Text Mining

on

Yelp Reviews

PROJECT OVERVIEW

The main objective is to perform sentiment analysis and create a model for insights.

Steps to achieve this objective :

- Data Description
- Data Loading
- Data Pre-processing
- Data exploration
- Identifying Model features
- Sentiment Analysis
- Correlation between Model Features
- Interpreting the results
- Conclusion

Data Description

- 10,000 Yelp reviews with 10 columns from Kaggle.
- Text data: Text of review
- Other attributes: Stars, Cool, Useful, Funny
- Stars is a categorical variable with the number of stars the reviewer gave, from 1 to 5
- Cool, Useful, and Funny are numerical variables with the number of votes the review got from other users in the 3 categories

	A	B	C	D	E	F	G	H	I	J
1	business_	date	review_id	stars	text	type	user_id	cool	useful	funny
2	9yKzy9PA	1/26/2011	fWKvX83p0-ka4JS3dc6E5A	5	My wife took me here on my birthday for breakfast and	review	rLtl8ZkDX5	2	5	0
3	ZRJwVLyz	7/27/2011	IjZ33sJrzXqU-0X6U8NwyA	5	I have no idea why some people give bad reviews about	review	0a2KyEL0c	0	0	0
4	6oRAC4uy	6/14/2012	IESLBzqUCLdSzSqm0eCSxQ	4	love the gyro plate. Rice is so good and I also dig their ca	review	0hT2KtfLic	0	1	0
5	_1QQZuf4	5/27/2010	G-WvGaISbqqaMHlNnByodA	5	Rosie, Dakota, and I LOVE Chaparral Dog Park!!! It's very	review	uZetl9T0N	1	2	0
6	6ozycU1R	1/5/2012	1uJFq2r5QfJG_6ExMRCaGw	5	General Manager Scott Petello is a good egg!!! Not to go	review	vYmM4KT	0	0	0
7	#NAME?	12/13/2007	m2CKSsepBCoRYWxiRUsxAg	4	Quiessence is, simply put, beautiful. Full windows and	review	sqYN3lNg	4	3	1
8	zp713qNh	2/12/2010	riFQ3vxNpP4rWLk_CSri2A	5	Drop what you're doing and drive here. After I ate here I	review	wFweIWh	7	7	4
9	hW0Ne_H	7/12/2012	JL7GXJ9u4YMx7Rzs05NfiQ	4	Luckily, I didn't have to travel far to make my	review	1ieuYcKS7	0	1	0
10	wNUea3IX	8/17/2012	XtnfnYmnJYi71yIuGsXIUA	4	Definitely come for Happy hour! Prices are amazing, sake	review	Vh_DlizgG	0	0	0
11	nMHhuYa	8/11/2010	jJAIXA46pU1swYyRCdfXtQ	5	Nobuo shows his unique talents with everything on the	review	sUNkXg8-	0	1	0
12	AsSCv0q_	6/16/2010	E11jzpKz9Kw5K7fuARWfRw	5	The oldish man who owns the store is as sweet as can	review	#NAME?	1	3	1
13	e9nN4Xxj	10/21/2011	3rPt0LxF7rgmEUrznoH22w	5	Wonderful Vietnamese sandwich shoppe. Their baguett	review	C1rHp3dm	1	1	0
14	h53YuCiID	1/11/2010	cGnKNX3I9rthE0-TH24-qA	5	They have a limited time thing going on right now with	review	UPtysDF6	1	2	0
15	WGNIYMe	12/23/2011	FvEEw1_OsrYdvwLV5Hrliw	4	Good tattoo shop. Clean space, multiple artists to choose	review	Xm8HXE1	1	2	0
16	yc5AH9H7	5/20/2010	pfUwBKYYmUXeiwrhDluQcw	4	I'm 2 weeks new to Phoenix. I looked up Irish bars in	review	JOG-4G4e	1	1	0
17	Vb9FPCEL	3/20/2011	HvqmdqWcerVWO3Gs6zbrOw	2	Was it worth the 21$ for a salad and small pizza?	review	ylWOj2y7	0	2	0
18	supigcPN	10/12/2008	HXP_0Ul-FCmA4f-k9CqvaQ	3	We went here on a Saturday afternoon and this place	review	SBbftLzfY	3	4	2
19	O510Re68	5/3/2010	j4SIzrIy0WrmW4yr4--Khg	5	okay this is the best place EVER! i grew up shopping at th	review	u1KWcbPI	0	0	0
20	b5cEoKR8	3/6/2009	v0cTd3PNpYCkTyGKSpOfGA	3	I met a friend for lunch yesterday.	review	UsULgP4b	5	6	4
21	4JzzbSbK9	11/17/2011	a0lCu-j2Sk_kHQsZi_eNgw	4	They've gotten better and better for me in the time	review	nDBly08j5	1	1	1
22	8FNO4D3e	10/8/2008	MuqugTuR5DdIPcZ2IVP3aQ	3	DVAP....	review	C6IOtaaYc	2	4	1
23	tdcjXyFLM	6/28/2011	LmuKVFh03Uz318VKnUWrxA	5	This place shouldn't even be reviewed - because it is the	review	YN3ZLOdg	1	1	2
24	eFA9dqXT	7/13/2011	CQYc8hgKxV4enApDkx0IhA	5	first time my friend and I went there... it was delicious!	review	6lg55RIP2	0	0	0
25	IJ0o6b8bJ	9/5/2010	Dx9sfFU6Zn0GYOckijom-g	1	U can go there n check the car out. If u wanna buy 1 there	review	zRlQEDYd	0	1	1
26	JhupPnWt	5/22/2011	cFtQnKzn2VDpBedy_TxlvA	5	I love this place! I have been coming here for ages.	review	13xj6FSvY	0	1	0
27	wzP2yNpV	5/26/2010	ChBeixVZerfFkeO0McdlbA	4	This place is great. A nice little ole' fashion homemade	review	rLtl8ZkDX5	0	0	0
28	qjmCVYkv	1/3/2013	kZ4TzrVX6qeF0OvrVTGVEw	5	I love love LOVE this place. My boss (who is into healthy	review	fpItLlgimc	0	0	0
29	wct7rZKyZ	3/21/2008	B5h25WK28rJjx4KHm4gr7g	4	Not that my review will mean much given the more in-	review	RRTraCQw	2	4	1
30	yz2zQQSi	3/30/2011	Y_FBKae0J5WsBiCtlKSNSA	4	Came here for breakfast yesterday, it had been years sin	review	FB3cGJvYi	1	1	1

