Natural Language Processing Assignment 2
Name: Shiang Hu
PID: A53267858

Supervised: Improve the Basic Classifier:

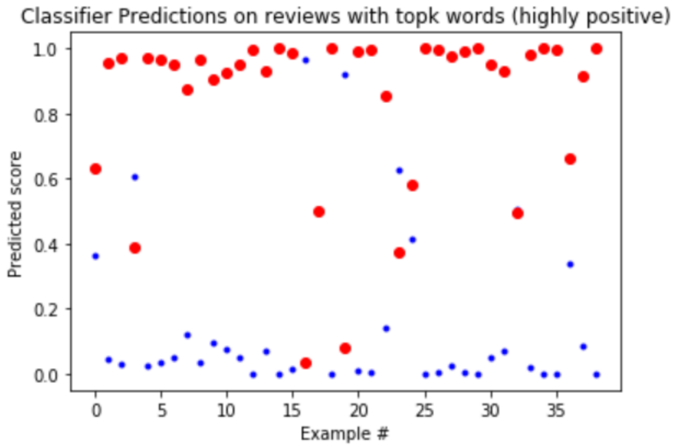
***stop_words*:**

Select words that are trivial, and set them to stop words. First, we produce a naïve count_vector model, we then sort the coefficient trained by logistic regression and select the top k highest coefficients meaning the top k most influential coefficients for positive prediction, and top k lowest coefficients meaning the top k most influential coefficients for negative prediction, and by ruling out these words in our training bag of words what's left are trivial words. We set the "stop_words" parameter in CountVectorizer to these trivial words so that these words will be set to stop_words at training. We experiment of several k values and try to find the best k value to set.

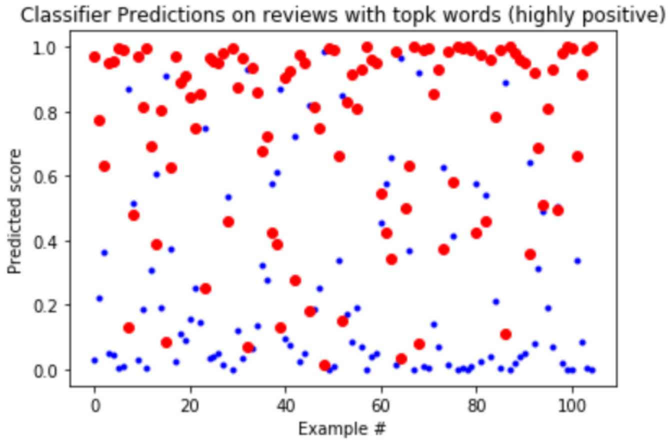
k	0	500	1000	2000	3000	3750	4000	4500
accuracy on dev	0.777292	0.744541	0.768558	0.762008	0.770742	0.783842	0.781659	0.779475

the intuition of selecting stop words in this way is that sentences with these words that has high coefficient has higher prediction confidence and has better accuracy. Confidence here means the difference between the possibility of predicting positive or negative in the logistic regression model. The higher the difference the higher the confidence and the classifier classify these sentences with ease. By selecting the sentences with words with the highest coefficients and test the accuracy of these sentences, and plot their confidence we see that these data have higher accuracy and confidence. So perhaps if we let sentence have more portion of these words we might increase accuracy.

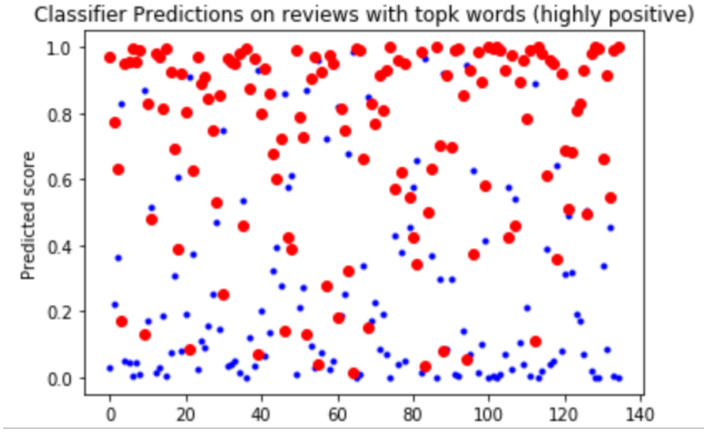
accuracy of sentence with top 5 highest coefficient words: 0.897435



