Sana Khan

Homework 4 - Naïve Bayes

Introduction

With the rise of the internet, especially social media, people now have vast amounts of information at their fingertips. It's also very common, and solicited by businesses, to leave reviews for different products or services received. People then use this information on a daily basis to make a decision on what product(s) to buy, what restaurant to eat at or who to hire for a specific job. With so much relying on reviews, fake reviews can damage a business or persons reputation. From the businesses standpoint, you could lose customers if you have fake negative reviews, which can result in a loss of revenue or if it gets serious enough, closing your business. . Managing the fake reviews can also be time consuming and costly as a business owner. From a customer's standpoint, you could lose your trust in a business if you're noticing either all positive reviews or negative reviews. This could result in not purchasing the product you're viewing, and a feeling frustrated with the brand.

Being able to write fake reviews is becoming more common thanks to A.I. and Large Language Model bots (LLM). It's also becoming increasingly difficult for humans to prove a determine if a review is fake thanks to the sophistication of these bots. We can however use models to help us better detect fake reviews or help categorize reviews based on their sentiment. We can train the model to identify key words that are used in fake/negative reviews and use that knowledge to determine if a sample review is real or fake and determine it's sentiment. By doing so, we can ensure the integrity of reviews and instill confidence in customers.

Our task is to use a subset of reviews to train our model to detect fake reviews vs real reviews and also the sentiment of the review. We will then use the trained model on the rest of the reviews to understand how well the model was able to predict for each category.

Analysis

Data Prep

The data was provided in a CSV file, with 3 columns total. The first column shows if the review is a lie with the labels f (fake) or t (true). The second column is the sentiment of the review with n (negative) or p (positive). The third column is the text of the review, which is split over multiple columns due to a formatting error.

lie	sentiment	review									
f	n	'Mike\'s Pizz	NY Service v	not. Stick to pre-made dishes like stuffed pasta or a salad. You should consider dining else wh							
f	n	'i really like t	japanese	and chinese dishes. we also got a free drink and free refill. there are also different kinds of d							
f	n	'After I went	we went to DODO restaurant for dinner. I found worm in one of the dishes .'								
f	n	'Olive Oil Ga	and the waitor had no manners whatsoever. Don\'t go to the Olive Oil Garden. '								
f	n	'The Seven H	never more. '								
f	n	'I went to XY	she rudely r	I had a bad	otherwise	my favorite place to dine. '					
f	n	'I went to AB	our food car	especially if you\'re in a hurry!'							
f	n	'I went to th	and then we	at which point we just paid and left. Don\'t go.'							
f	n	'OMG. This r	we just stoo	the crust of a pizza was so hard like plastics. My dog didn\'t even eat that pizza. I hate this pl							
f	n	'Yesterday	I went to a	I thought its	when I orde	instead of s	the attenda	the casino d	the more they will loose. S		
f	n	'Last weeker	and I wasn\'	but it seems	but I may come back for a quick bite of their shockingly delicious Cheesecake. '						
f	n	'I once went	I found that	there is no sauce! No any kind of sauce! I can\'t believe my eyes! I will never go to the Chipot							
f	n	'I entered th	said coldly	help yoursel	I waited and	but no one e	I finally had the most terrible food in my life and ever				
f	n	'Carlo\'s Plat	the staff wa	I had a sick	and my frier	but bland	iced tea (the	while my fri	a good 15 m	and was ext	
f	n	'This is the la	but the taste	the place ha	but they hav	nobody cam	and when th	we had to ca	and they we	and they we	
f	n	'In each of th	and when th	it could be a nightmare that I really want to wake up. The attitude of waiters is bad enough t							
f	n	'I went there	some drinks	a little bit to	but acceptable. My friends didn\'t like their lamb chop and cod filet that much.						
f	n	'I had heard	macaroni salad and a ravioli. The iced tea was too cold for my comfort and lacked flavor. Macaroni salad w								
f	n	'The service was way below average and we had to wait 10 minutes before we could be seated. Then had to wait 20 mi									
f	n	'This place ca	which serve	we went as	Matar Pane	Kadhai Chick	however the	it was on ou	which come	and we wer	
f	n	'Usually	I use Yelp to	I use it to se	and then it g	and then I w	he said that you can bought ham to do subway by you				
f	n	'I don\'t usua	but my expe	and when w	he said	\'well	no one\'s ev	per se	but it was m	which is NEVI	
f	n	'I recently at	the lettuce had black patches and the tomatoes looked horrible. Overall the burger was a disaster. When I								

