Sana Khan

Homework 6 - Bernoulli and Multinomial Naïve Bayes

Introduction

With the rise of the internet, especially the pervasive influence of social media, people now have vast amounts of information at their fingertips in real time. Leaving reviews for different products or services has become a regular thing and is actively encouraged by many businesses. It's now a crucial part of how we make decisions 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 its 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 detect 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. We will be using two models to determine which is better at predicting fake reviews and the sentiment of the reviews. Being able to use models to detect reviews is an integral method to gain customers trust and protect a brands reputation.

Analysis - Bernoulli and Multinomial Naïve Bayes

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. I						
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. Since there are two models that are being used, **CountVectorizer** was run twice, once with the parameter binary=True for the Bernoulli model, and once without that parameter to represent Multinomial Naïve Bayes.

```
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 the four data frames which were then used in each model.

Model/Method

There are two models that were used in this classification task, Multinomial Naïve Bayes (MNB) and Bernoulli 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. The first model that was used for this text classification was Multinomial Naïve Bayes, which takes the frequency of the feature count whereas Bernoulli Naïve Bayes takes the binary frequency of feature. In other words Bernoulli shows if a feature is present or not.

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

Multinomial Naïve Bayes

CountVectorizer was ran with the **max_feature** parameter set at 500.

{'mike': 269, 'pizza': 318, 'high': 191, 'point': 323, 'really': 342, 'like': 237, 'buffet': 70, 'restaurant': 351, 'marshall': 259, 'street': 410, 'lot': 251, 'selection': 372, 'american': 18, 'went': 484, 'shopping': 381, 'friend': 179, 'olive': 287, 'oil': 286, 'disappointing': 151, 'good': 184, 'food': 175, 'service': 377, 'meal': 261, 'cold': 106, 'seven': 379, 'heaven': 189, 'known': 227, 'superior': 414, 'week': 482, 'disaster': 152, 'waiter': 471, 'notice': 282, 'asked': 28, 'times': 436, 'bring': 66, 'menu': 266, 'exceptional': 167, 'took': 440, 'minutes': 270, 'check': 89, 'spotted': 403, 'finished': 174, 'eating': 164, 'ordering': 293, 'xyz': 496, 'terrible': 428, 'experience': 169, 'yelp': 497, 'appetizer': 20, 'coupon': 122, 'applied': 23, 'checking': 90, 'person': 315, 'serving': 378, 'rude': 361, 'didn': 140, 'acknowledge': 9, 'abc': 7, 'days': 134, 'ago': 10, 'kept': 222, 'waiting': 473, 'hour': 200, 'just': 220, 'seated': 370, 'ordered': 292, 'chilis': 94, 'blvd': 62, 'worst': 490, 'life': 235, 'arrived': 25, 'waited': 470, 'hostess': 197, 'omg': 288, 'horrible': 196, 'receptionist': 345, 'did': 139, 'yesterday': 498, 'weekend': 483, 'place': 319, 'called': 78, 'ratastic': 338, 'chipotle': 98, 'dinner': 146, 'began': 47, 'order': 291, 'entered': 165, 'waitress': 474, 'came': 80, 'blanking': 58, 'looking': 250, 'threw': 433, 'table': 419, 'carlo': 82, 'plate': 322, 'shack': 380, 'dining': 145, 'southern': 397, 'comfort': 108, 'sounded': 395, 'die': 141, 'want': 478, 'dine': 142, 'price': 328, 'expensive': 168, 'diner': 143, 'dish': 153, 'cooked': 118, 'taste': 424, 'reminds': 350, 'smell': 390, 'try': 447, 'friends': 181, '6pm': 6, 'long': 247, 'queue': 336, 'wait': 469, 'nice': 280, 'worked': 487, 'hurry': 204, 'today': 437, 'special': 399, 'panera': 303, 'bread': 65, 'unfortunately': 454, 'great': 185, 'peach': 311, 'iced': 207, 'tea': 426, 'way': 481, 'average': 33, '10': 0, '20': 2, 'shown': 383, 'let': 234, 'tell': 427, 'kitty': 225, 'hoynes': 202, 'irish': 214, 'pub': 332, 'vegies': 464, 'lamb': 229, 'dry': 162, 'beer': 46, 'need': 278, 'say': 368, 'samarkand': 363, 'near': 277, 'su': 413, 'main': 254, 'campus': 81, 'usually': 460, 'don': 158, 'write': 494, 'reviews': 353, 'tripadvisor': 445, 'recently': 344, 'ate': 29, 'white': 485, 'castle': 84, 'busy': 73, '30': 4, 'come': 107, 'veggie': 463, 'burger': 71, 'patty': 309, 'properly': 331, 'hut': 205, 'syracuse': 418, 'staff': 404, 'unfriendly': 455, 'isn': 216, 'friday': 177, 'worse': 489, 'gone': 183, 'dishe

Snapshot of Vocabulary w/max_feature set to 500

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 MNB model, and the second is the confusion matrix.