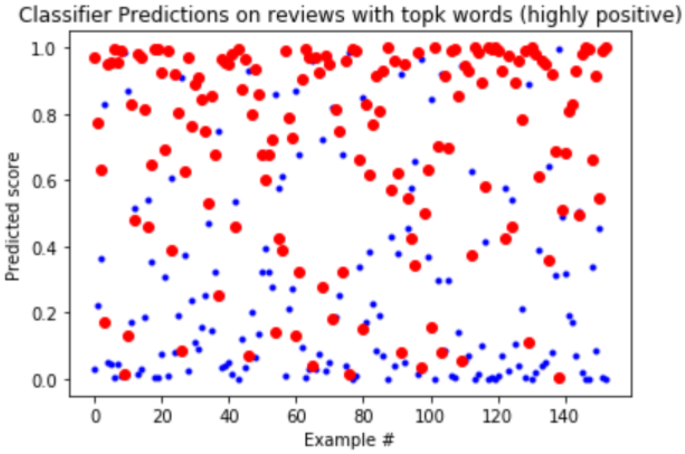
accuracy of sentence with top 10 highest coefficient words: 0.876190



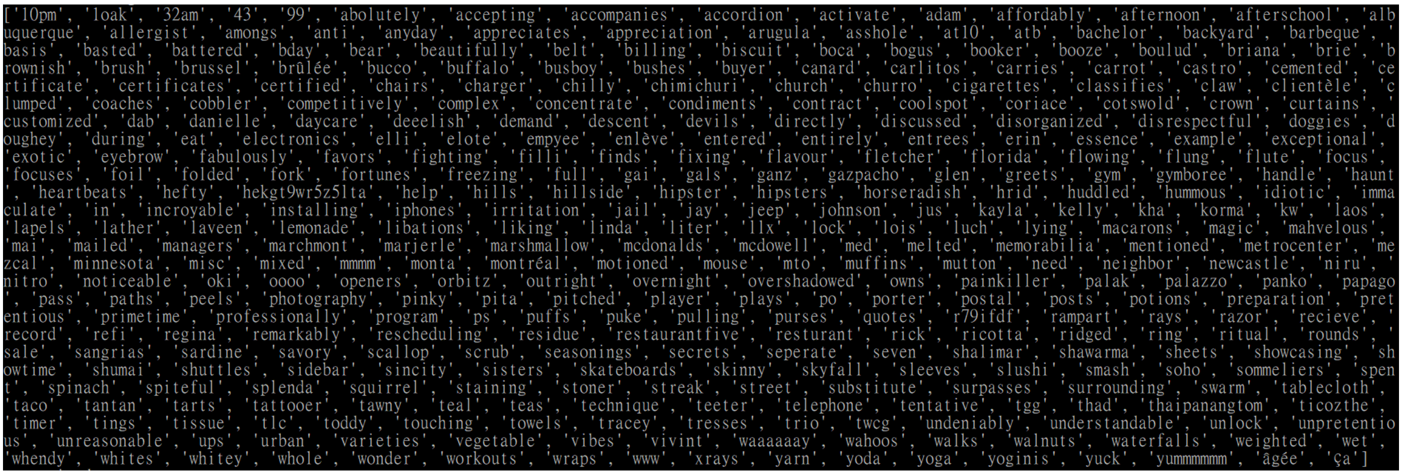
accuracy of sentence with top 15 highest coefficient words: 0.837037



