Yelp Data Challenge: What Makes a Review Useful, Funny or Cool?

Ziyang Gao, Lichun Gao, Yang Wu, Mingjie Zeng

Introduction

Problem Definition

- In Yelp app, users can vote for an review with 3 tags:
  - Useful, Cool, Funny
- Each review has a final voting number with this three dimension

- Goal: Build a Natural Language Processing model to predict if a review can be useful, funny, and cool
  - More challenging:Predict how useful, funny and cool a review is?

Problem Significance

- Valuable Review Identification
  - Core Competitiveness of Yelp
  - move out the less useful reviews
  - New review ranking system

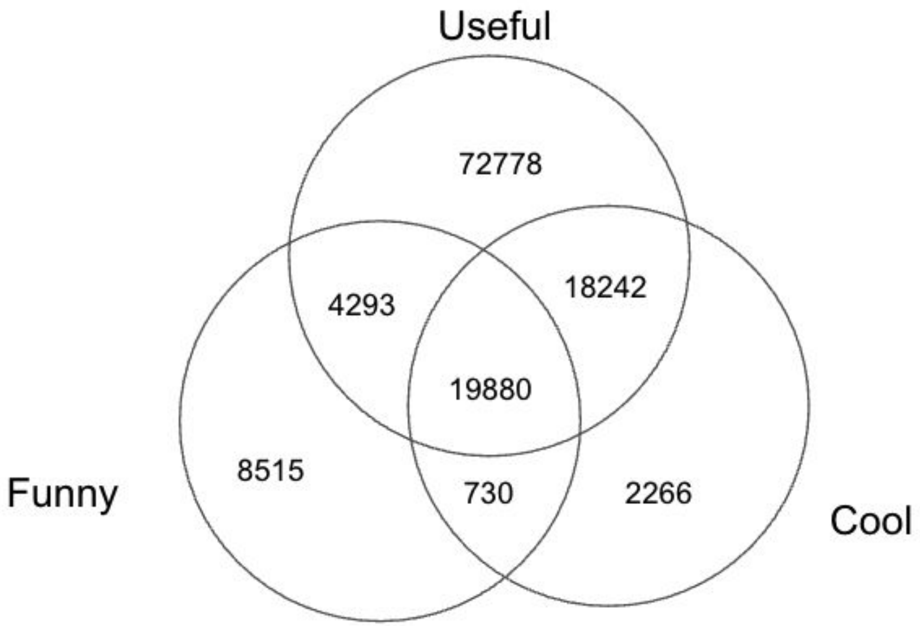
- Increase the “stickiness” of users
  - Current Mode: Search based; More ideal mode: Feed based
  - learn a user’s habitat: prefer useful/funny/cool reviews for which topic?
  - User-specific review feed system

Methodology

Data cleaning

- Raw data: 4 million + reviews
  - Uneven text length, qualities and tag numbers

- Step 1: Filter for reviews with text length greater than 256
- Step 2: useful/funny/cool value > 10 ⇒ encode as 1, else as 0
- Step 3: Random select the reviews with all 0, for negative control



Data Overview

	text	useful	funny	cool	sum_ufc	useful_label	funny_label	cool_label
83921	I've had Copacabana Cuba Café bookmarked for a...	9.0	6.0	11.0	26	0.0	0.0	1.0
83922	What an informative workshop. I live alone but...	10.0	1.0	11.0	22	0.0	0.0	1.0
83923	Stopped here quickly twice during my stay at t...	8.0	1.0	11.0	20	0.0	0.0	1.0
83924	So I really enjoy visiting this location. Spe...	10.0	9.0	13.0	32	0.0	0.0	1.0
83925	My review is specifically for their coffee del...	10.0	2.0	11.0	23	0.0	0.0	1.0

Data shape (83926, 8)