```
[[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]]
```

<-MNB feature probabilities

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

The model shows a slightly higher probability per row for reviews being true. The accuracy score of the model was 53%. The confusion matrix confirms the accuracy, with the first model putting more reviews in the true category.

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. This shows that there may not be easily identifiable key words in this dataset for fake and real reviews.

Sentiment Data Set

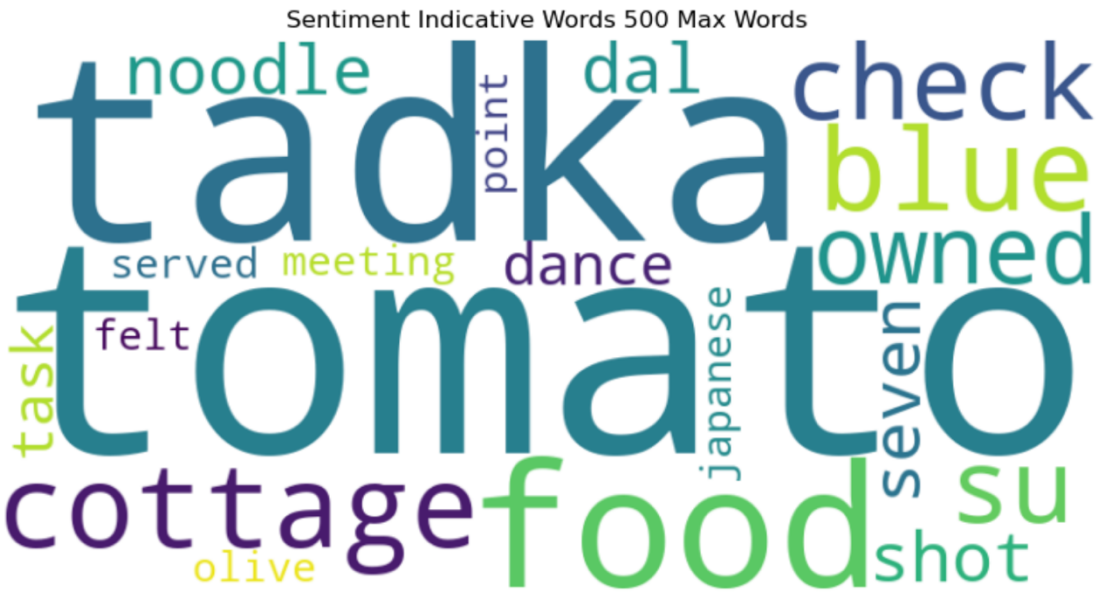
The sentiment data set was prepped exactly like the lie dataset, with a testing and training dataset created from the original data. If we take a look at the probabilities per row, the MNB models have a higher probability of predicting a negative review. This model has a much higher accuracy of 75% which is also confirmed by the confusion matrix.

```
[[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]]
```

<- MNB feature probabilities

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

When taking a look at the word cloud for sentiment analysis, there aren't key words that would indicate a sentiment but if looked at the context in the review, we may be able to understand what words impact the sentiment of a review.



Bernoulli Naïve Bayes

To instantiate the Bernoulli Model the parameter binary=True was used.

Lie Data Set

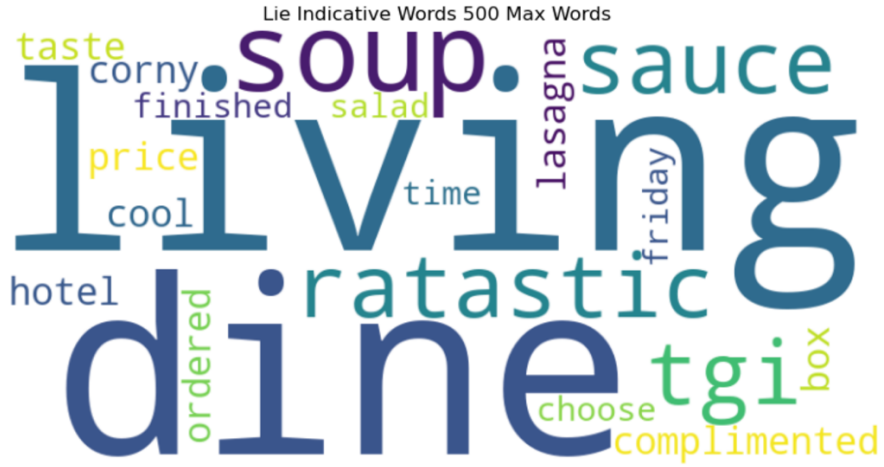
The model was first run on the Lie data set with the feature probability and confusion matrix shown below. This model has a higher probability for reviews being in the real category but has a lower accuracy at 39% than the MNB model. We can see in the confusion matrix it has incorrectly categorized 13/28 reviews as true when they should have been categorized as fake.