Data Loading and Data Pre-processing

- Loading the Data from CSV
- Removing the new lines
- Putting all reviews in the List

Data Exploration

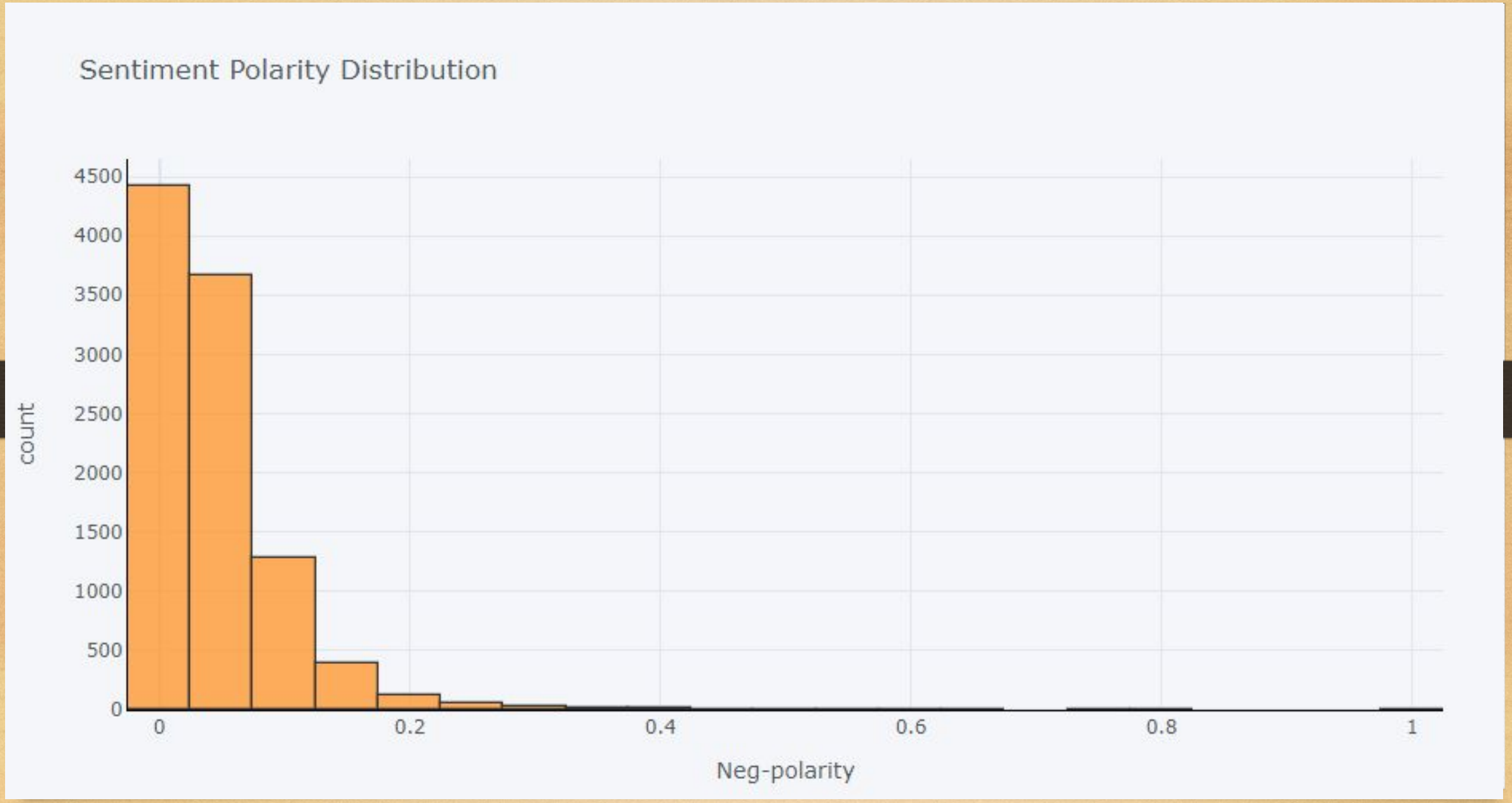
Identifying Model features

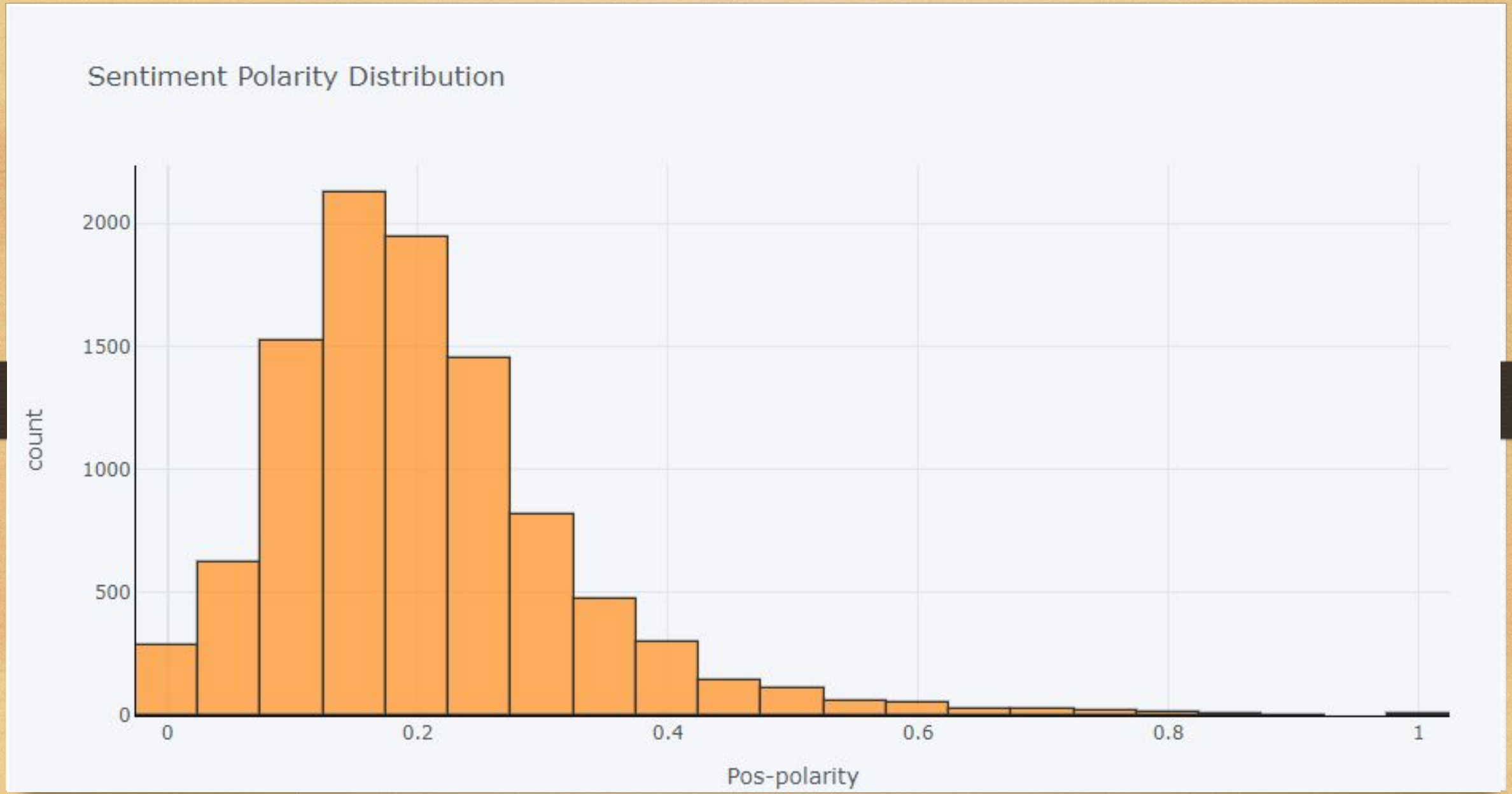
- We Identified the Model features for our Data Set
- Cool, Funny, Useful and Total number of Stars
- Number of words, sentences, and paragraphs

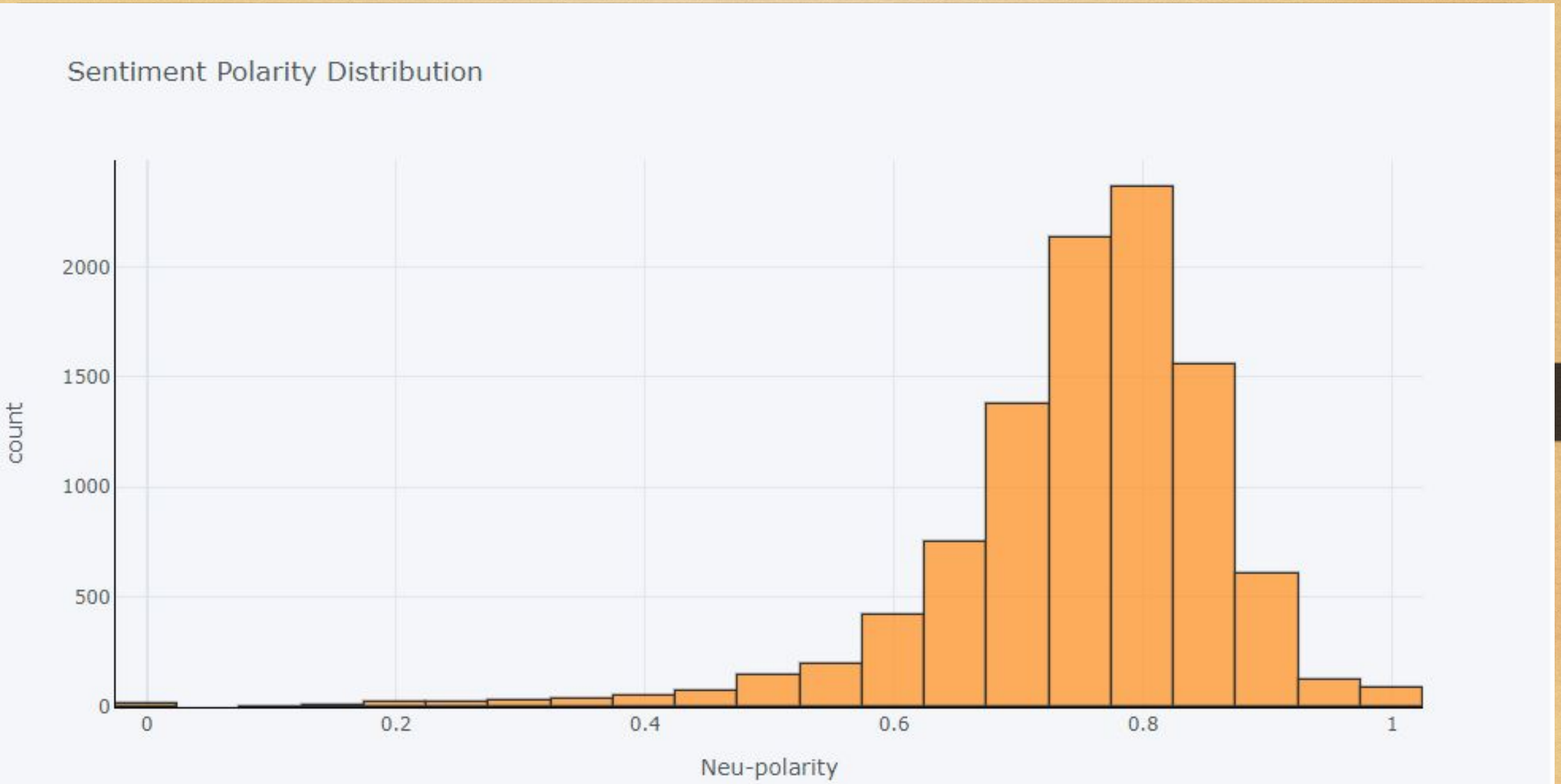
```
#create binary variables for cool, funny, and useful - consider something cool/etc if 2+ votes
df['is_cool']= df.cool>1
df['is_funny']= df.funny>1
df['is_useful']= df.useful>1

#create binary variables for stars
df['star_1']= df.stars == 1
df['star_2']= df.stars == 2
df['star_3']= df.stars == 3
df['star_4']= df.stars == 4
df['star_5']= df.stars == 5

#save other features
df['avg_non_zero_tf_idf'] = avg_non_zero_tf_idf
df['sum_tf_idf'] = sum_tf_idf
df['n_words']= n_words
df['n_sent']= n_sent
df['n_paras']= n_paras

df['exclaim'] = df.text.str.contains('!')
```