Raw data image

To begin, the file was loaded and iterated over to create three new lists. One for the lie labels, another for the sentiment labels and the last containing the review. Then **CountVectorizer** from the scikit learn library was used to tokenize the words and put into a matrix. The matrix was then transformed to a data frame using pandas.

```
df_vector.head()
```

	10	15	20	2nd	30	5pm	6pm	abc	absolutely	acknowledge	...	worst	worth	wqr	wreck	write	written	xyz	yelp	yesterday	york
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 500 columns

Data frame of tokenized words

The **get_feature_names_out** method was used to identify the column names in order to remove columns that contain numbers or are tokens that do not add value. These columns were removed using a function that identified numbers and a column with a length of less than three letters. This resulted in a final data frame that was then split into two data frames, one for lies and the other for sentiment detection.

```
df_vector.head()
```

	absolutely	acknowledge	agreed	alfredo	amazing	ambiance	amer	america	american	appealing	...	world	worse	worst	worth	wreck	write	written
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	...	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0

Cleaned data frame

Both the lie and sentiment data frame had their respective list of labels added to each data frame.

```
dflies.head()
```

	LABEL	absolutely	acknowledge	agreed	alfredo	amazing	ambiance	amer	america	american	...	world	worse	worst	worth	wreck	write	written
0	f	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
1	f	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0
2	f	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
3	f	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
4	f	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0

5 rows × 448 columns

Lies data frame with labels

```
df_sentiment.head()
```

	LABEL	absolutely	acknowledge	agreed	alfredo	amazing	ambiance	amer	america	american	...	world	worse	worst	worth	wreck	write	written
0	n	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
1	n	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0
2	n	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
3	n	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
4	n	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0

5 rows × 448 columns

Sentiment data frame with labels

The data was then split into testing and training datasets for each data frame which was then used in the model.

Model/Method

The model that was used for this text classification was Naïve Bayes. Since the dataset was labeled, this is a supervised learning model. Naïve Bayes works by taking the probability that a feature (word) will appear in a category. It does so by taking the prior probability of each feature and the likelihood of observing each feature given a category. It also makes the assumption that all features are independent of each other.

Once the model is trained on the training data using the fit method, it is applied to the testing data using the predict method. The testing data has the labels removed, since that is what the model is trying to predict. It takes the probabilities from the testing data and uses that to predict which category the labels should appear in. We then compare the data results with the labels that were removed from the dataset . Then a confusion matrix is used to access the models performance. It reveals the number of true positives, true negatives, false positives, and

false negatives for each dataset. For the lie data set, we are trying to detect if the review if real or fake and for the sentiment dataset we are trying to predict if the review is a positive or negative review.

Analysis

CountVectorizer was ran twice, once with the **max_feature** parameter set at 500 then again at 100.

```
print(vectorizer.vocabulary_)
```

```
{'pizza': 63, 'really': 68, 'like': 50, 'restaurant': 71, 'street': 85, 'went': 98, 'good': 37, 'food': 30, 'servic
e': 77, 'meal': 53, 'week': 97, 'waiter': 94, 'asked': 2, 'menu': 54, 'took': 90, 'minutes': 55, 'terrible': 88, 'e
xperience': 27, 'didn': 19, 'days': 15, 'hour': 42, 'just': 48, 'ordered': 60, 'worst': 99, 'life': 49, 'did': 18,
'place': 64, 'called': 9, 'dinner': 23, 'order': 59, 'waitress': 95, 'table': 86, 'plate': 65, 'dining': 22, 'wan
t': 96, 'dine': 20, 'expensive': 26, 'diner': 21, 'dish': 24, 'cooked': 13, 'taste': 87, 'friends': 34, 'wait': 93,
'nice': 57, 'special': 81, 'bread': 7, 'great': 38, 'need': 56, 'campus': 10, 'usually': 91, 'recently': 69, 'ate':
3, 'veggie': 92, 'staff': 84, 'friday': 32, 'gone': 36, 'dishes': 25, 'quite': 67, 'served': 76, 'pasta': 61, 'sala
d': 74, 'bad': 4, 'looked': 51, 'glass': 35, 'home': 41, 'happened': 39, 'indian': 44, 'time': 89, 'chinese': 12,
'chicken': 11, 'best': 5, 'soup': 80, 'hard': 40, 'fresh': 31, 'favorite': 28, 'noodle': 58, 'love': 52, 'amazing':
0, 'felt': 29, 'delicious': 17, 'quality': 66, 'sauce': 75, 'ice': 43, 'cream': 14, 'birthday': 6, 'ambiance': 1,
'people': 62, 'friendly': 33, 'japanese': 46, 'restaurants': 72, 'spicy': 82, 'recommend': 70, 'cafe': 8, 'spinac
h': 83, 'deck': 16, 'soft': 79, 'right': 73, 'ingredients': 45, 'joe': 47, 'sister': 78}
```