```
[[0.16 0.84]
 [0.1  0.9 ]
 [0.88 0.12]
 [0.67 0.33]
 [0.2  0.8 ]
 [0.42 0.58]
 [0.53 0.47]
 [0.35 0.65]
 [0.7  0.3 ]
 [0.3  0.7 ]
 [0.45 0.55]
 [0.52 0.48]
 [0.35 0.65]
 [0.52 0.48]
 [0.24 0.76]
 [0.31 0.69]
 [0.31 0.69]
 [0.43 0.57]
 [0.99 0.01]
 [0.59 0.41]
 [0.15 0.85]
 [0.83 0.17]
 [0.1  0.9 ]
 [0.36 0.64]
 [0.06 0.94]
 [0.33 0.67]
 [0.17 0.83]
 [0.15 0.85]]
```

<- Bernoulli feature probabilities

```
The confusion matrix
for the 500 max word model is:
[[ 5 13]
 [ 4  6]]
```

This can primarily be attributed to the challenge of identifying words that differentiate a positive review from a negative one. The Bernoulli model, which focuses solely on the presence or absence of words, might face difficulties capturing nuanced information within the text, potentially hindering its ability to accurately detect whether a review is fake or genuine. We can also see from the word cloud, that the top 20 words taken out of context would be difficult to use to identify reviews as fake or real.



Sentiment Data Set

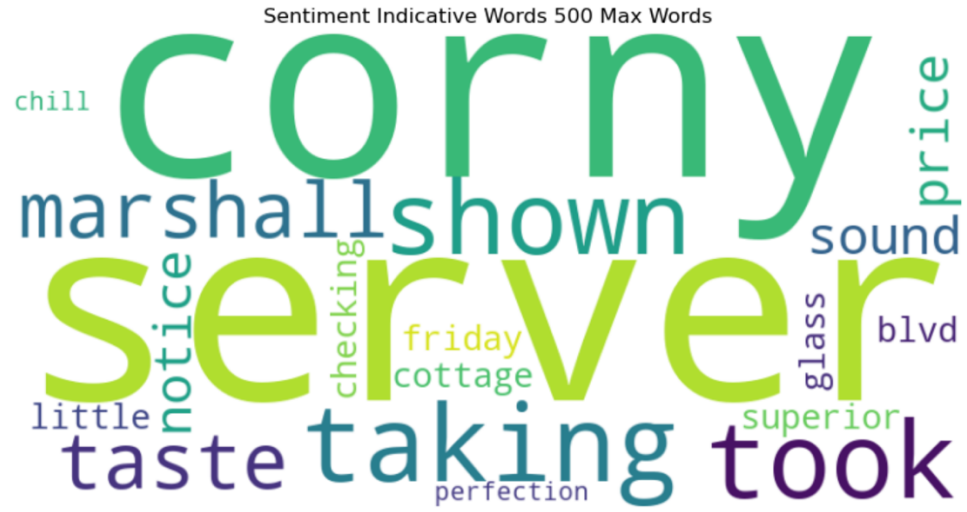
We then ran the Bernoulli model to predict sentiment analysis which preformed better than the lie prediction, but not as good as the MNB model. This model also had a higher probability of categorizing a review as negative and had a 68% accuracy. The confusion matrix supports this as seen in the screenshot below.

```
[[0.05 0.95]
 [0.53 0.47]
 [0.19 0.81]
 [0.82 0.18]
 [0.52 0.48]
 [0.84 0.16]
 [0.51 0.49]
 [0.53 0.47]
 [0.66 0.34]
 [0.61 0.39]
 [0.78 0.22]
 [0.99 0.01]
 [0.88 0.12]
 [0.32 0.68]
 [0.02 0.98]
 [0.   1.  ]
 [0.82 0.18]
 [0.07 0.93]
 [0.87 0.13]
 [0.61 0.39]
 [0.98 0.02]
 [0.85 0.15]
 [0.69 0.31]
 [1.   0.  ]
 [0.27 0.73]
 [0.99 0.01]
 [0.44 0.56]
 [0.66 0.34]]
```

<- Bernoulli feature probabilities

```
The confusion matrix
for the 500 word model is:
[[13  2]
 [ 7  6]]
```

We can also compare the word cloud of the top 20 words for sentiment analysis and identify features like price, server, and taste which are indicative of the sentiment of the review.



Overall the MNB model had higher accuracy for both lie and sentiment detection compared to the Bernoulli 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 were real or fake like there are positive and negative words for a sentiment review. It's also more difficult for Bernoulli to accurately predict compared to MNB, and this is most likely because Bernoulli is only looking at binary feature results, whereas MNB counts at the frequency of each feature. Being able to use the frequency count may lead to a better understanding of the context of a review, which can provide more accurate results.

We may also 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.