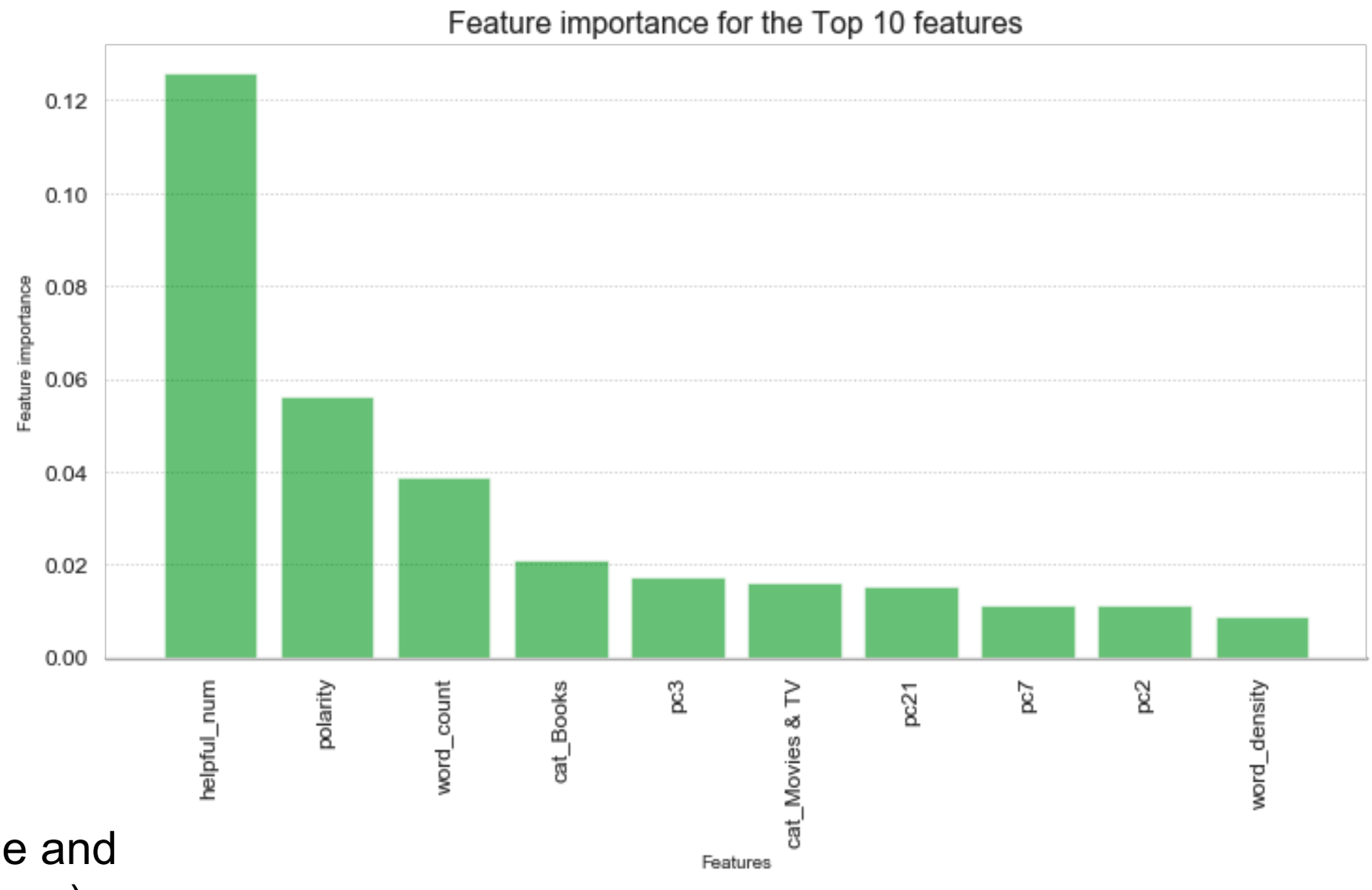


04 Machine Learning

XGB classifier

```
Best XGBclassifier parameters:
{'max_depth': 5, 'learning_rate': 0.2, 'min_child_weight': 1, 'gamma': 5, 'subsample': 0.8, 'colsample_bytree': 1}
```

