CSE 156 Project-Report

Minwoo Yoo, Jiajin Liang, Tianheng Liu, Dian Yu

TOTAL POINTS

102 / 100

QUESTION 1

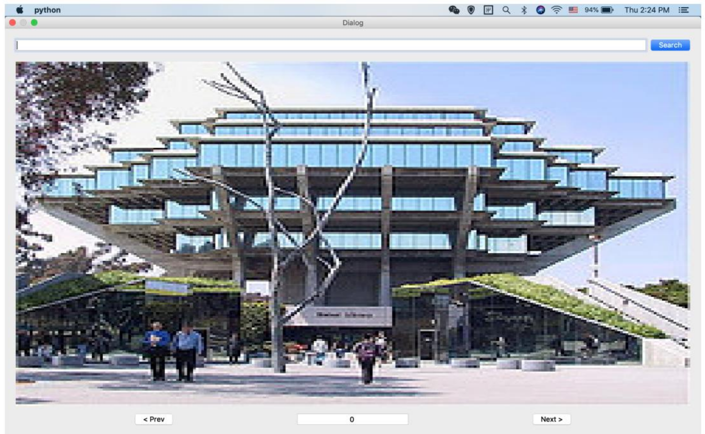
1 102 / 100

✓ **- 0 pts** Correct

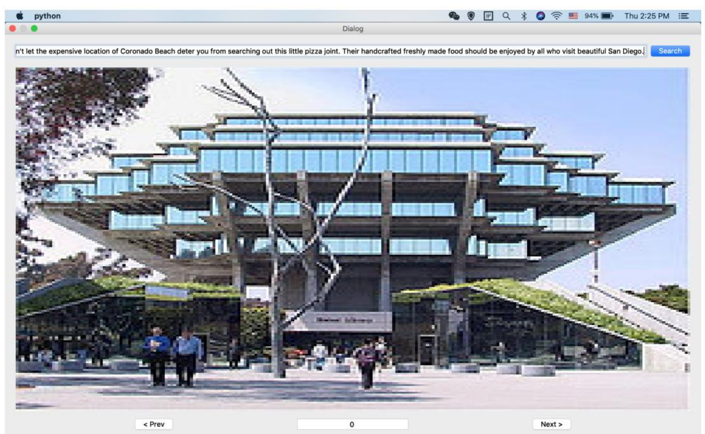
+ 2 Point adjustment

CSE 156 Final Project Report (Team 1)

Application Setup Page (see README to run the application):

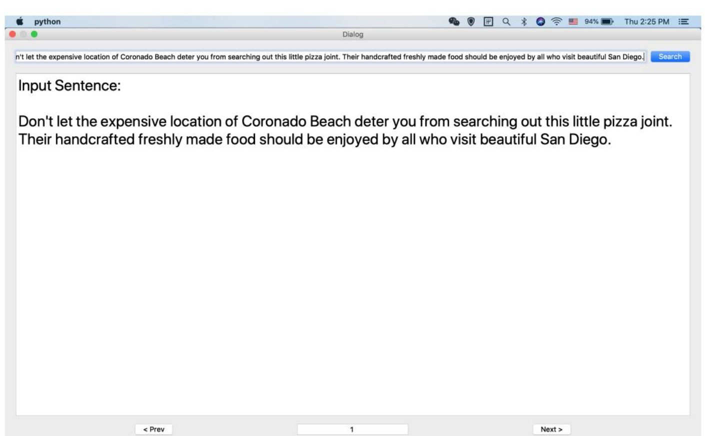


How to use:
1. Input review sentence on the top bar and click search button (it will take several seconds to generate the outputs).
2. The result will be shown by click Next button at the bottom of the application.