accuracy of sentence with top 20 highest coefficient words: 0.836601



from the result of the stop words selection we can see that it’s only when k is above 3000 when the accuracy on dev set starts to improve, when k is too low too many words are set to stop words for instance when k=500 there are 9882 – 500*2 = 8882 stop words and it’s not surprising that the accuracy is low in this case, and at high k values the accuracy improves only slightly, for example these are the stop words when k =4750



```
['10pm', '1oak', '32am', '43', '99', 'abolutely', 'accepting', 'accompanies', 'accordion', 'activate', 'adam', 'affordably', 'afternoon', 'afterschool', 'alb
uquerque', 'allergist', 'amongs', 'anti', 'anyday', 'appreciates', 'appreciation', 'arugula', 'asshole', 'at10', 'atb', 'bachelor', 'backyard', 'barbeque', '
basis', 'basted', 'battered', 'bday', 'bear', 'beautifully', 'belt', 'billing', 'biscuit', 'boca', 'bogus', 'booker', 'booze', 'boulud', 'briana', 'brie', 'b
rownish', 'brush', 'brussel', 'brûlée', 'bucco', 'buffalo', 'busboy', 'bushes', 'buyer', 'canard', 'carlitos', 'carries', 'carrot', 'castro', 'cemented', 'ce
rtificate', 'certificates', 'certified', 'chairs', 'charger', 'chilly', 'chimichuri', 'church', 'churro', 'cigarettes', 'classifies', 'claw', 'clientèle', 'c
lumped', 'coaches', 'cobbler', 'competitively', 'complex', 'concentrate', 'condiments', 'contract', 'coolspot', 'coriace', 'cotswold', 'crown', 'curtains', '
customized', 'dab', 'danielle', 'daycare', 'deeelish', 'demand', 'descent', 'devils', 'directly', 'discussed', 'disorganized', 'disrespectful', 'doggies', 'd
oughey', 'during', 'eat', 'electronics', 'elli', 'elote', 'empyee', 'enlève', 'entered', 'entirely', 'entrees', 'erin', 'essence', 'example', 'exceptional',
'exotic', 'eyebrow', 'fabulously', 'favors', 'fighting', 'filli', 'finds', 'fixing', 'flavour', 'fletcher', 'florida', 'flowing', 'flung', 'flute', 'focus',
'focuses', 'foil', 'folded', 'fork', 'fortunes', 'freezing', 'full', 'gai', 'gals', 'ganz', 'gazpacho', 'glen', 'greets', 'gym', 'gymboree', 'handle', 'haunt
', 'heartbeats', 'hefty', 'hekgt9wr5z5lta', 'help', 'hills', 'hillside', 'hipster', 'hipsters', 'horseradish', 'hrid', 'huddled', 'hummous', 'idiotic', 'imma
culate', 'in', 'incroyable', 'installing', 'iphones', 'irritation', 'jail', 'jay', 'jeep', 'johnson', 'jus', 'kayla', 'kelly', 'kha', 'korma', 'kw', 'laos',
'lapels', 'lather', 'laveen', 'lemonade', 'libations', 'liking', 'linda', 'liter', 'llx', 'lock', 'lois', 'luch', 'lying', 'macarons', 'magic', 'mahvelous',
'mai', 'mailed', 'managers', 'marchmont', 'marjerle', 'marshmallow', 'mcdonalds', 'mcdowell', 'med', 'melted', 'memorabilia', 'mentioned', 'metrocenter', 'me
zcal', 'minnesota', 'misc', 'mixed', 'mmmm', 'monta', 'montréal', 'motioned', 'mouse', 'mto', 'muffins', 'mutton', 'need', 'neighbor', 'newcastle', 'niru', '
nitro', 'noticeable', 'oki', 'oooo', 'openers', 'orbitz', 'outright', 'overnight', 'overshadowed', 'owns', 'painkiller', 'palak', 'palazzo', 'panko', 'papago
', 'pass', 'paths', 'peels', 'photography', 'pinky', 'pita', 'pitched', 'player', 'plays', 'po', 'porter', 'postal', 'posts', 'potions', 'preparation', 'pret
entious', 'primetime', 'professionally', 'program', 'ps', 'puffs', 'puke', 'pulling', 'purses', 'quotes', 'r79ifdf', 'rampart', 'rays', 'razor', 'recieve', '
record', 'refi', 'regina', 'remarkably', 'rescheduling', 'residue', 'restaurantfive', 'resturant', 'rick', 'ricotta', 'ridged', 'ring', 'ritual', 'rounds', '
sale', 'sangrias', 'sardine', 'savory', 'scallop', 'scrub', 'seasonings', 'secrets', 'seperate', 'seven', 'shalimar', 'shawarma', 'sheets', 'showcasing', 'sh
owtime', 'shumai', 'shuttles', 'sidebar', 'sincity', 'sisters', 'skateboards', 'skinny', 'skyfall', 'sleeves', 'slushi', 'smash', 'soho', 'sommeliers', 'spen
t', 'spinach', 'spiteful', 'splenda', 'squirrel', 'staining', 'stoner', 'streak', 'street', 'substitute', 'surpasses', 'surrounding', 'swarm', 'tablecloth',
'taco', 'tantan', 'tarts', 'tattooer', 'tawny', 'teal', 'teas', 'technique', 'teeter', 'telephone', 'tentative', 'tgg', 'thad', 'thaipanangtom', 'ticozthe',
'timer', 'tings', 'tissue', 'tlc', 'toddy', 'touching', 'towels', 'tracey', 'tresses', 'trio', 'twcg', 'undeniably', 'understandable', 'unlock', 'unpretentio
us', 'unreasonable', 'ups', 'urban', 'varieties', 'vegetable', 'vibes', 'vivint', 'waaaaaay', 'wahoos', 'walks', 'walnuts', 'waterfalls', 'weighted', 'wet',
'whendy', 'whites', 'whitey', 'whole', 'wonder', 'workouts', 'wraps', 'www', 'xrays', 'yarn', 'yoda', 'yoga', 'yoginis', 'yuck', 'yummmmmm', 'âgée', 'ça']
```