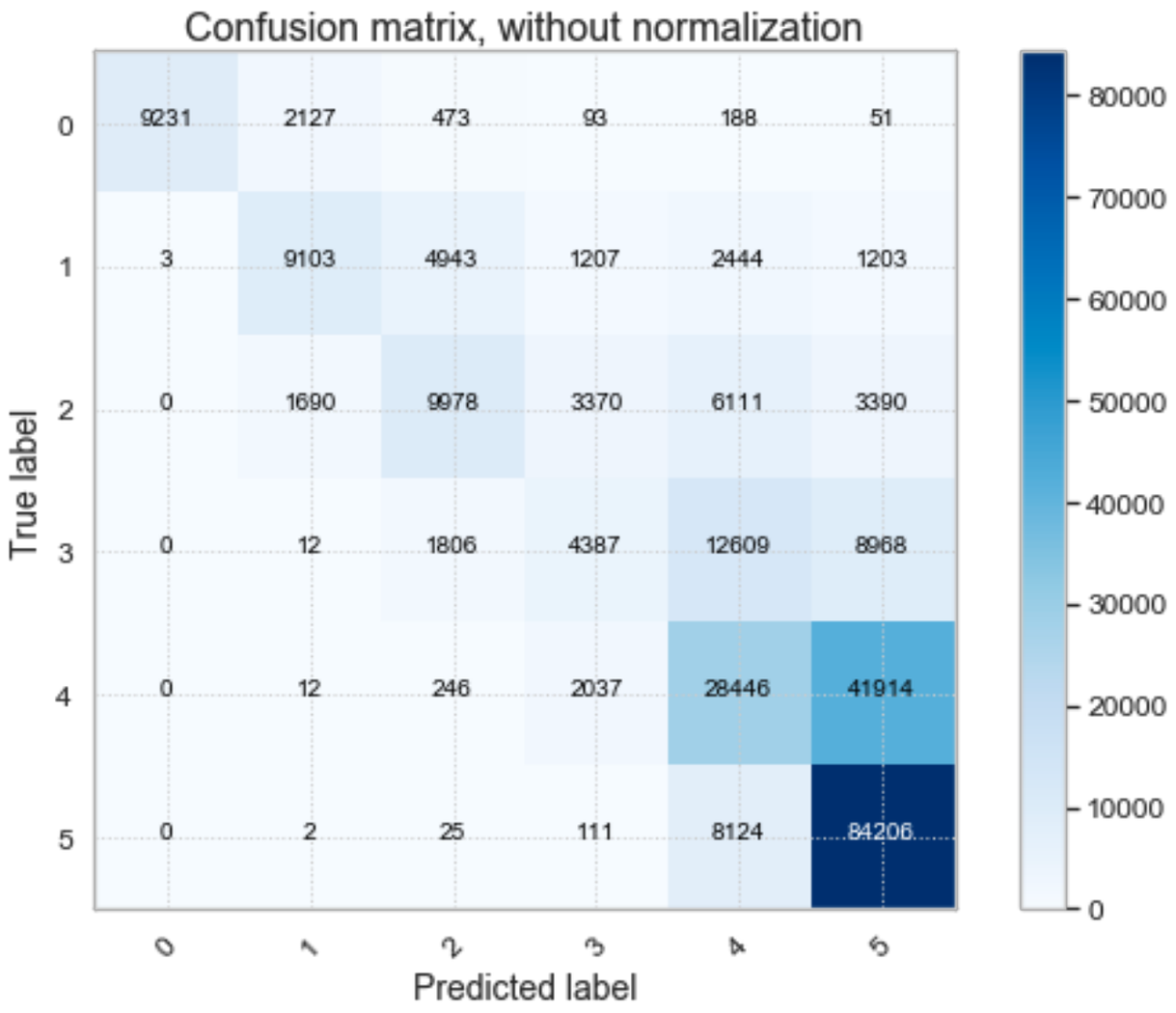
	y_test	y_hat
112762	4	4
232791	5	5
127097	3	3
146880	5	5
101245	5	5
110583	2	2
149336	3	4
245482	4	5
49712	4	5
71690	5	4