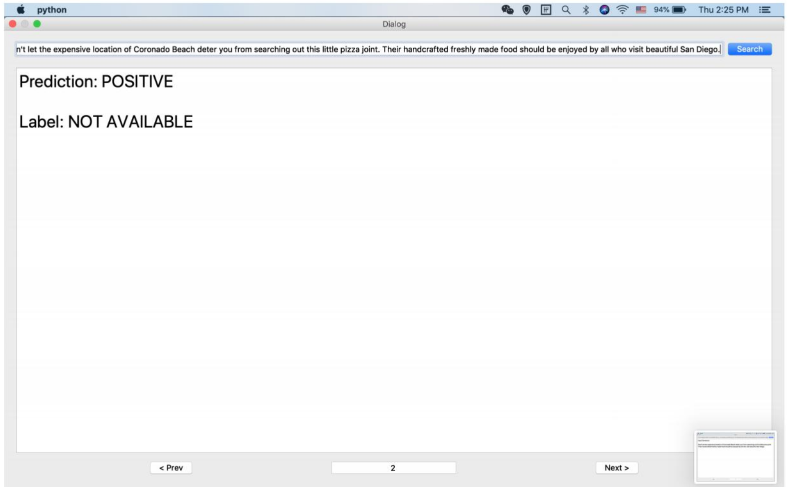


Part 1

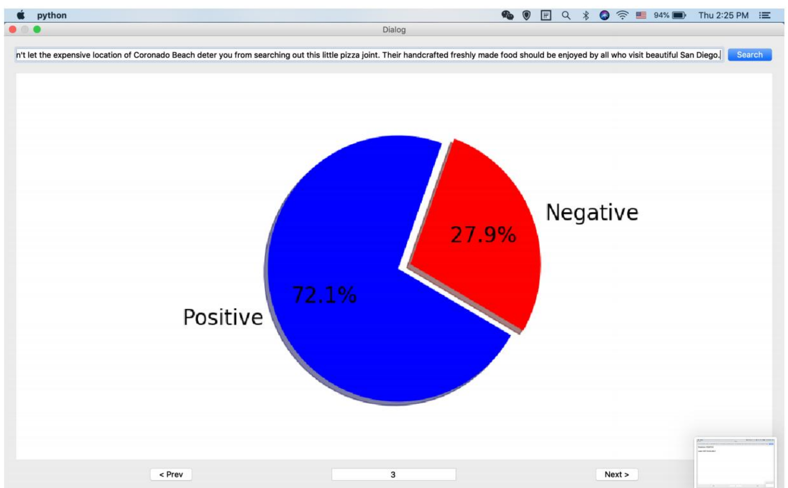
1.1 Favorite Review: (From Yelp, it is a 5 star review, thus a positive review)



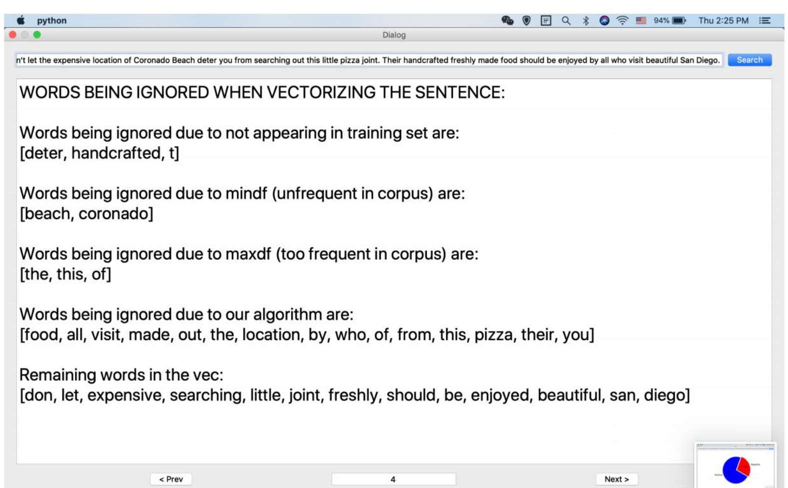
Prediction:



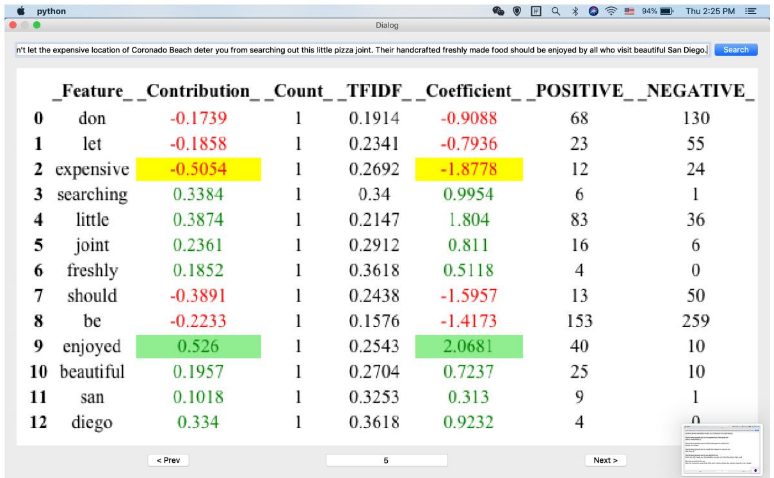
Pie Chart for Probabilities of Positive and Negative Labels:



Words Ignored by the algorithm and TFIDF Vectorizer:

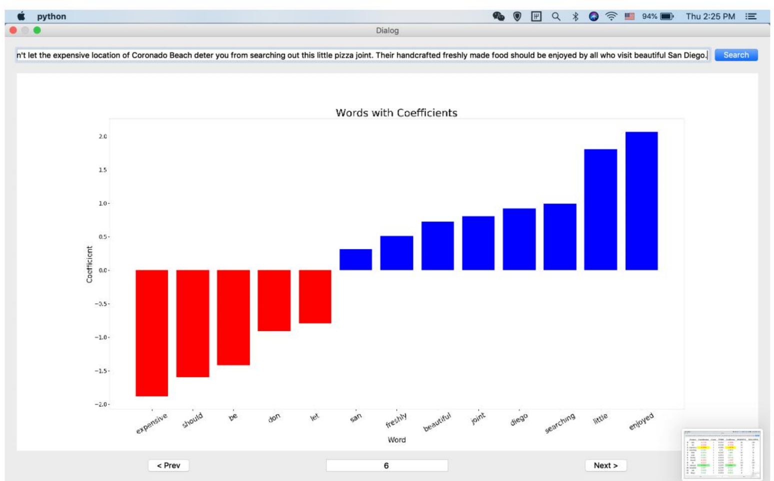


This is the information chart from remaining features (words), including numbers of occurrence of positive and negative labeled data from training set, coefficients from Logistic Regressor, TFIDF values from vectorization, counts in input sentence, and contributions towards labels. Red numbers indicate a word related to negative review and green numbers indicate a word related to positive review. Highlighted numbers are the most contributed words.