Vocabulary w/max_feature set to 100

Lie Data Set

The results below show the probability of each review being either a lie (the first column) or real (the second column). The first image below shows the probabilities for the model with 500 max words, and the second is the max features for 100 words.

```
[[1.   0.  ]
 [1.   0.  ]
 [0.   1.  ]
 [0.64 0.36]
 [0.44 0.56]
 [0.69 0.31]
 [0.94 0.06]
 [0.07 0.93]
 [0.16 0.84]
 [0.07 0.93]
 [0.11 0.89]
 [0.06 0.94]
 [0.75 0.25]
 [0.41 0.59]
 [0.09 0.91]
 [0.88 0.12]
 [0.73 0.27]
 [0.29 0.71]
 [0.45 0.55]
 [0.43 0.57]
 [0.05 0.95]
 [0.32 0.68]
 [0.51 0.49]
 [0.17 0.83]
 [0.31 0.69]
 [0.03 0.97]
 [0.7  0.3 ]
 [0.41 0.59]]
```

<-500 feature probabilities

```
[[0.96 0.04]
 [0.4  0.6 ]
 [0.52 0.48]
 [0.49 0.51]
 [0.49 0.51]
 [0.65 0.35]
 [0.76 0.24]
 [0.9  0.1 ]
 [0.73 0.27]
 [0.61 0.39]
 [0.59 0.41]
 [0.98 0.02]
 [0.69 0.31]
 [0.42 0.58]
 [0.65 0.35]
 [0.74 0.26]
 [0.74 0.26]
 [0.45 0.55]
 [0.67 0.33]
 [0.19 0.81]
 [0.32 0.68]
 [0.89 0.11]
 [0.32 0.68]
 [0.5  0.5 ]
 [0.97 0.03]
 [0.56 0.44]
 [0.52 0.48]
 [0.88 0.12]]
```

<- 100 feature probabilities

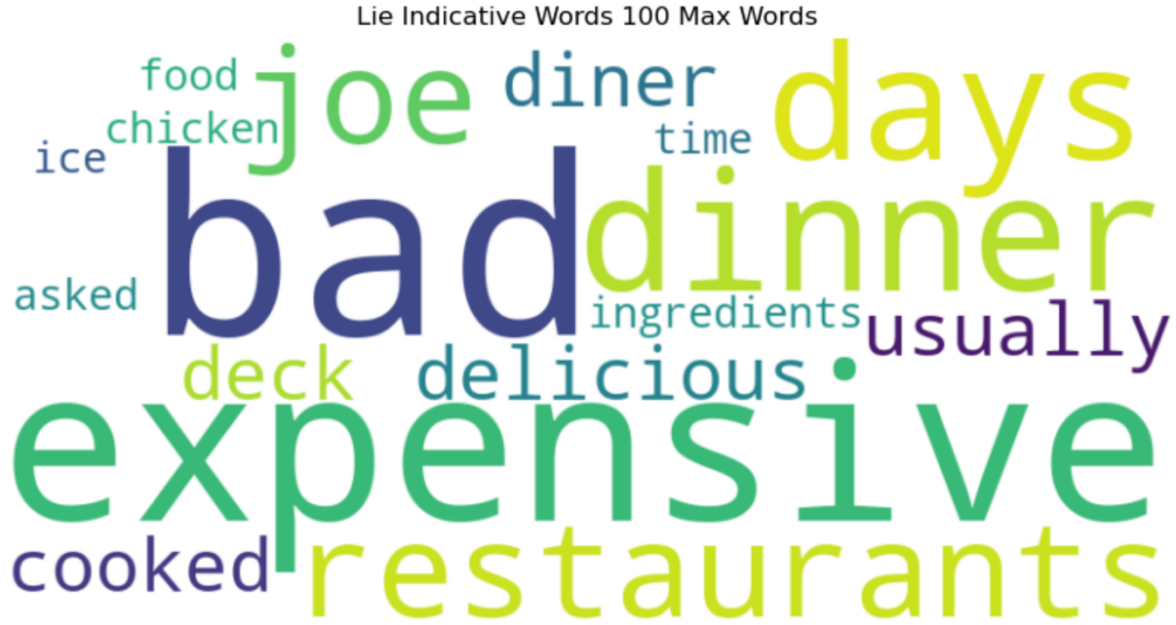
Both probability counts are very close in terms of fake vs real reviews, with the 500 max word model having a slightly higher probability per row for reviews being true compared to the 100 max word model. The accuracy score of the 500 max word model was slightly higher at 53%