we can see that these words have little correspondence with negative or positive meaning, so by setting these words to stop words the and increase the accuracy seems reasonable, at k = 3750 the accuracy of the dev set increases the most though still not very much. The hyper parameter stop_words only turn the selected list of words into stop words which is not the same as selecting features so the improvement has its limits

TfidfVectorizer:

as opposed to CountVectorizer which convert the collection of text into a matrix of
Convert a collection of text documents to a matrix of token counts, the TfidVectorizer converts the collection of text documents to a matrix of TF-IDF features (frequency rather than count) or term-frequency times inverse document-frequency. The goal of using tf-idf instead of the raw frequencies of occurrence of a token in a given document is to scale down the impact of tokens that occur very frequently in a given corpus and that are hence empirically less informative than features that occur in a small fraction of the training corpus.
by using TfidfTransformer we can transform count matrix produce by CountVectorize into TfidfVectorizer
However, the result didn't improve.

	CountVectorizer	TfidVectorizer
accuracy on dev set	0.777292	0.762008

It's possible that countvectorizer out performed tf-idf in this case, it's also possible that in this case some common words(words with high frequency) are helpful in distinguishing positive and negative
It may be that common words (words which will appear in multiple documents) are helpful in distinguishing between classes. Some words like pronouns are very common and would be down weighted in tf-idf, but given equal weight to rare words in countvectorizer.

max_df:

ignores words having high frequency, the vocabs frequency higher than the threshold are ignored at training
the usage of max_df is to ignore common words that has little effect on the classification

max_df	0.1	0.2	0.3	0.4	0.5
Accuracy on dev	0.766375	0.783842	0.788209	0.790393	0.783842

max_df	0.6	0.7	0.8	0.9	1 default
Accuracy on dev	0.783842	0.781659	0.777292	0.777292	0.777292

we can see that max_df = 0.4 has the highest accuracy, when max_df is too high, not much words are ignored so it's reasonable that the accuracy doesn't change, whereas when max_df is too low, too many words are ignoreds and the accuracy declines

min_df:
ignores terms that have a document frequency strictly lower than the given threshold. This value is also called cut-off in the literature, the value of min_df means the lowest count threshold

min_df	1 default	2	3	4	
Accuracy on dev	0.777292	0.777292	0.764192	0.759825	0.753275

min_df	6	7	8	9	10
Accuracy on dev	0.755458	0.751091	0.753275	0.759825	0.751091

we can see that when min_df is low at 1,2 the accuracy is the same as the baseline, since only a few words are ignored the results are the same, when min_df increases we can see that the accuracy is well below the baseline, it's possible that the words that has great influence on the prediction are ignored since these words doesn't appear a lot, such as awesome, delicious, amazing, they don't appear a lot but it's apparent that if these words appear in a sentence the prediction will be positive.