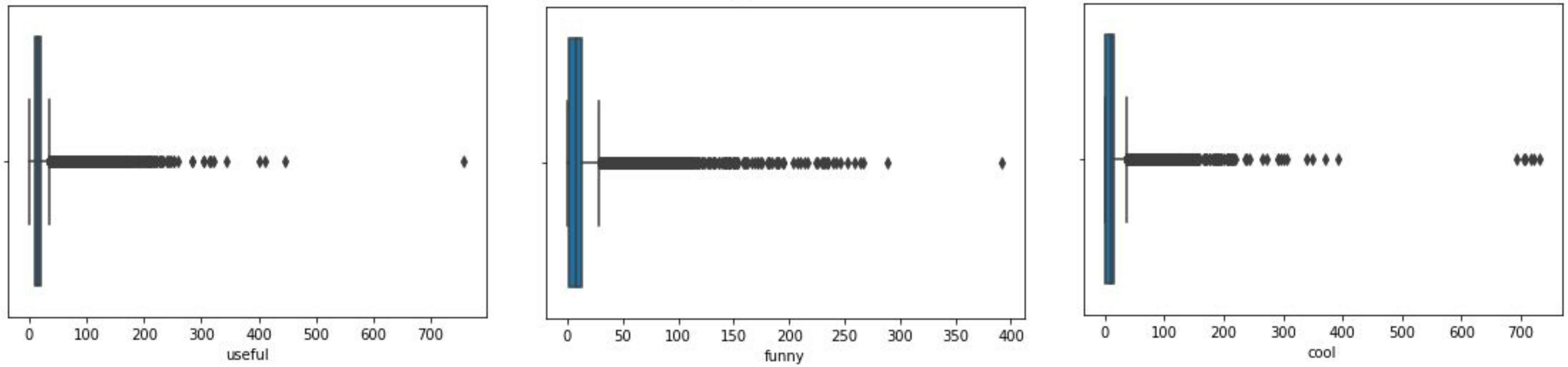
Check missing values

```
main_data.isna().sum()
```

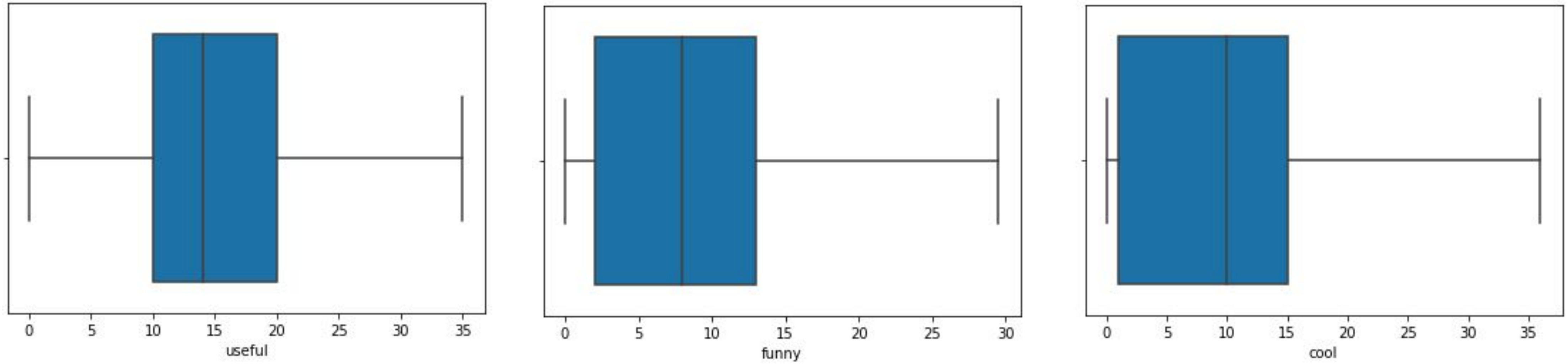
```
text            0
useful          0
funny           0
cool            0
sum_ufc         0
useful_label    0
funny_label     0
cool_label      0
dtype: int64
```