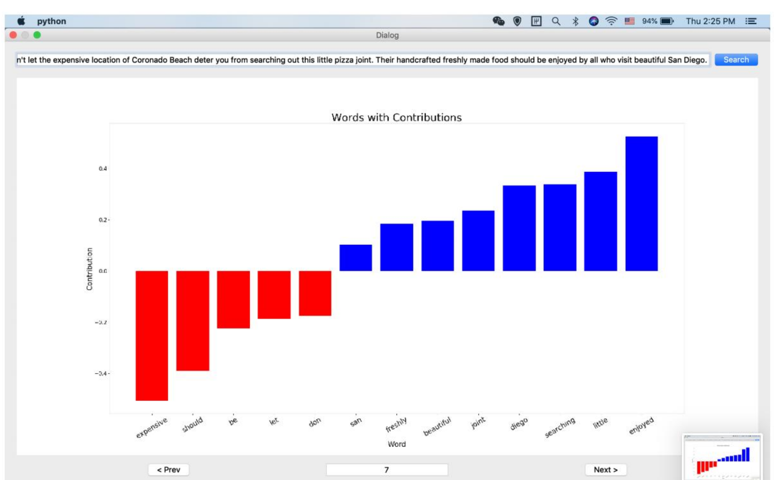


	Feature	_Contribution_	_Count_	_TFIDF_	_Coefficient_	_POSITIVE_	_NEGATIVE_
0	don	-0.1739	1	0.1914	-0.9088	68	130
1	let	-0.1858	1	0.2341	-0.7936	23	55
2	expensive	-0.5054	1	0.2692	-1.8778	12	24
3	searching	0.3384	1	0.34	0.9954	6	1
4	little	0.3874	1	0.2147	1.804	83	36
5	joint	0.2361	1	0.2912	0.811	16	6
6	freshly	0.1852	1	0.3618	0.5118	4	0
7	should	-0.3891	1	0.2438	-1.5957	13	50
8	be	-0.2233	1	0.1576	-1.4173	153	259
9	enjoyed	0.526	1	0.2543	2.0681	40	10
10	beautiful	0.1957	1	0.2704	0.7237	25	10
11	san	0.1018	1	0.3253	0.313	9	1
12	diego	0.334	1	0.3618	0.9232	4	0

Bar Chart of Coefficients from Different Words:



Bar Chart of Contributions from Different Words:



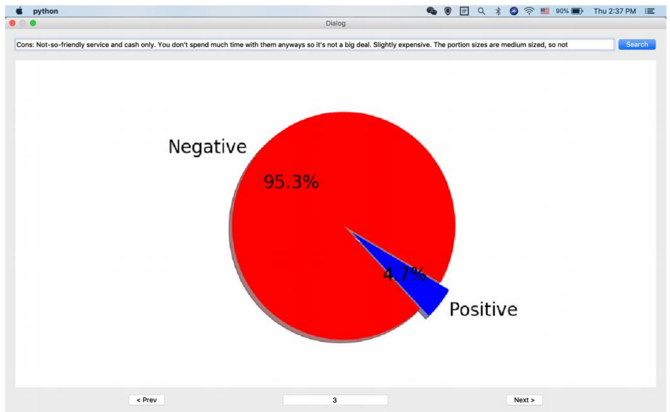
Evaluation: as we can see from the above charts, the contributions from positively related words are more than that of negatively related words. From Logistic Regression, probability is directly related to the sum of contribution. Hence, the probability of being positive review is higher than being negative review.

1.2 Overconfident Example

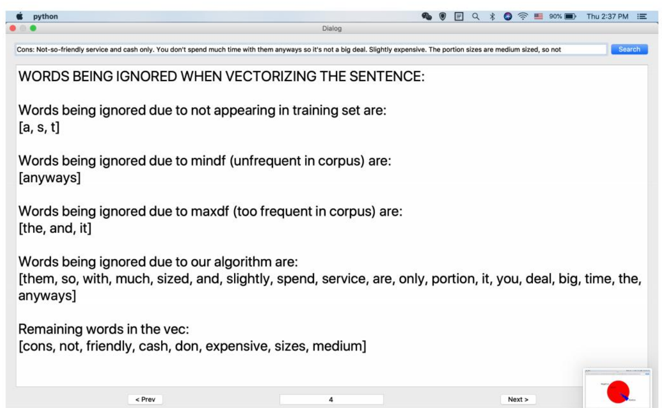
Review (from dev set):

```
POSITIVE	Cons: Not-so-friendly service and cash only. You don't spend much time
with them anyways so it's not a big deal. Slightly expensive. The portion sizes are
medium sized, so not
```

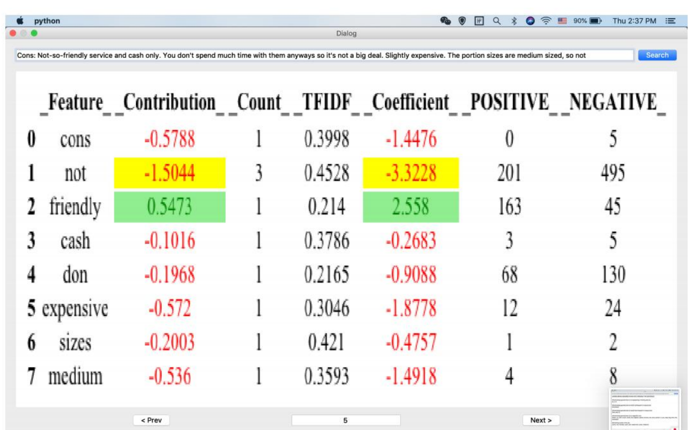
Our Prediction:



Vectorization Process:

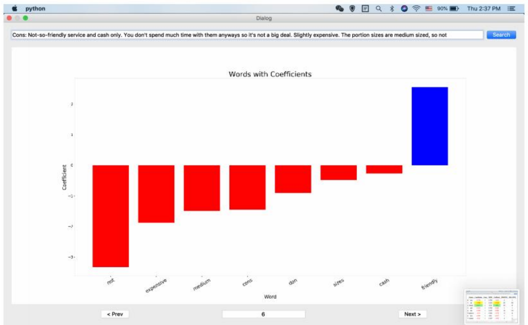


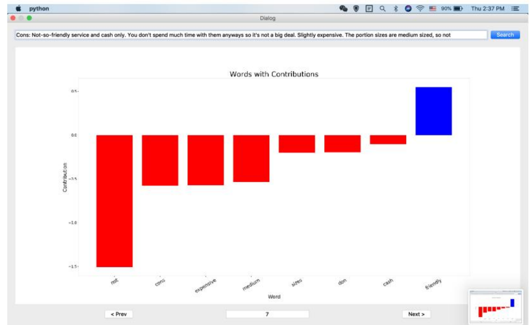
Features Info Chart:



	Feature	Contribution	Count	TFIDF	Coefficient	POSITIVE	NEGATIVE
0	cons	-0.5788	1	0.3998	-1.4476	0	5
1	not	-1.5044	3	0.4528	-3.3228	201	495
2	friendly	0.5473	1	0.214	2.558	163	45
3	cash	-0.1016	1	0.3786	-0.2683	3	5
4	don	-0.1968	1	0.2165	-0.9088	68	130
5	expensive	-0.572	1	0.3046	-1.8778	12	24
6	sizes	-0.2003	1	0.421	-0.4757	1	2
7	medium	-0.536	1	0.3593	-1.4918	4	8

Coefficient and Contribution Bar Charts:





Explanation:
As we can see from above charts, there are much more negatively related words than positively related words remaining from our vectorized step. Thus, it predicts “negative” with over 90% confidence. One possible reason is that our vectorizer possibly eliminates some positively related words due to appearing too often or too rare. Another possible reason is that this data is not properly labeled since it only contains a part of positive review but with a lot of negative words. We believe this review also contains a part of “Pros” but is cut due to some preprocessing reasons.