ngram_range:
The lower and upper boundary of the range of n-values for different n-grams to be extracted. All values of n such that min_n <= n <= max_n will be used

ngram_range	(1,1) default	(1,2)	(1,3)	(2,3)
accuracy on dev	0.777292	0.783842	0.779475	0.733624

ngram_range	(3,4)	(1,4)	(2,4)	(3,3)
accuracy on dev	0.670305	0.775109	0.713973	0.652838

for ngram_range selection we see that unigram along with bigram has the highest accuracy, for onl y unigram the accuracy are low since it doesn't consider context, but when we use unigram along with bigram and trigram the accuracy declines, this might be the result of overfitting, since for clas sifying a sentence it might be that the appearance of strong positive or negative words is more important than the context in classification for instance in the training set there is this sentence: "*Dr. Greenberg is attentive, caring and listened to all of my concerns. He made me feel comfortable before my laser procedure and calmed my nerves. I recommend him to all*" when our model uses only unigram and bigram it sees *calmed, calmed my* but when it uses trigram as well it sees *calmed my nerves* the word nerves might miss-guide the model

for supervised learning we eventually select min_df = 1, max_df = 0.31, ngram_range = (1,2), stop_words with k = 3750, we get accuracy on dev set equals 0.792576, and accuracy on kaggle equals 0.78923

Semi-supervised: Exploiting Unlabeled Data:

after training the supervised classifier, we use the trained logistic regression model to predict the unlabeled data, for *selection* of what data to add to the labeled train data I use the confidence of the prediction. Confidence here is defined as the absolute value of difference between the negative and positive probability, we first sort the unlabeled data according to the confidence from in reverse order and pick the highest batch amount of data as the data to add, we also excluded these data from the unlabeled data set to avoid picking adding the same data, batch size is the amount of sentences we want to add to the labeled data,
every time after adding the new data we retrained the model and use the newly trained model to predict the rest of unlabeled data,
for stop criterion I just use the fix number of iterations, or the percentage of the unlabeled data

for batch size = 500 sentences and add 10 times retrain ten times the accuracy of the dev set is 0.792576 and the kaggle submission is 0.78948 which has slight improvement compare with supervised training

batch size = 500

Iteration	5	10	15	20	25
Accuracy of dev	0.792576	0.792576	0.794759	0.794759	0.792576
Accuracy of test	0.78948	0.78948	0.78911	0.78850	0.78861

the size of training data increases with 500 sentences for every iteration, it seems that the accuracy of the test set will eventually decline, it might be because that the more uncertain data you add the less accurate the model trains, since our classifier of supervise learning is not 100%, most likely even when we use the most confident predictions a lot of the data are predicted incorrectly so even though we add some test data accuracy will go up at the beginning it will eventually decline due to the fact that the supervised classifier's accuracy cannot be 100%