Clean outliers

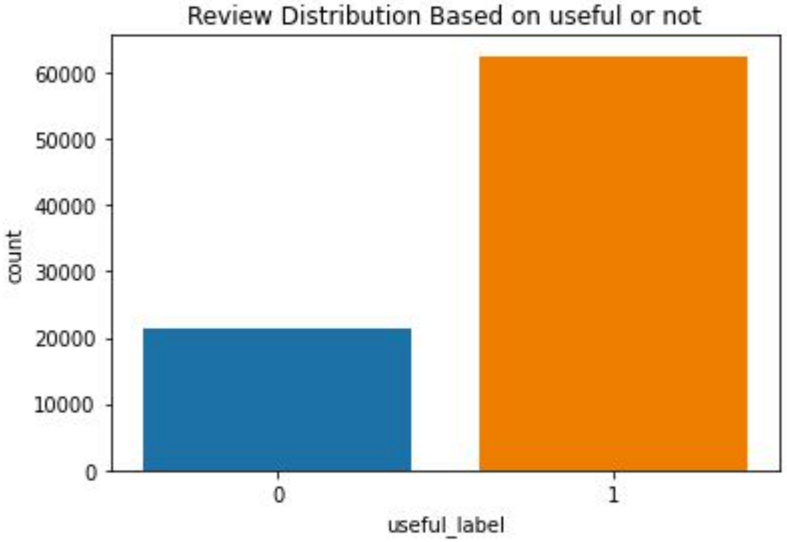
Before cleaning

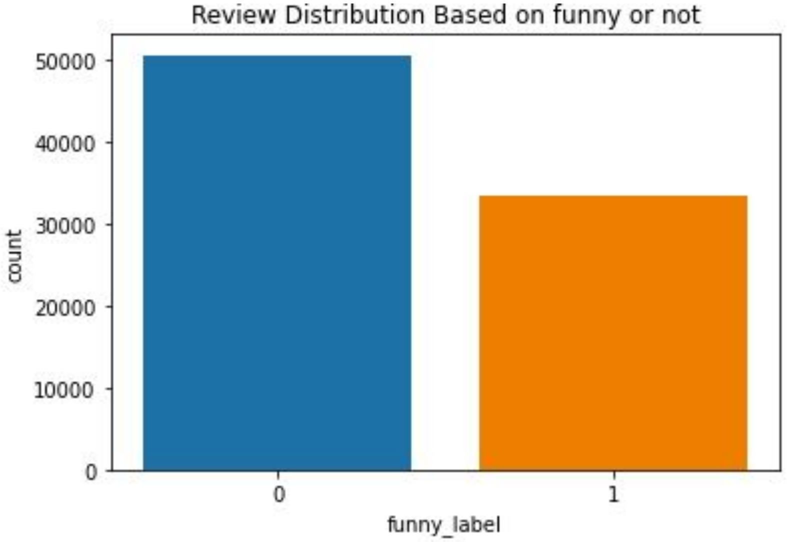


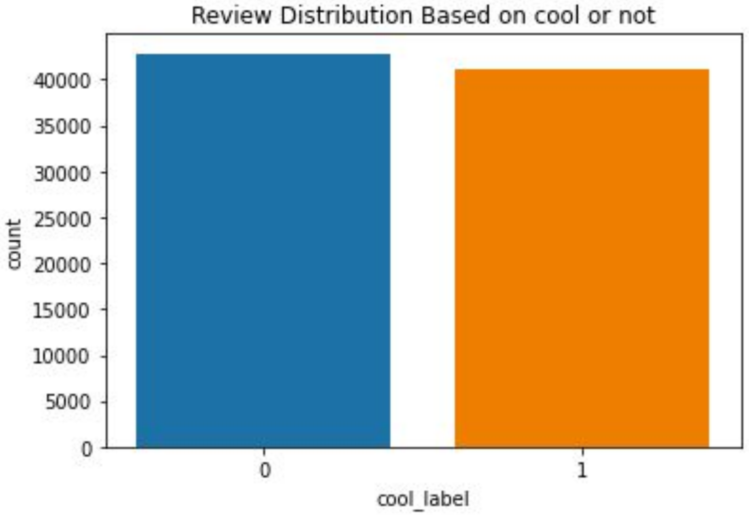
After cleaning



Exploratory Data Analysis (EDA)







