

DataMining Project

Emotional Analytics Mining

EECS4412

Alexandra Zaslavsky

Omar Anwar

<b>Introduction</b>	<b>3</b>
<b>Our approach:</b>	<b>4</b>
Data Preprocessing:	4
Special Cases of Punctuation	5
Removing Common Cases	5
Removing Stop Words	6
Stemming words	6
Tokenizing	6
Text Representation	6
Creating our Model:	7
Attribute Selection	8
Learning Algorithms	8
Classifying the Test Data	8
<b>Conclusions</b>	<b>8</b>

# Introduction

The purpose of this assignment was to train and create a model of data from Yelp reviews which classified reviews as positive, negative and neutral. Using this model, created using our training data of 40000 reviews, we were required to classify our test data of 10,000 records as positive, negative and neutral.

This process can be divided into 3 major portions: preprocessing the data, training a classification model using the filtered data and classifying the test data by predicting. Each of these portions are further subdivided into many parts.

The data preprocessing stage was the most resource intensive part of the project. In data mining, this step is done to extract only the useful portion of data from the entire dataset. Data is also transformed in a way where redundancies are highlighted to emphasize its relation to the classification. Any portion of the data that could negatively skew the model and its training is also removed. Paying attention to details when conducting this step can tremendously improve the accuracy and efficiency of training the classification model as there will be attributes used for classification than if we just used the raw data set. These steps are explained in full detail in the preprocessing section of the report. As there were thousands of records to process, we wrote python scripts to automate this process.

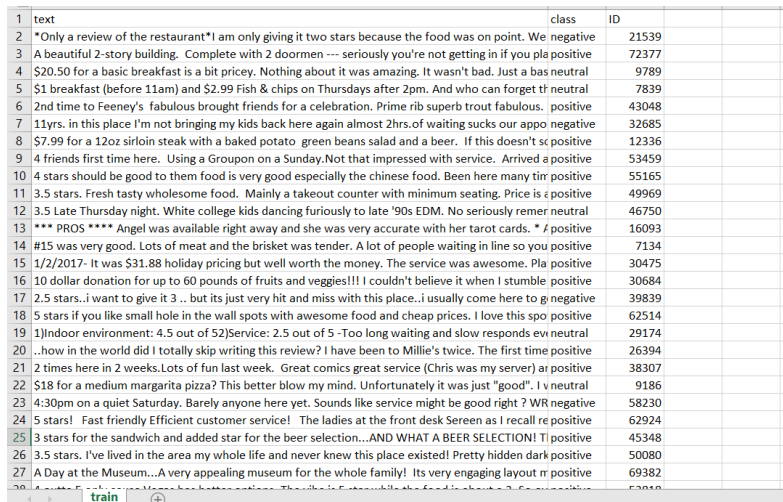
The next step is to train the model. As this project is about text mining, we must identify the most useful terms and their weights on the correct classifications. We must then choose a training algorithm that works best for our solution.

Once we've created the training model, we are ready to classify the data.

# Our approach:

## Data Preprocessing:

The raw data provided was an excel spreadsheet of yelp reviews with a column for the review text, one for its correct classification and one for the ID of the reviews as in the picture below. We wrote python scripts to carry out a number of data preprocessing steps which allowed us to filter the input that would train our classification model more efficiently and return more accurate results when used for prediction on a test set.