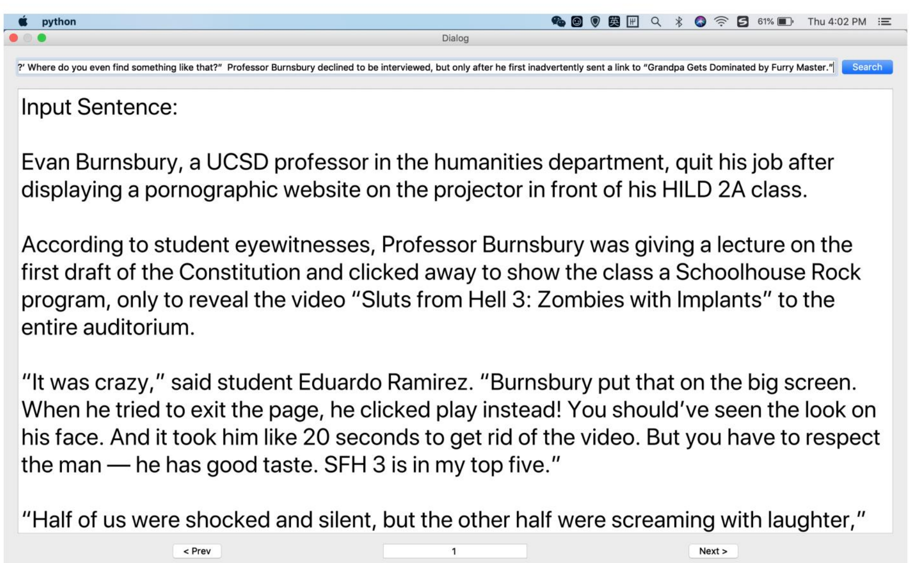
Part 2

2.1 Description

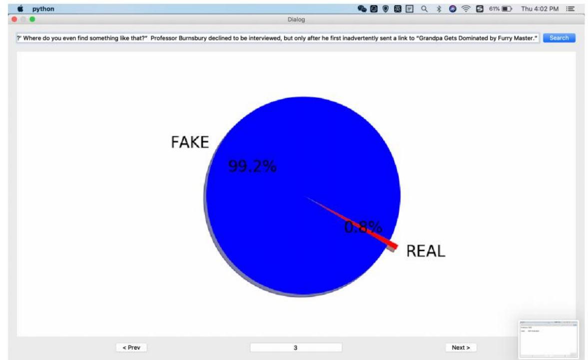
Classification: Predict if a news is FAKE or REAL using Logistic Regression similar to Part 1.
Data: Each data is a news paragraph contains hundreds of words and the two labels are FAKE and REAL.

2.2 Favorite Example

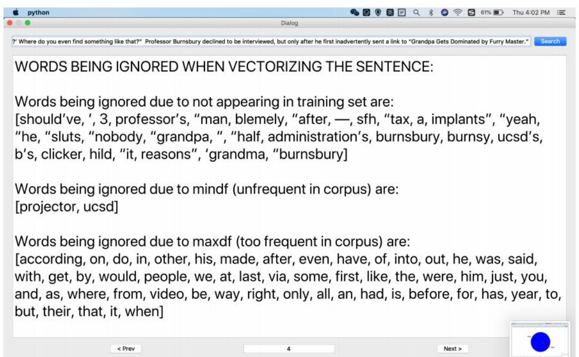
Input news: a FAKE news of UCSD professor showing porn in class

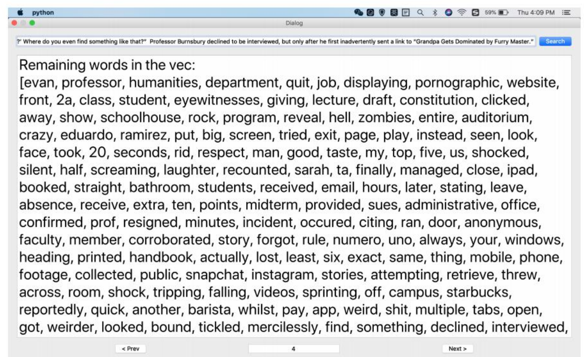


Prediction:

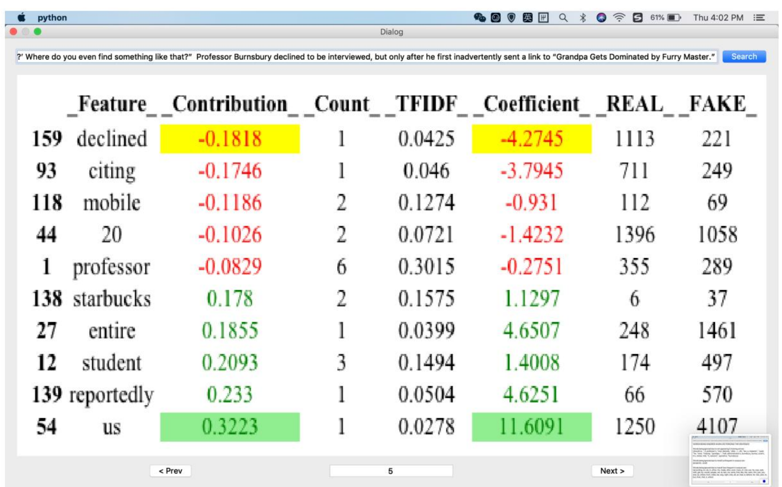


Vectorization:





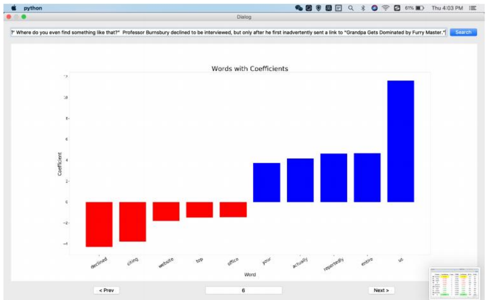
Features Info Chart:

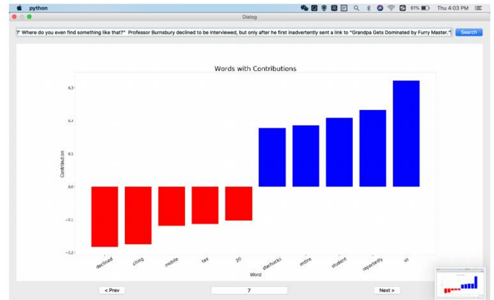


	Feature	_Contribution_	_Count_	_TFIDF_	_Coefficient_	_REAL_	_FAKE_
159	declined	-0.1818	1	0.0425	-4.2745	1113	221
93	citing	-0.1746	1	0.046	-3.7945	711	249
118	mobile	-0.1186	2	0.1274	-0.931	112	69
44	20	-0.1026	2	0.0721	-1.4232	1396	1058
1	professor	-0.0829	6	0.3015	-0.2751	355	289
138	starbucks	0.178	2	0.1575	1.1297	6	37
27	entire	0.1855	1	0.0399	4.6507	248	1461
12	student	0.2093	3	0.1494	1.4008	174	497
139	reportedly	0.233	1	0.0504	4.6251	66	570
54	us	0.3223	1	0.0278	11.6091	1250	4107

Explanation: words with green values are prone to appear in FAKE news while words with red values are prone to appear in REAL news. Highlighted values are most contributed ones. Due to larger data than part 1, we only show top 5 most FAKE related words and top 5 most REAL related words here.

Coefficient and Contribution Bar Charts:





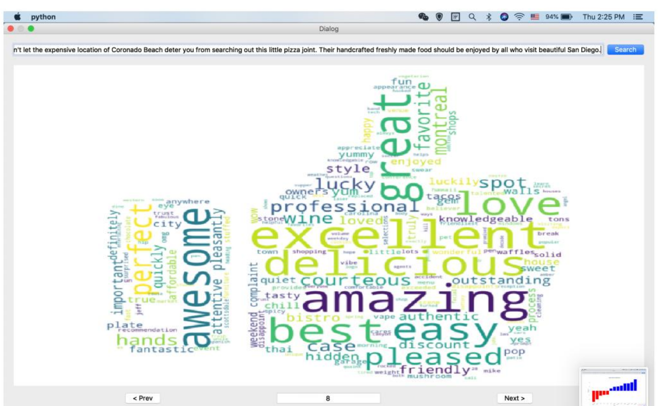
Explanation: similar to Part 1, since there are more FAKE related words appearing in the sample, the total contributions from FAKE related words are much higher than that from REAL related words. From Logistic Regression, probability is directly related to the sum of contribution. Thus, it predicts FAKE.

Part 3 Creativity

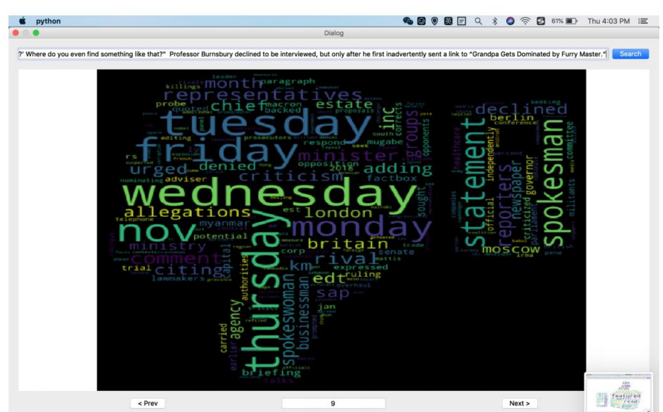
Word Clouds

We generate word clouds based on the coefficients from Logistic Regressors. The larger the word appears, the higher coefficient it has. The word cloud pictures can be seen after all the explanation pages.