Sentiment Analysis with VADER

```
analyzer=SentimentIntensityAnalyzer()

## to account for sentiment, we can include the results from sentinment analysis
#into the model as features

## try both numerical (positive, negative, neutral scores) or binary
# a review can have a high amount of both positive and negative sentiment, so keep both
# positive and negtive aspects seperate instead of using compound score


neg = []
pos =[]
neu = []
for review in all_reviews:
    sent=analyzer.polarity_scores(review)

    neg.append(sent['neg'])
    pos.append(sent['pos'])
    neu.append(sent['neu'])

df['neg']= neg
df['pos']= pos
df['neu']= neu

df['is_neg']= df['neg']>.5
df['is_pos']= df['pos']>.5
df['is_neu']= df['neu']>.5
```

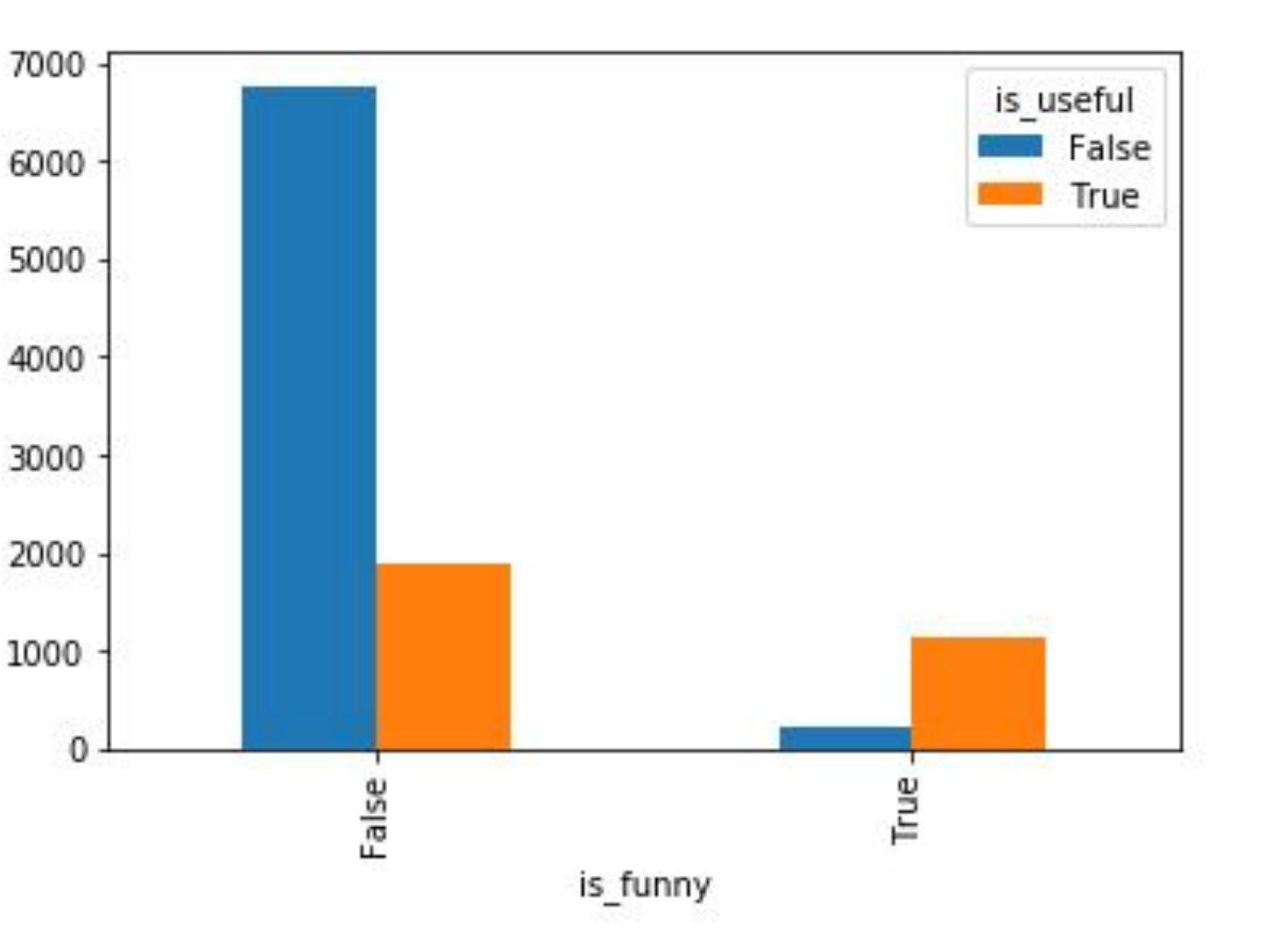

Sentiment Polarity Distribution
4500
4000
3500
3000
2500
2000
1500
1000
500
0
count
0
0.2
0.4
0.6
0.8
1
Neg-polarity


Sentiment Polarity Distribution
count
2000
1500
1000
500
0
0
0.2
0.4
0.6
0.8
1
Pos-polarity


Sentiment Polarity Distribution
count
2000
1500
1000
500
0
0
0.2
0.4
0.6
0.8
1
Neu-polarity

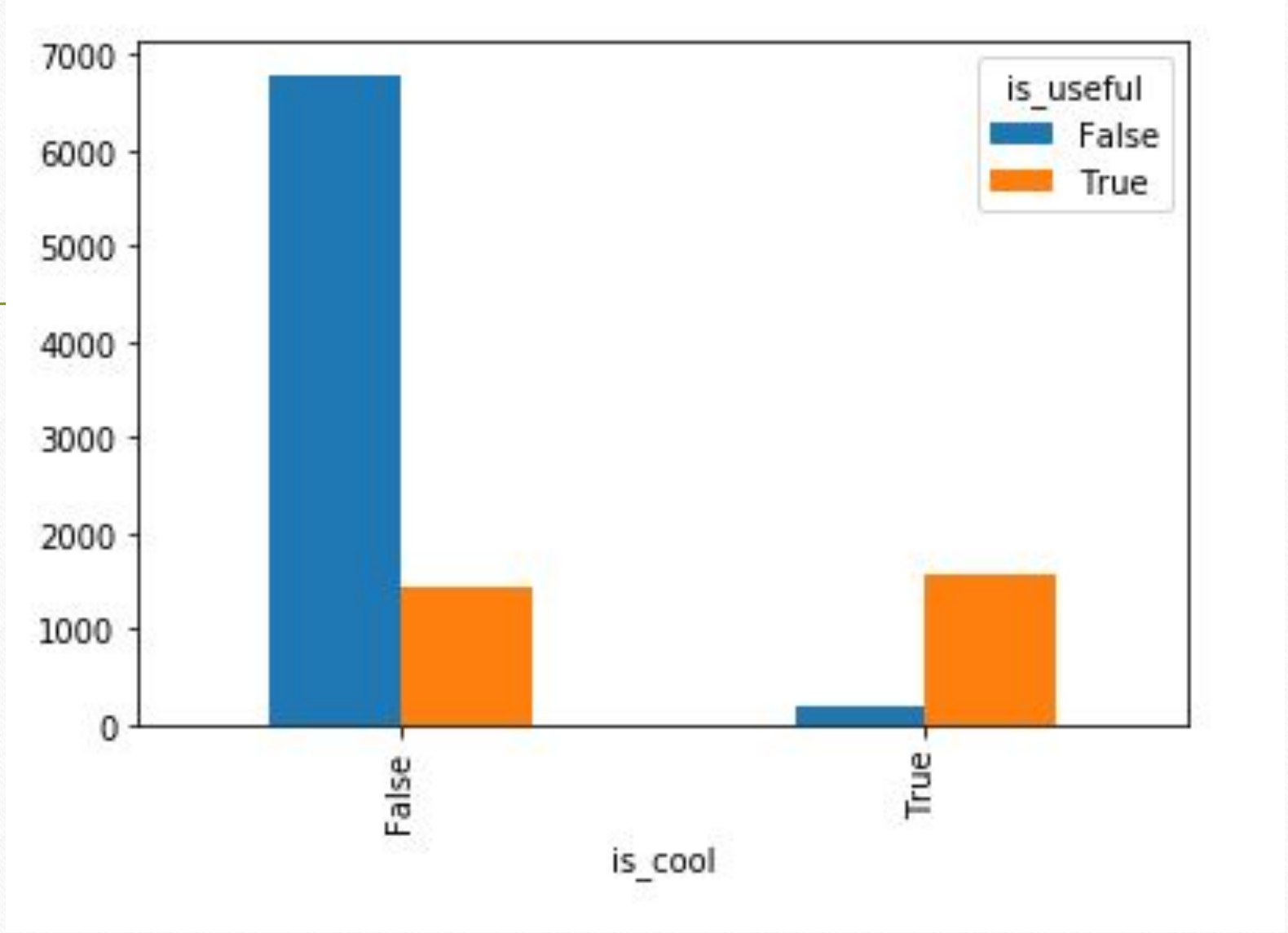
Cross-Tabulation between Model Features (**Funny vs Useful**)