	text	class	ID
1			
2	*Only a review of the restaurant*I am only giving it two stars because the food was on point. We	negative	21539
3	A beautiful 2-story building. Complete with 2 doormen ---- seriously you're not getting in if you pla	positive	72377
4	\$20.50 for a basic breakfast is a bit pricey. Nothing about it was amazing. It wasn't bad. Just a bas	neutral	9789
5	\$1 breakfast (before 11am) and \$2.99 Fish & chips on Thursdays after 2pm. And who can forget th	neutral	7839
6	2nd time to Feeney's fabulous brought friends for a celebration. Prime rib superb trout fabulous.	positive	43048
7	11yrs. in this place I'm not bringing my kids back here again almost 2hrs.of waiting sucks our appo	negative	32685
8	\$7.99 for a 12oz sirloin steak with a baked potato green beans salad and a beer. If this doesn't sc	positive	12336
9	4 friends first time here. Using a Groupon on a Sunday.Not that impressed with service. Arrived a	positive	53459
10	4 stars should be good to them food is very good especially the chinese food. Been here many tim	positive	55165
11	3.5 stars. Fresh tasty wholesome food. Mainly a takeout counter with minimum seating. Price is €	positive	49969
12	3.5 Late Thursday night. White college kids dancing furiously to late '90s EDM. No seriously reme	neutral	46750
13	*** PROS **** Angel was available right away and she was very accurate with her tarot cards. *	positive	16093
14	#15 was very good. Lots of meat and the brisket was tender. A lot of people waiting in line so you	positive	7134
15	1/2/2017- It was \$31.88 holiday pricing but well worth the money. The service was awesome. Pla	positive	30475
16	10 dollar donation for up to 60 pounds of fruits and veggies!!! I couldn't believe it when I stumble	positive	30684
17	2.5 stars..i want to give it 3 ... but its just very hit and miss with this place..i usually come here to g	negative	39839
18	5 stars if you like small hole in the wall spots with awesome food and cheap prices. I love this spo	positive	62514
19	1)Indoor environment: 4.5 out of 52)Service: 2.5 out of 5 -Too long waiting and slow responds evi	neutral	29174
20	..how in the world did I totally skip writing this review? I have been to Millie's twice. The first time	positive	26394
21	2 times here in 2 weeks.Lots of fun last week. Great comics great service (Chris was my server) at	positive	38307
22	\$18 for a medium margarita pizza? This better blow my mind. Unfortunately it was just "good". I v	neutral	9186
23	4:30pm on a quiet Saturday. Barely anyone here yet. Sounds like service might be good right ? WR	negative	58230
24	5 stars! Fast friendly Efficient customer service! The ladies at the front desk Sereen as I recall re	positive	62924
25	3 stars for the sandwich and added star for the beer selection...AND WHAT A BEER SELECTION! TI	positive	45348
26	3.5 stars. I've lived in the area my whole life and never knew this place existed! Pretty hidden dark	positive	50080
27	A Day at the Museum...A very appealing museum for the whole family! Its very engaging layout rr	positive	69382
28	A cute place with a nice atmosphere. The staff is friendly and the food is about a 3.5 star	positive	53818

Figure 1. screenshot of raw data

Our data preprocessing involved the following steps:

1. Special cases of punctuation
2. Remove common cases
3. Removing stop words
4. Stemming words
5. Tokenizing
6. Removing low frequency words
7. Text representation

## 1. Special Cases of Punctuation

Initially, we had decided that punctuation was irrelevant to classification as it did not signify anything. In fact, this case added more words to process which would further confuse our model. “Review.” and “review” would be considered different words and would therefore not give us any data of value if treated as different attributes. Therefore, we decided to remove all punctuation from the words.

Exclamation mark was another symbol we had to consider. An exclamation is generally a good indicator of strong emotion. Some reviews had multiple exclamation marks and would therefore emphasize an emotion. However, this ended up being unreliable as these appeared in both positive and negative reviews and were not a good indicator of either.

In dealing with periods, we had to be careful not to interfere any other punctuation. Some reviewers had no space on either side of the period and so dropping them would combine words to create a new word. There were also numbers and decimals to consider. If a period was replaced with a space, it would create 2 numbers instead of the decimal. To counter this, we dropped all periods from decimals and replaced all other periods with spaces.

The last special case of punctuation we encountered was emoticons. These were a good indicator for classes. To deal with these, we replaced them with words that would demonstrate this emotion. We used “happyface” for a smiley and “sadface” for sad ones. This corresponded well with classification.

All other punctuations were dropped as they were not beneficial for the classification process.

## 2. Removing Common Cases

The first of these cases was to replace numbers written as words by their numerical values. This increased the frequency of the numbers in each entry increased their weights with respect to a certain classification. This was a good strategy as both representations essentially indicated the same value and could therefore be considered synonyms.

We also dealt with other synonyms similarly. We replaced the string “% off” with discount as both were present in the data. This was generally something that carried a positive sentiment with it and was therefore useful.

Another good indicator of which class a review belonged to, was the stars representation of each review. There were over 11 different representations of this and required standardizations. We changed these phrases to “[x]/5stars” where x was the number given. In the case of decimals, we dropped the decimals and made these single numbers. This worked since all instances of “45/5stars” indicated a positive review while smaller ones represented negative ones.

We then removed any times and dates from the dataset as this was completely irrelevant to the class.

### 3. Removing Stop Words

Stop words represent common words in sentences that are not relevant to the classification. To remove stop words, we used the stop words list provided in the project. The list is included as part of the report. We dropped these words and replaced them with spaces. This was the list of most commonly occurring words. As these were everyday words such as “a” or other common words, they would appear in almost all reviews and would not give any significant data.

### 4. Stemming words

The next step was to standardize different instances of the same word. If, for example, we had instances of “plan”, “planned”, “planning” etc or any variation of the word “plan” in the text, we combined it into one word to have a useful representation and increase the count of the words we do have in our data, giving each word more value and reducing the overall number of words in the document. The words were also converted to lowercase to help with standardizing the process.