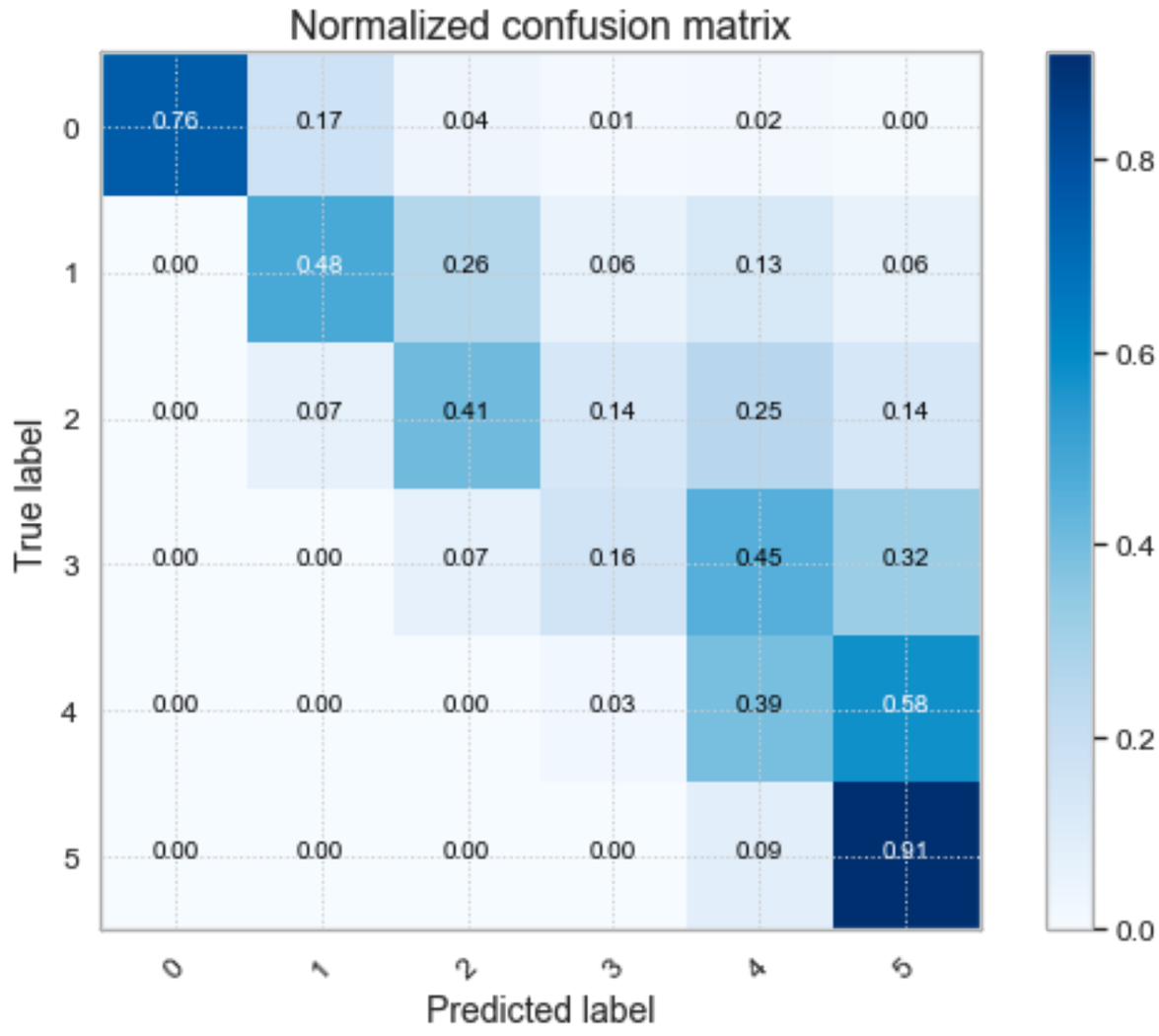


(Ten randomly chosen true and predicted helpfulness ratings)

XGB classifier

04 Machine Learning





Summary

- This project set off with one clear goal in mind:
  “How to predict the helpfulness of reviews”
- **Data Wrangling:**
  From data obtaining to feature selection
- **Exploratory Data Analysis:** Stories from data
- **Inferential Statistical Analysis:**
  Statistical investigation of relationship between variables
- **Machine Learning:**
  Searching for optimal model and predicting helpfulness

Future work

- Adding more data onto middle classes
  (helpfulness rating 0,1,2,3)

- SVD Reconstruction to better understand important features

- Regression model instead of classification
  (i.e., transform helpfulness into continuous values)

Thank you