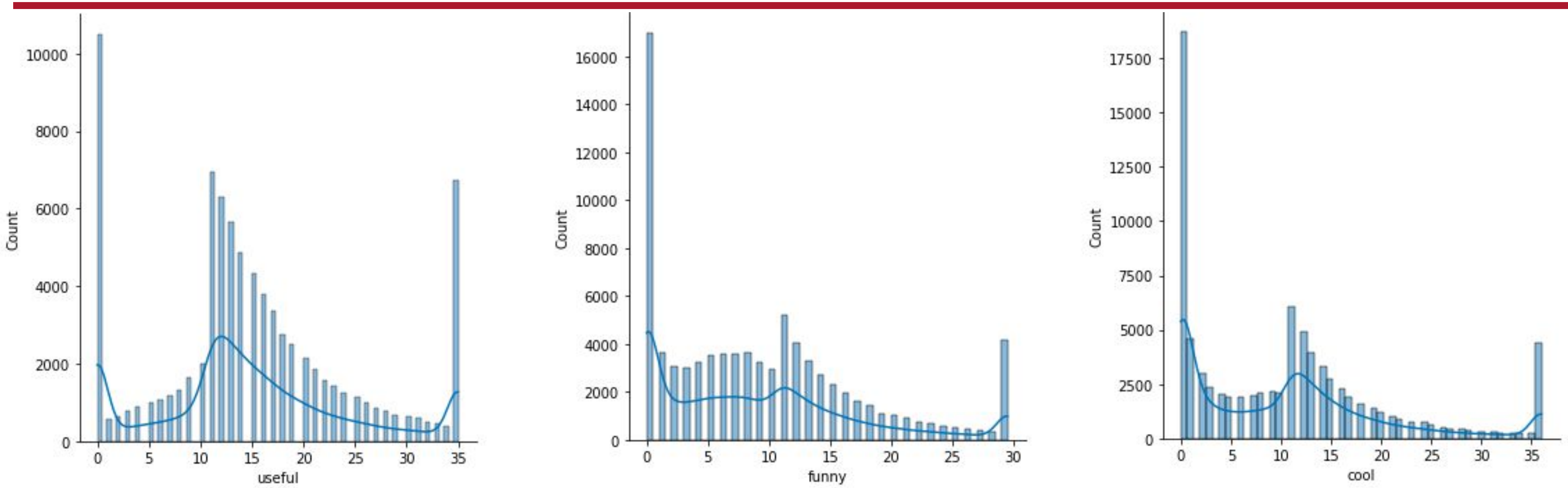
Useful label 0 / label 1 = 1:3

Funny label 0 / label 1 = 5:3

Cool label 0 / label 1 = 1:1

No need to downsample.

Distribution based on count



Data preparation for modeling

- Data split (80% as training set, 20% as testing set)

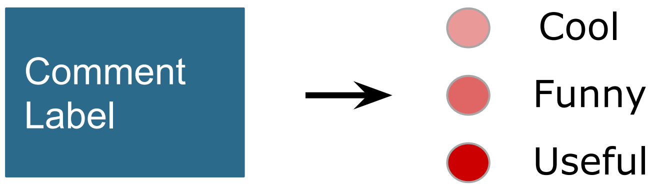
- Features: 1. TFIDF.   2. Word2vec embedding (dim=300)

Multi-label Classification

- Why we use Multi-label Classification?

- Example of multi-label classification

- Multi-label in our project

- 

Techniques for Multi-label Classification: Binary Relevance

- Binary relevance

- Treats each label independently, separating the multi-labels as single-class classification