### 5. Tokenizing

We then iterated over the remaining words and obtained a list of unique words appearing in the whole dataset. We decided to carry out phase one of feature selection here. At this stage, we had some words that had a very low frequency, meaning their significance to the classification was negligible. This helped reduce noise in our data and gave us a more useful list of unique words that were related to our classes.

### 6. Text Representation

At this stage, we decided to create a matrix of with unique words appearing in all the reviews as attributes at the top, the last column being our class and the second last being the IDs of the reviews. Doing so, the training model would be able to see the importance of each word in the document with respect to a positive, negative and neutral classification for the test set.

We decided the best representation of this data is TF-IDF(Term Frequency-Inverted Document Frequency) method and use weights as attribute values. This is a good representation as we account for word frequency in each review, and we multiply it by the IDF which would help regularize the data. This makes rare words across the records more important. The Term frequency increases the weights of the most important words and the IDF increases the importance of the rarest words. By increasing the importance of both extremes, words not too that do not affect the review as much are less important in their weights and this allows the training model to have a more accurate understanding of the words in the reviews.

The picture below shows how the raw data was transformed at the end of the preprocessing stage. If a unique word did not appear in a review it got a weight of 0, otherwise it was given a

weight helped identify its importance to the classification

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	class	review	restaur	give	2/5star	food	point	order	fri	green	tomato	bowl	mac	n	chees	0.69897	2 burger	took	hour	half	extrem	hot	run	ba
2	neg	1.113943	0.954243	1.80618	1.568202	1.431364	1.544068	1.39794	2.227887	1.653213	1.90309	1.681241	1.991226	1.633468	1.255273	0.69897	1.342423	1.255273	1.079181	1.491362	1.643453	1.255273	1.60206	1.
3	posit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.39794	0	0	0	0	0	0	0	0
4	neutral	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.
5	neutral	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	posit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	neg	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	posit	0	0	0	0	0	0	0.69897	0	3.306425	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	posit	0	0	0	0	0	0	0.69897	0	1.653213	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	posit	0	0	0	0	1.431364	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	posit	0	0	0	0	0.477121	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	neutral	0	0	0	0	0.954243	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	posit	0	0	0	0	0	0	0	0	0	0	0	0	0	1.633468	0	0	0	0	0	0	0	0	0
14	posit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	posit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.255273	0	0
16	posit	1.113943	0	0	0	0	0	0	0	0	1.90309	0	0	0	0	1.39794	0	0	0	0	0	0	0	0
17	neg	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.491362	0	0	0	0
18	posit	0	0	0	0	0.477121	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	neutral	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.69897	0	0	0	0	0	0	0	0
20	posit	1.113943	0	0	0	0	0	1.39794	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	posit	0	0	0	0	0.954243	0	0.69897	0	0	0	0	3.982452	0	2.510545	1.39794	0	0	0	0	0	1.255273	0	0
22	neutral	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	neg	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	posit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	posit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	posit	0	0.954243	0	0	0.477121	0	2.09691	0	0	0	3.362482	0	0	1.255273	0	0	0	0	0	0	0	0	0
27	posit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.079181	0	0	0	0	0
28	posit	0	0	0	0	0.477121	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	posit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2. Preprocessed Training Data

## Creating our Model

For this process, we used Weka, a data analysis tool that would trains data using csv input files. It has multiple training algorithms available and each have their own benefits in terms of accuracy and performance. In this tool we can also see a graphic representation of datapoints and further prune our data using attribute selection to reduce the number of attributes and increase performance and speed of the training process.

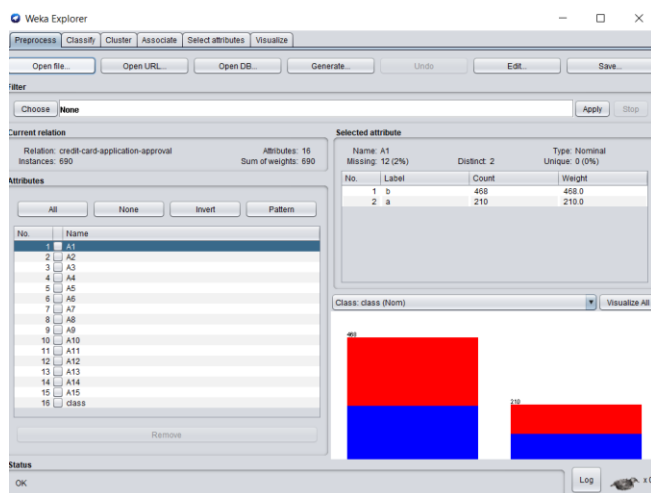


Figure 3. Weka software with sample attributes loaded to begin training a classification model