than the 100 word model at 46%. The confusion matrix confirms the accuracy, with the first model putting more reviews in the true category and the second model in fake category.

```
The confusion matrix
for the 500 max word model is:
[[ 7 10]
 [ 3  8]]
```

```
The confusion matrix
for the 100 word model is:
[[ 9  4]
 [11  4]]
```

If we take a look at the 20 most indicative words we see no overlap between the words. These words also do not seem to indicate a fake or real review when taken out of context of the model.



Sentiment Data Set

The sentiment data set was prepped exactly like the lie dataset, with a testing and training dataset created from the original data. If we take a look at the probabilities per row, both models have a higher probability of predicting a negative review. The first model has a much higher accuracy of 75% compared to 57% for the second model.

```
[[0.   1.  ]
 [0.79 0.21]
 [0.49 0.51]
 [0.9  0.1 ]
 [0.49 0.51]
 [0.82 0.18]
 [1.   0.  ]
 [0.85 0.15]
 [0.97 0.03]
 [0.   1.  ]
 [0.67 0.33]
 [0.27 0.73]
 [0.99 0.01]
 [0.3  0.7 ]
 [0.07 0.93]
 [0.01 0.99]
 [1.   0.  ]
 [0.08 0.92]
 [0.6  0.4 ]
 [0.65 0.35]
 [0.79 0.21]
 [0.5  0.5 ]
 [0.03 0.97]
 [1.   0.  ]
 [0.   1.  ]
 [0.4  0.6 ]
 [0.99 0.01]
 [0.67 0.33]]
```

<- 500 feature probabilities

```
[[0.89 0.11]
 [0.36 0.64]
 [0.7  0.3 ]
 [0.73 0.27]
 [0.98 0.02]
 [0.53 0.47]
 [0.37 0.63]
 [0.88 0.12]
 [0.   1.  ]
 [0.48 0.52]
 [0.42 0.58]
 [0.97 0.03]
 [0.53 0.47]
 [0.28 0.72]
 [0.09 0.91]
 [1.   0.  ]
 [0.28 0.72]
 [0.14 0.86]
 [0.69 0.31]
 [0.02 0.98]
 [0.04 0.96]
 [0.07 0.93]
 [0.54 0.46]
 [0.33 0.67]
 [0.22 0.78]
 [0.56 0.44]
 [0.48 0.52]
 [0.62 0.38]]
```

<-100 feature probabilities

```
The confusion matrix
for the 500 word model is:
[[13  5]
 [ 2  8]]
```

```
The confusion matrix
for the 100 word model is:
[[8 7]
 [5 8]]
```

Overall the 500 max word model had higher accuracy for both lie and sentiment detection compared to the 100 word model. We do need to be mindful of overfitting the model so adding in additional parameters may also help with predicting better results. As we can also see, sentiment prediction yields higher accuracy compared to lie detection.

Conclusion

Based on the models results, it seems rather difficult for a computer to accurately predict if a review is fake or true. It's easier to predict sentiment analysis because the model can identify key words such as 'good', 'bad', 'great', 'horrible' which are indicative of a sentiment. There aren't indicative words that would suggest if a review is real or fake like there are positive and negative words for a sentiment review. We may need to employ some other methods such as understanding subjectivity or more tedious work like analyzing the profile of a poster to

determine if they are a human or a bot. We also may need a larger dataset to come to more definitive conclusions on detecting a real versus fake review.