is_useful	False	True
is_funny		
False	6771	1874
True	207	1148



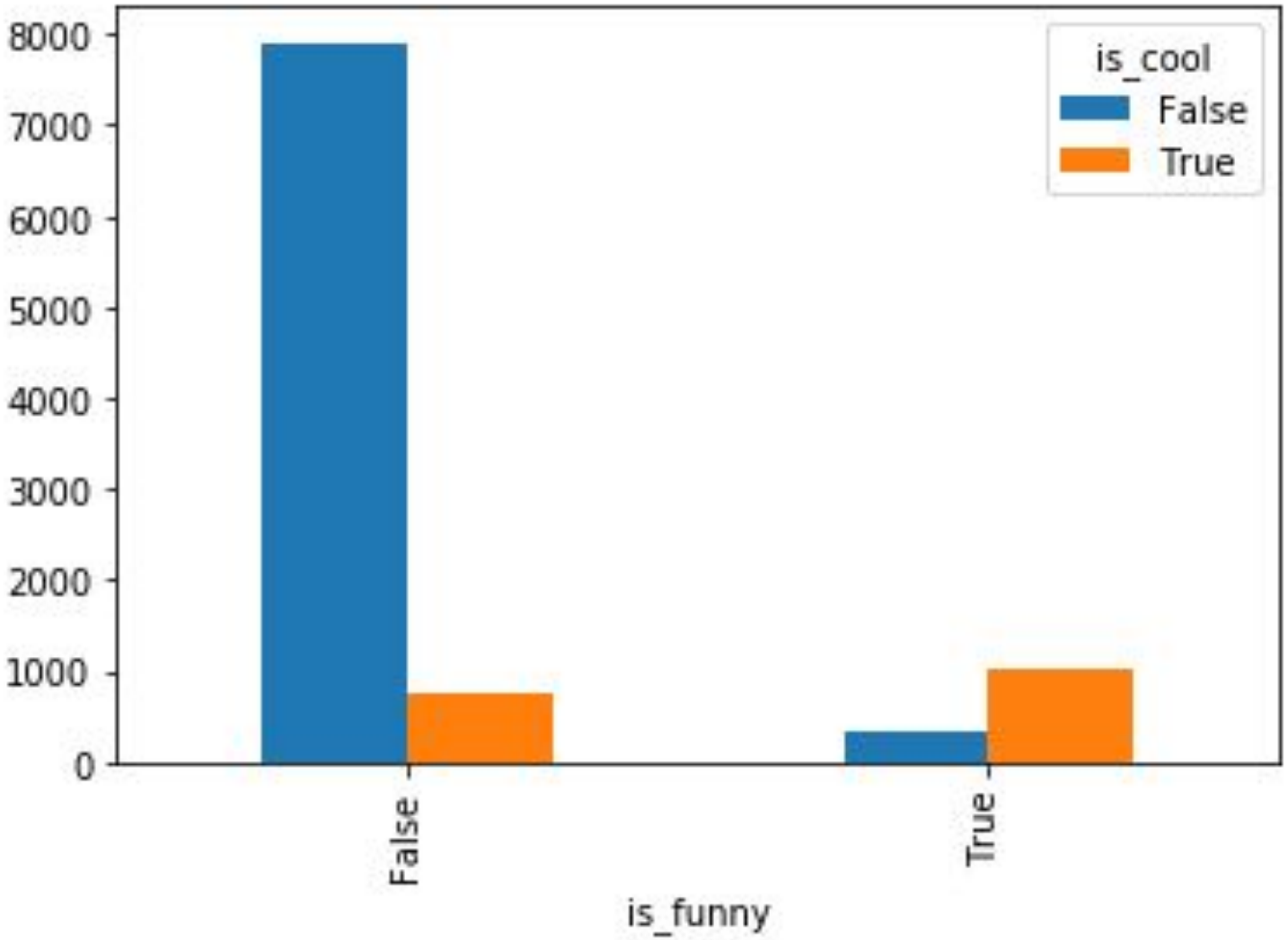
Cross-Tabulation between Model Features (**Cool vs Useful**)