X	Class1	Class2	Class3
X_1	0	0	1
X_2	0	0	0
X_3	1	0	1

X	Class1
X_1	0
X_2	0
X_3	1

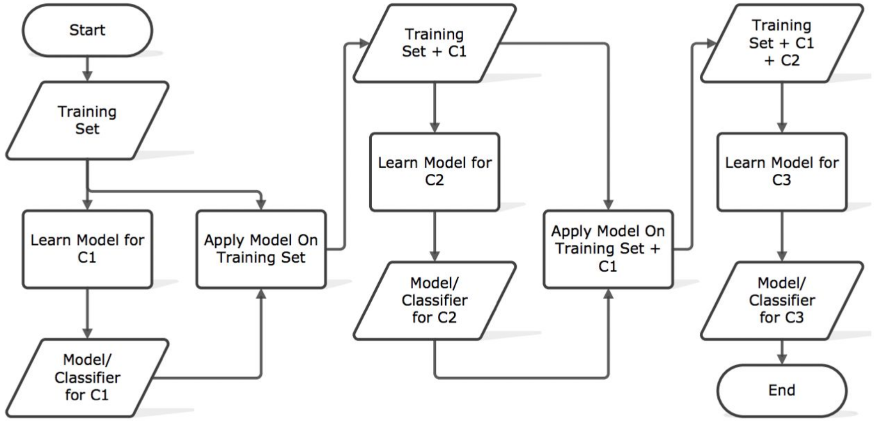
X	Class2
X_1	0
X_2	0
X_3	0

X	Class3
X_1	1
X_2	0
X_3	1

Techniques for Multi-label Classification: Classifier chains

- Classifier chains

- Multiple classifiers connected in a chian



X	Class1	Class2	Class3
X1	0	0	1
X2	0	0	0
X3	1	0	1

X	Class
X1	0
X2	0
X3	1

X	Y1	Class
X1	0	0
X2	0	0
X3	1	0

X	Y1	Y2	Class
X1	0	0	1
X2	0	0	0
X3	1	0	1

Techniques for Multi-label Classification: Label powerset

- Label powerset

- Transform problem to a multi-class problem, train each multi-class classifier with unique label combinators found in data

X	Cla ss1	Cla ss2	Cla ss3
X_1	0	0	1
X_2	0	0	0
X_3	1	0	1

X	Class
X_1	1
X_2	2
X_3	3

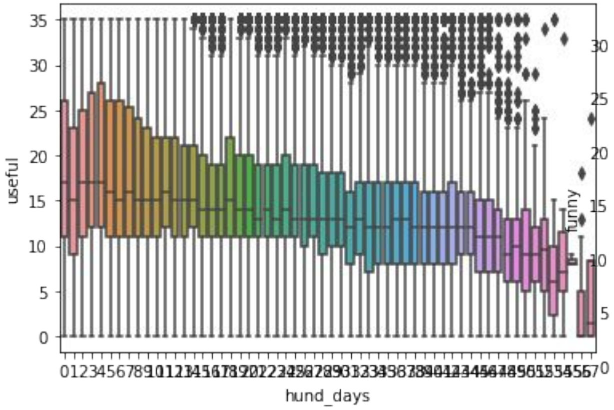
Measuring Metrics

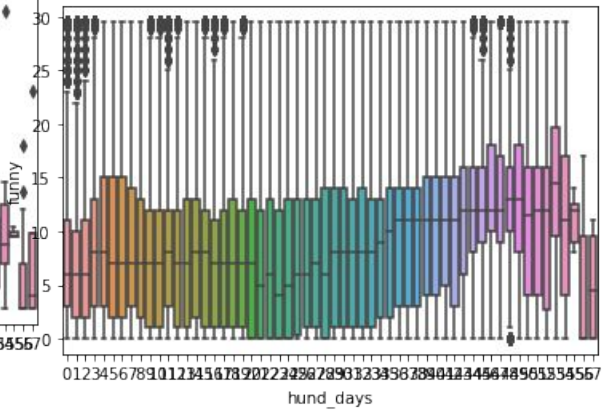
- Accuracy score
- representation of accuracy to make accurate prediction
- Hamming loss
- used to determine the fraction of incorrect predictions of a given model.