Part 1 Positive Review Word Cloud and Negative Review Word Cloud

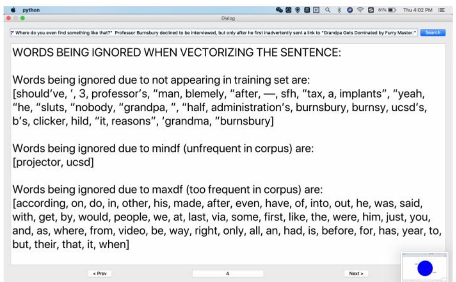


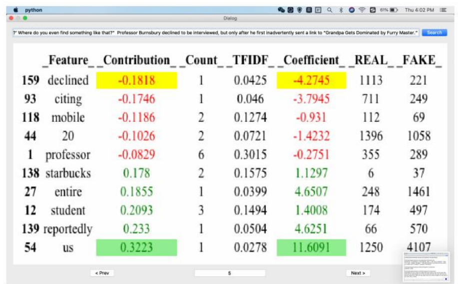
Part 2 FAKE News Word Cloud and REAL News Word Cloud



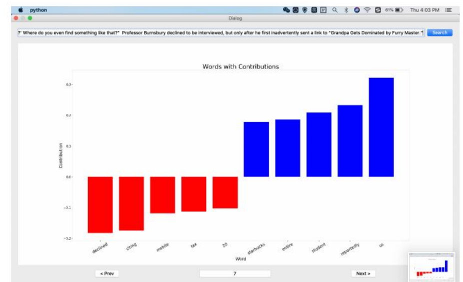
Interactive Application with User-Friendly Interface

We've designed and implemented a Python Application using PyQt5 from scratch. Users can input their own examples into the search bar to generate all the explanations. Once the application is launched, the processing time of generating results is within seconds depending on the length of the input. The interface is easy to use and instructive for user to look through the outputs our program generates. Our explanation pages combine words, numbers, and charts together to comprehensively illustrate how our model makes such prediction from Feature Selection to Feature Analysis to Prediction Explanation.





	Feature	_Contribution_	_Count_	_TFIDF_	_Coefficient_	_REAL_	_FAKE_
159	declined	-0.1818	1	0.0425	-4.2745	1113	221
93	citing	-0.1746	1	0.046	-3.7945	711	249
118	mobile	-0.1186	2	0.1274	-0.931	112	69
44	20	-0.1026	2	0.0721	-1.4232	1396	1058
1	professor	-0.0829	6	0.3015	-0.2751	355	289
138	starbucks	0.178	2	0.1575	1.1297	6	37
27	entire	0.1855	1	0.0399	4.6507	248	1461
12	student	0.2093	3	0.1494	1.4008	174	497
139	reportedly	0.233	1	0.0504	4.6251	66	570
54	us	0.3223	1	0.0278	11.6091	1250	4107



Dive Deep into How Model Makes Prediction

We've implemented our own version of how Logistic Regressor makes prediction once the coefficient set is trained from training data. Functions such as calculating contribution from each feature (word) and calculating probabilities of both labels are re-implemented from scratch rather than simply calling the built-in functions in Logistic Regressor Library. In this way, we better understand the prediction process and how each feature contributes to the predicted label.

```
def prob(self, x):

    z = self.intercept
    for i,v in zip(x.indices, x.data):
        z += self.cls.coef_[0][i] * v

    pos = 1 / (1 + math.exp(-z))
    neg = 1 - pos

    return [neg,pos]
```

```
def tfidf_x_coef(self, x):

    cd = {}
    for i,v in zip(x.indices, x.data):
        cd[self.cv[i]] = self.cls.coef_[0][i] * v

    return cd
```

1 **102** / **100**

✓ **- 0 pts** Correct

+ 2 **Point adjustment**