is_useful	False	True
is_cool		
False	6795	1450
True	183	1572

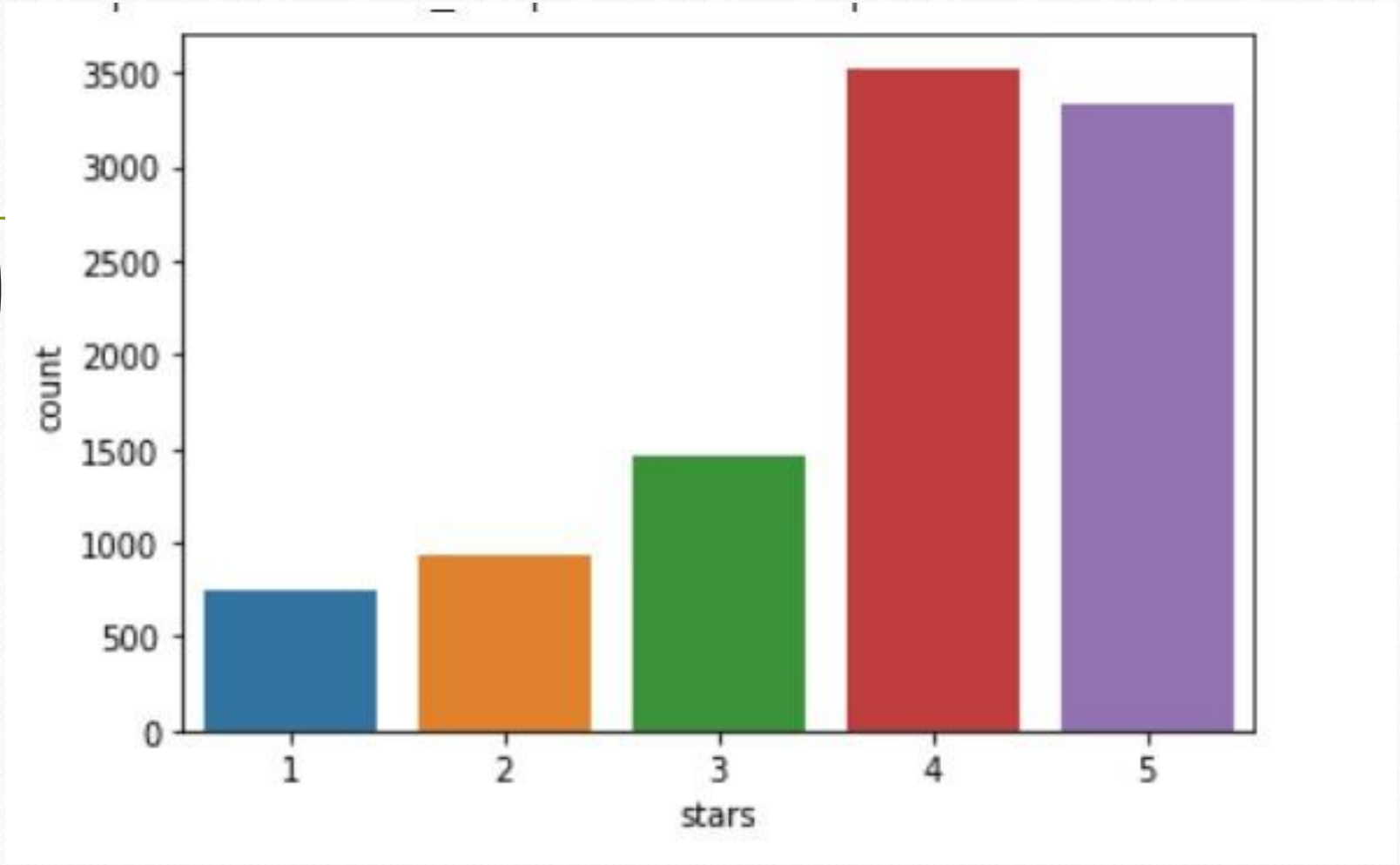


Cross-Tabulation between Model Features (**Funny vs Cool**)

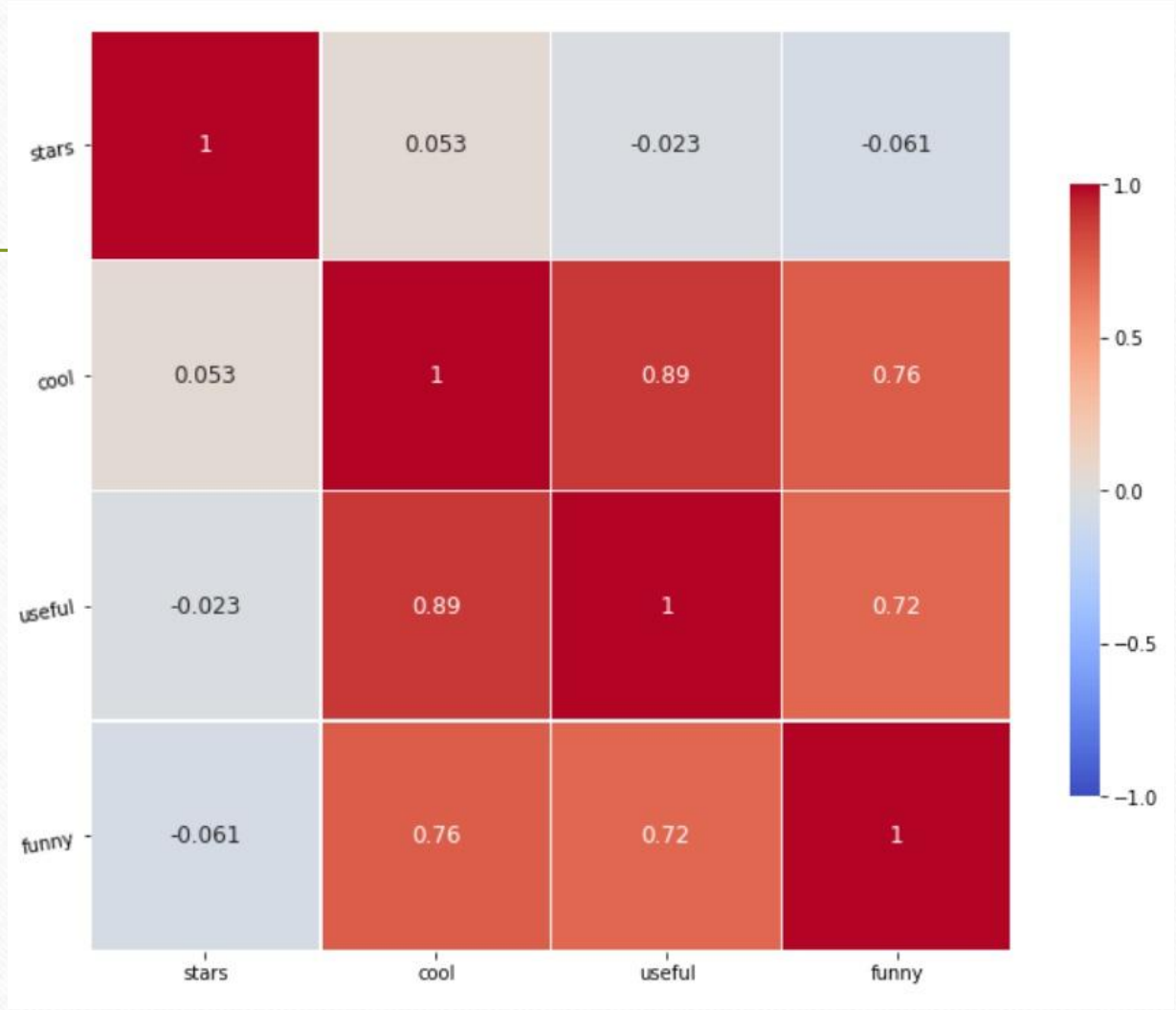
is_cool	False	True
is_funny		
False	7909	736
True	336	1019