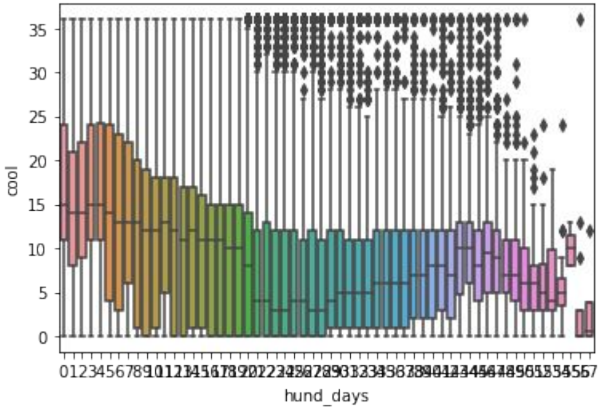
Model Choosing

- Logistic Regression
- Random Forest
- SVM

Normalization and Score Computation

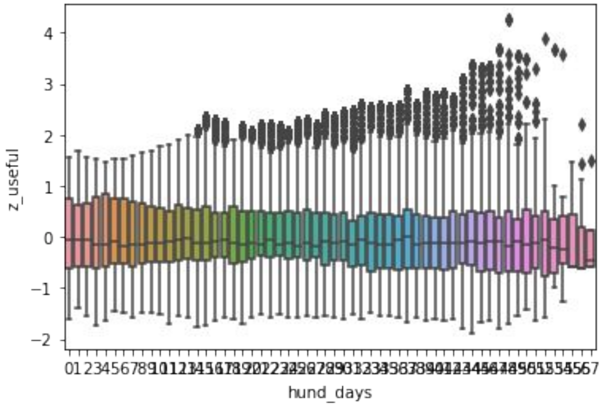


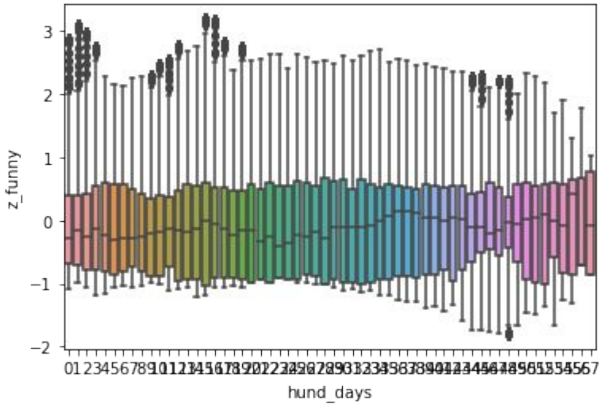


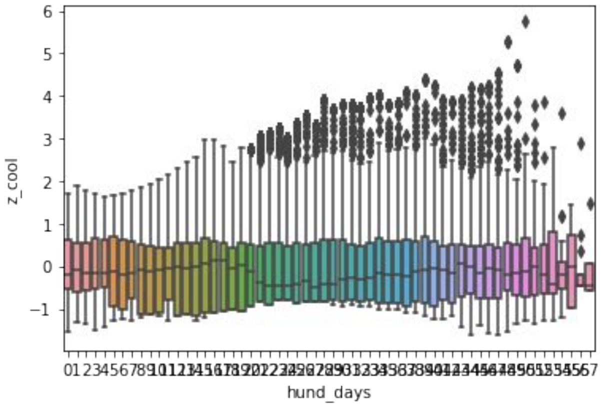


Before

Z-score in every hundred days







After

Multi-output Regression

- Purpose: How useful/funny/cool a review is?

- Input data: review text

```
                                                     text
portland friend gorgeous chinese garden lan su...
rating food beverage star rating business prac...
firm really piece work guardian sister law nev...
place absolute armpit generous fill rental car...
one best escape room ever done currently one e...
```

- Expected output: [useful funny cool] → not a single value

z_useful	z_funny	z_cool
-1.779443468	-1.586943714	-1.418935076
1.501970019	1.850102089	1.977511166
-0.2281003462	-0.05229082098	-0.3410317262

Methods for Multi-output Regression

- Inherently multi-output regression algorithms
- Wrapper multi-output regression algorithms

Multi-output Regression

- Inherently multi-output regression algorithms
    - Linear Regression
    - K-Nearest
    - Random Forest
    - Decision Tree
- Wrapper multi-output regression algorithms
    - SVR - MultiOutputRegressor
    - SVR - RegressorChain

Measuring Metrics

- MSE

$$\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

- MAE

$$\frac{1}{m}\sum_{i=1}^{m}\left|(y_i - \hat{y}_i)\right|$$

- RMSE

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}$$

- R-Squared

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}}$$

Results and Comparison

Classification Results and Comparison

Tf-idf

	Binary Relevance	Classifier Chians	Label Powerset
Logistic Regression	Accuracy: 0.39094 Hamming loss: 0.27692	Accuracy: 0.39094 Hamming loss: 0.27692	Accuracy: 0.42193 Hamming loss: 0.28844
Random Forest	Accuracy: 0.38498 Hamming loss: 0.27811	Accuracy: 0.40286 Hamming loss: 0.27096	Accuracy: 0.42074 Hamming loss: 0.29360
SVM	Accuracy: 0.38975 Hamming loss: 0.28804	Accuracy: 0.38975 Hamming loss: 0.28804	Accuracy: 0.43027 Hamming loss: 0.28963

Classification Results and Comparison

Word2vec

	Binary Relevance	Classifier Chians	Label Powerset
Logistic Regression	Accuracy: 0.37664 Hamming loss: 0.27890	Accuracy: 0.37664 Hamming loss: 0.27890	Accuracy: 0.41955 Hamming loss: 0.29122
Random Forest	Accuracy: 0.41955 Hamming loss: 0.29122	Accuracy: 0.40524 Hamming loss: 0.26897	Accuracy: 0.43147 Hamming loss: 0.28685
SVM	Accuracy: 0.39452 Hamming loss: 0.27652	Accuracy: 0.49452 Hamming loss: 0.27652	Accuracy: 0.42074 Hamming loss: 0.29241