['00', '30', '45', 'accommodating', 'actually', 'agree', 'all', 'all around', 'all my', 'also have', 'always', 'always friendly', 'always great', 'amazing', 'amazing and', 'amazing experience', 'an hour', 'and also', 'and awesome', 'and both', 'and delicious', 'and friendly', 'and great', 'and have', 'and helpful', 'and reasonably', 'and service',

```
'and such', 'and the', 'another', 'anywhere', 'around', 'asked', 'at the', 'atmosphere', 'atmosphere is', 'attentive',
'attentive and', 'authentic italian', 'average', 'awesome', 'awful', 'back', 'bad', 'bbq', 'be back', 'beautiful
handbags', 'because', 'because the', 'beer selection', 'best', 'best and', 'best buffet', 'best couture', 'best
mexican', 'better than', 'beyond', 'bf and', 'bland', 'buffet', 'burrito', 'but', 'but not', 'but that', 'but the', 'but to',
'but was', 'by', 'can', 'can be', 'cannot', 'care', 'chef', 'chicago', 'cocktails', 'coffee', 'coffee is', 'cold', 'cold and',
'combo', 'company', 'cool', 'could', 'could give', 'couture', 'couture ever', 'curry was', 'customer', 'customer
service', 'dark', 'deals', 'dealsfriendly', 'def', 'definitely', 'delicious', 'delicious and', 'delicious food', 'delicious
the', 'deliver', 'desk', 'did', 'did not', 'die', 'dirty', 'disappointed', 'do not', 'dogs', 'downtown', 'drink', 'driver',
'employees', 'ended', 'enjoyed the', 'enough', 'especially', 'even have', 'ever', 'ever to', 'ever was', 'everyone',
'everyone is', 'everything', 'excellent', 'excellent and', 'expensive', 'experience', 'experience ever', 'experience
from', 'extremely', 'fairly', 'fantastic', 'fast', 'favorite', 'favorite is', 'favorite place', 'feels', 'find', 'fish', 'flavor',
'food and', 'food great', 'food perfect', 'food the', 'food was', 'fresh', 'friendly', 'friendly and', 'friendly service',
'friendly staff', 'front', 'front desk', 'fun', 'gel', 'generous', 'get your', 'give', 'gluten free', 'good', 'good
dealsfriendly', 'good sales', 'good service', 'gotta', 'great', 'great and', 'great atmosphere', 'great however',
'great little', 'great on', 'great place', 'great service', 'great spot', 'great staff', 'great time', 'great variety',
'greeted', 'greeted nicely', 'groupon', 'handbags great', 'happy', 'have no', 'he', 'helpful', 'helpful the', 'her',
'highly', 'highly recommend', 'hit', 'horrible', 'horrible customer', 'horrible food', 'horrible horrible', 'horrible
service', 'however', 'impressed by', 'ingredients', 'is amazing', 'is awesome', 'is by', 'is excellent', 'is
extremely', 'is fantastic', 'is great', 'is horrible', 'is my', 'is ok', 'is really', 'is so', 'is the', 'is very', 'italian las',
'know', 'las', 'last night', 'least', 'leave', 'list', 'little', 'lives', 'location', 'love', 'love love', 'love this', 'loved',
'lunch', 'make sure', 'manager', 'meal', 'menu', 'mexican', 'midnight', 'mins', 'minutes', 'mistake', 'moment step',
'money', 'most authentic', 'much good', 'must', 'my favorite', 'nail', 'never go', 'new york', 'nice to', 'nicely',
'nicely until', 'not', 'not bad', 'not go', 'not the', 'of beer', 'of food', 'oh', 'ok', 'online', 'or', 'or the', 'order',
'ordered', 'ordering', 'other', 'our food', 'our order', 'outstanding', 'owned', 'patio', 'pizza', 'place', 'place great',
'place have', 'place was', 'polish', 'poor', 'pork', 'potato', 'price', 'professional', 'pulled', 'pumpkin', 'purchase',
'put', 'quality', 'quick', 're', 'really', 'really good', 'reason', 'recommend', 'relaxing', 'restaurants', 'right', 'right
away', 'roll', 'rolls', 'rude', 'rude and', 'salad', 'sales beautiful', 'sandwiches', 'sauce', 'sauces', 'saw', 'seating',
'selection', 'self', 'seriously', 'service', 'service and', 'service great', 'service have', 'service is', 'service was',
'she', 'shop was', 'should', 'since was', 'single', 'slow', 'so friendly', 'some', 'some good', 'somewhat', 'soup',
'spices curry', 'spot', 'staff', 'staff friendly', 'staff great', 'staff is', 'step inside', 'straight', 'style', 'sunday', 'super',
'super nice', 'sure', 'sweet', 'tacos', 'taken', 'taste', 'tasted', 'tasty', 'teens', 'terrible', 'terrible service', 'terrible
terrible', 'thai', 'thanks', 'that great', 'that is', 'the and', 'the atmosphere', 'the best', 'the big', 'the front', 'the girl',
'the great', 'the hood', 'the hotel', 'the inside', 'the kids', 'the manager', 'the moment', 'the place', 'the selection',
'the shop', 'the waitress', 'the worst', 'then', 'there the', 'there were', 'they', 'they also', 'they charge', 'they re',
'they use', 'they would', 'this is', 'this little', 'this location', 'this place', 'this store', 'thought was', 'time tonight',
'times', 'times and', 'times the', 'to after', 'to be', 'to die', 'to get', 'to love', 'to so', 'to wait', 'to write', 'told', 'too',
'too much', 'town', 'town the', 'true', 'try', 'twice', 'use', 'used to', 'usually', 'variety', 've', 've the', 'venue', 'very',
'very friendly', 'very good', 'waited', 'waitress was', 'walked', 'want to', 'was amazing', 'was awful', 'was
excellent', 'was friendly', 'was great', 'was greeted', 'was not', 'was rude', 'was so', 'was terrible', 'was which',
'waste', 'waste your', 'went', 'were great', 'when', 'which of', 'will', 'will definitely', 'within', 'work', 'worst',
'worst customer', 'worst experience', 'worst service', 'would', 'would be', 'would do', 'write', 'wrong', 'york']
```

these are the coefficients that changes the most, (the coefficients that changes above 0.005) from my observation it seems that these words have strong meanings such as *love, rude , very good*…and are the words that will affect the classifier's prediction the most, seems that the classifier is always adjusting and thus the words that matters the most change the coefficients the most and it is apparent from out test set accuracy that these coefficients do matter