Stars Distribution



Heat Map


stars
cool
useful
funny
1
0.053
-0.023
-0.061
0.053
1
0.89
0.76
-0.023
0.89
1
0.72
-0.061
0.76
0.72
1
stars
cool
useful
funny
1.0
0.5
0.0
−0.5
−1.0

Average TF- IDF over non zero values

```
#split into train and test set

X = df[['is_pos','is_neg','is_cool','is_funny','is_useful','star_1','star_2','star_3','star_4','
y = df[['avg_non_zero_tf_idf']]

# Split X and y into X_
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=1)

#fit model
regression_model = LinearRegression(fit_intercept=True)
regression_model.fit(X_train, y_train)


for idx, col_name in enumerate(X_train.columns):
    print("The coefficient for {} is {}".format(col_name, regression_model.coef_[0][idx]))

pred=regression_model.predict(X_test) #make prediction on test set
error = math.sqrt(metrics.mean_squared_error(y_test,pred)) #calculate rmse

print('Test RMSE:: ',error)
print('Test score::',regression_model.score(X_test,y_test)) #R2 score
```

```
The coefficient for is_cool is -0.0045355362765607515
The coefficient for is_funny is 0.0037150613l5379831
The coefficient for is_useful is -0.006841094631285491
```

```
The coefficient for is_pos is 0.1776609982034197
The coefficient for is_neg is 0.2950899698478633
The coefficient for is_cool is -0.0045355362765607515
The coefficient for is_funny is 0.003715061315379831
The coefficient for is_useful is -0.006841094631285491
The coefficient for star_1 is 0.005164424825925323
The coefficient for star_2 is 0.001334938989456654
The coefficient for star_3 is -0.0037651641889698143
The coefficient for star_4 is -0.002491606937099381
The coefficient for star_5 is -0.0002425926893122126
The coefficient for n_sent is -0.0007720840994842391
The coefficient for n_paras is 0.0006264516518845405
The coefficient for n_words is -0.00026134767730915076
Test RMSE::  0.0448485889013482
Test score:: 0.5815141610624752
```

Sum of TF-IDF over all words

```
#split into train and test set

X = df[['is_pos','is_neg','is_cool','is_funny','is_useful','star_1','star_2','star_3','star_4
y = df[['sum_tf_idf']]

# Split X and y into X_
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=1)

#fit model
regression_model = LinearRegression(fit_intercept=True)
regression_model.fit(X_train, y_train)


for idx, col_name in enumerate(X_train.columns):
    print("The coefficient for {} is {}".format(col_name, regression_model.coef_[0][idx]))

pred=regression_model.predict(X_test) #make prediction on test set
error = math.sqrt(metrics.mean_squared_error(y_test,pred)) #calculate rmse

print('Test RMSE:: ',error)
print('Test score::',regression_model.score(X_test,y_test)) #R2 score
```

```
The coefficient for is_cool is 0.1252788605723105
The coefficient for is_funny is -0.05591191427811685
The coefficient for is_useful is 0.20614544660674405
```

```
The coefficient for is_pos is -2.308488693167044
The coefficient for is_neg is -2.959191248893721
The coefficient for is_cool is 0.1252788605723105
The coefficient for is_funny is -0.05591191427811685
The coefficient for is_useful is 0.20614544660674405
The coefficient for star_1 is -0.18245609725604203
The coefficient for star_2 is 0.0029790400279612363
The coefficient for star_3 is 0.14098843552614784
The coefficient for star_4 is 0.0844621058890759
The coefficient for star_5 is -0.04597348418716128
The coefficient for n_sent is 0.013628495451273195
The coefficient for n_paras is 0.005249693844832066
The coefficient for n_words is 0.01811124864823142
The coefficient for exclaim is 0.16591592124229274
Test RMSE::  1.0519696719454859
Test score:: 0.825400746543938
```

Lexical Diversity

```
#split into train and test set

X = df[['pos','neg','is_cool','is_funny','is_useful','star_1','star_2','star_3','star_4','star_
y = df[['LD']]

# Split X and y into X_
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=1)

#fit model
regression_model = LinearRegression(fit_intercept=True)
regression_model.fit(X_train, y_train)


for idx, col_name in enumerate(X_train.columns):
    print("The coefficient for {} is {}".format(col_name, regression_model.coef_[0][idx]))

pred=regression_model.predict(X_test) #make prediction on test set
error = math.sqrt(metrics.mean_squared_error(y_test,pred)) #calculate rmse

print('Test RMSE:: ',error)
print('Test score::',regression_model.score(X_test,y_test)) #R2 score
```

```
The coefficient for is_cool is -0.004106111914817156
The coefficient for is_funny is 0.0075405529330061175
The coefficient for is_useful is -0.009053514687201168
```

```
The coefficient for pos is 0.18038689538091862
The coefficient for neg is 0.13303935534939482
The coefficient for is_cool is -0.004106111914817156
The coefficient for is_funny is 0.0075405529330061175
The coefficient for is_useful is -0.009053514687201168
The coefficient for star_1 is 0.013376026077075909
The coefficient for star_2 is 0.0024796082358712716
The coefficient for star_3 is -0.007479356107359969
The coefficient for star_4 is -0.0053784397282642905
The coefficient for star_5 is -0.002997838477322832
The coefficient for n_sent is -0.004101873019909818
The coefficient for n_paras is 0.0017757875341482304
The coefficient for n_words is -0.0005984189063210627
Test RMSE::  0.07168683795910731
Test score:: 0.6640064175131186
```

Conclusion

- We were able to create linear models for text quality that controlled for different features of a review.
- 'Useful' has negative relationships with lexical diversity and average non-zero of TF-IDF.
- This suggests that lexical diversity and average non-zero of TF-IDF are not measuring Yelp review quality since usefulness has a negative relationship with these measures.
- Sum of TF-IDF may a better quality measure.
- Even though cool and funny usually align, they contribute in opposite directions in the linear models.

Q & A

Thank You