Classification Results and Comparison

Tf-idf

	Binary Relevance	Classifier Chains	Label Powerset
Logistic Regression	'accuracy_useful:': 0.757, 'accuracy_funny:': 0.706, 'accuracy_cool:': 0.735 'hamming_score_useful': 0.243, 'hamming_score_funny': 0.294, 'hamming_score_cool': 0.265	'accuracy_useful:': 0.757, 'accuracy_funny:': 0.708, 'accuracy_cool:': 0.691, 'hamming_score_useful': 0.243, 'hamming_score_funny': 0.292, 'hamming_score_cool': 0.309	'accuracy_useful:': 0.763, 'accuracy_funny:': 0.657, 'accuracy_cool:': 0.711, 'hamming_score_useful': 0.237, 'hamming_score_funny': 0.343, 'hamming_score_cool': 0.289
Random Forest	'accuracy_useful:': 0.713, 'accuracy_funny:': 0.65, 'accuracy_cool:': 0.656, 'hamming_score_useful': 0.287, 'hamming_score_funny': 0.35, 'hamming_score_cool': 0.344	'accuracy_useful:': 0.714, 'accuracy_funny:': 0.649, 'accuracy_cool:': 0.651, 'hamming_score_useful': 0.286, 'hamming_score_funny': 0.351, 'hamming_score_cool': 0.349	'accuracy_useful:': 0.725, 'accuracy_funny:': 0.619, 'accuracy_cool:': 0.636, 'hamming_score_useful': 0.275, 'hamming_score_funny': 0.381, 'hamming_score_cool': 0.364
SVM	'accuracy_useful:': 0.747, 'accuracy_funny:': 0.687, 'accuracy_cool:': 0.711, 'hamming_score_useful': 0.253, 'hamming_score_funny': 0.313, 'hamming_score_cool': 0.289	'accuracy_useful:': 0.747, 'accuracy_funny:': 0.689, 'accuracy_cool:': 0.689, 'hamming_score_useful': 0.253, 'hamming_score_funny': 0.311, 'hamming_score_cool': 0.311	'accuracy_useful:': 0.748, 'accuracy_funny:': 0.67, 'accuracy_cool:': 0.7, 'hamming_score_useful': 0.252, 'hamming_score_funny': 0.33, 'hamming_score_cool': 0.3

Classification Results and Comparison

Word2vec

	Binary Relevance	Classifier Chains	Label Powerset
Logistic Regression	'accuracy_useful:': 0.744, 'accuracy_funny:': 0.595, 'accuracy_cool:': 0.522, 'hamming_score_useful': 0.256, 'hamming_score_funny': 0.405, 'hamming_score_cool': 0.478	'accuracy_useful:': 0.744, 'accuracy_funny:': 0.593, 'accuracy_cool:': 0.534, 'hamming_score_useful': 0.256, 'hamming_score_funny': 0.407, 'hamming_score_cool': 0.466	'accuracy_useful:': 0.727, 'accuracy_funny:': 0.513, 'accuracy_cool:': 0.51, 'hamming_score_useful': 0.273, 'hamming_score_funny': 0.487, 'hamming_score_cool': 0.49
Random Forest	'accuracy_useful:': 0.664, 'accuracy_funny:': 0.577, 'accuracy_cool:': 0.493, 'hamming_score_useful': 0.336, 'hamming_score_funny': 0.423, 'hamming_score_cool': 0.507	'accuracy_useful:': 0.663, 'accuracy_funny:': 0.572, 'accuracy_cool:': 0.508, 'hamming_score_useful': 0.337, 'hamming_score_funny': 0.428, 'hamming_score_cool': 0.49	'accuracy_useful:': 0.698, 'accuracy_funny:': 0.514, 'accuracy_cool:': 0.488, 'hamming_score_useful': 0.302, 'hamming_score_funny': 0.486, 'hamming_score_cool': 0.512
SVM	'accuracy_useful:': 0.745, 'accuracy_funny:': 0.627, 'accuracy_cool:': 0.52, 'hamming_score_useful': 0.255, 'hamming_score_funny': 0.373, 'hamming_score_cool': 0.48	'accuracy_useful:': 0.745, 'accuracy_funny:': 0.627, 'accuracy_cool:': 0.508, 'hamming_score_useful': 0.255, 'hamming_score_funny': 0.373, 'hamming_score_cool': 0.492	accuracy_useful:': 0.745, 'accuracy_funny:': 0.506, 'accuracy_cool:': 0.526, 'hamming_score_useful': 0.255, 'hamming_score_funny': 0.494, 'hamming_score_cool': 0.474