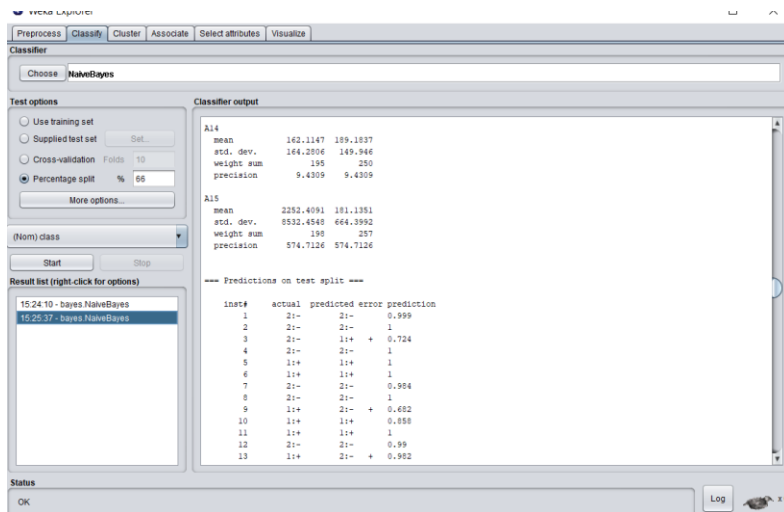


Figure 4. Training a classification model

## 1. Attribute Selection

For the attribute selection part, we loaded the training data into Weka as a CSV file. We chose Information Gain to rank our attributes in terms of importance. We reduced the number of attributes by half. We chose to use Information Gain as it considers all the classes each review could have which was a nice complement to our use of TF-IDF.

## 2. Learning Algorithms

We chose to use a Naive Bayesian model as it is one of the faster running models. We found that while pre-processing our data, our scripts would take large amounts of time to complete as there were large amounts of records to clean as a result, when it came time to training a model, we decided to sacrifice accuracy for speed. We also knew that having such a large dataset would increase accuracy, which made us more comfortable making the sacrifice.

## Classifying the Test Data

For this step, we had to preprocess the training data as before by giving weights to each unique word that appeared in the test data. The attributes we used for this step were the unique words that appeared in the training data as they were the words that the classification model was trained on. The weight given to each word with respect to each review showed how important each word in the training data was with respect to the test reviews. After assigning the weights, the test data was used as an input file in weka and the classification was done. The images below show the test data that was input into weka and the final predictions that we obtained using our trained model. Both files containing this data were submitted along with this report.



1	review	restaur	give	2/5star	food	point	order	fri	green	tomato	bowl
2	0	0	0	0	0.477121	0	0	0	0	0	0
3	0	0	0	0	0.477121	0	0	0	0	0	0
4	0	0	0	0	0.954243	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	2.089905	0
9	1.322219	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0.477121	0	0	0	0	0	0
11	0	0	0	0	0	0	1	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0
13	1.322219	0	0	0	0.954243	0	0	0	0	0	0
14	0	0	0	0	0	0	1	0	0	0	0
15	0	0	0	0	0.477121	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0.477121	0	0	0	0	0	0
19	0	0	0	0	0.954243	0	0	0	0	0	0
20	0	0	0	0	0.954243	0	0	0	0	3.584783	0
21	0	0	0	0	0.477121	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	1.792392

Figure 5. Preprocessed Test Data

	A	B	C
1	REVIEW-ID	CLASS	
2	52447	positive	
3	68745	positive	
4	69711	positive	
5	35164	positive	
6	31360	positive	
7	67627	negative	
8	39339	neutral	
9	61907	positive	
10	37550	negative	
11	44796	positive	
12	36976	negative	
13	68899	positive	
14	55569	negative	
15	7366	positive	
16	33952	positive	
17	27689	positive	
18	55763	positive	
19	29553	negative	
20	49221	neutral	
21	56576	positive	
22	8533	negative	
23	7397	neutral	
24	35460	negative	
25	30184	negative	
26	12470	positive	
27	72201	positive	
28	50617	positive	

Figure 6. Our Final Prediction Results

## Conclusions

Overall, this project taught us the complexities involved with pre-processing and working with large datasets. We learned the importance of having not only accurate models, but also fast running time algorithms as real world analysis is often done on large datasets. We were required to read, transform and process large sets and rewrite it in a usable form for the training process.

Our solution could be further improved by pruning the data more to reduce the amount of noise our data had and improve the overall prediction. We could also experiment with other, more accurate prediction models such as neural networks and weigh the pros and cons of each further and choose the most accurate or