Regression Results and Comparison

Method	MSE	RMSE	MAE	R-Squared
TFIDF - Linear Regression	0.6853857979599436	0.8278803041261119	0.6411115414057668	0.3021841300063293
TFIDF - K-Nearest	0.7446548296979877	0.8629338501287267	0.6702153622706692	0.24191593378901977
TFIDF - Random Forest	0.3566786241916813	0.59722577321452	0.4096955829560201	0.6367505112711895
TFIDF - Decision Tree	0.38215466428350764	0.6181865934194203	0.2730110347877886	0.6108191139659797
TFIDF - SVR(MultiOutputRegressor)	0.7527118874029148	0.8675896999174868	0.6004562673472121	0.23361113672673514
TFIDF - SVR(RegressorChain)	0.7526402373572249	0.8675484063481558	0.60044845086726	0.2336829272385209

Method	MSE	RMSE	MAE	R-Squared
Word2Vec - Linear Regression	0.8325556307210368	0.9124448644828008	0.6979680816051165	0.1524191078393177
Word2Vec - K-Nearest	0.7639216277423456	0.8740261024376478	0.6792615413720968	0.22227114090446143
Word2Vec - Random Forest	0.3599642341720213	0.5999701944030398	0.412263837701742	0.6334021816911375
Word2Vec - Decision Tree	0.4215603233905168	0.6492767694831818	0.28490958987344767	0.57060026790459
Word2Vec - SVR(MultiOutputRegressor)	0.8654279432961328	0.9302837971802652	0.6791484176277084	0.11899644439783852
Word2Vec - SVR(RegressorChain)	0.8651992748144194	0.9301608865214767	0.6790205290254425	0.11924084417762522

Regression Results and Comparison

Method	Label	MSE	RMSE	MAE	R-Squared
TFIDF - Linear Regression	useful	0.7309947190817195	0.8549822916772718	0.6760122032390765	0.2624088261386168
TFIDF - Linear Regression	funny	0.6807706765964469	0.8250882841226428	0.6358222049814012	0.311599585303329
TFIDF - Linear Regression	cool	0.644391998201664	0.8027403055793723	0.611500215996825	0.33254397857704276

Method	Label	MSE	RMSE	MAE	R-Squared
Word2Vec - Decision Tree	useful	0.45127936421291354	0.6717732982285866	0.29681560910000737	0.5446483164647726
Word2Vec - Decision Tree	funny	0.39322857646312637	0.8250882841226428	0.2762162708462687	0.6023643138374114
Word2Vec - Decision Tree	cool	0.42017302949551055	0.648207551248449	0.28169688967406664	0.5647881734115858

Method	Label	MSE	RMSE	MAE	R-Squared
Word2Vec - SVR(RegressorChain)	useful	0.922821475236321	0.9606359743609028	0.7264321129814866	0.06885103624400013
Word2Vec - SVR(RegressorChain)	funny	0.8753397163951009	0.8250882841226428	0.666003732679292	0.1148499127789856
Word2Vec - SVR(RegressorChain)	cool	0.7974366328118356	0.8929930754557034	0.6446257414155495	0.17402158350989128

Conclusion

Conclusion and Limitation

- Tag identification
  - Decent accuracy for the tag identification
  - For individual case, the model has better performance
  - Useful is the easiest tag to identify compared with Cool and Funny

- Tag score regression
  - Not as good as the tag identification
  - tf-idf with decision tree has the best overall performance
  - different methods have various sensitivity to different tags

Future Works

- Challenging question to think about:
- 1. Can we predict how useful, funny or cool the comment is?
- 2. Using the prediction of labels to judge if the comments are from real customers.
- 3. Analyze images in comments and predict labels for them.

Questions?