DISNEYLAND REVIEWS SENTIMENT ANALYSIS

By Megan Tan

DATASET INFORMATION AND PREPROCESSING

DATASET INFORMATION - EXPLORATION

```
In [2]: df.shape

Out[2]: (42656, 6)

In [3]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42656 entries, 0 to 42655
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   ï»¿Review_ID       42656 non-null  int64
 1   Rating             42656 non-null  int64
 2   Year_Month         42656 non-null  object
 3   Reviewer_Location  42656 non-null  object
 4   Review_Text        42656 non-null  object
 5   Branch             42656 non-null  object
dtypes: int64(2), object(4)
memory usage: 2.0+ MB
```

DATASET INFORMATION - EXPLORATION

```
In [2]: df.head()
Out[2]:
```

	ï»¿Review_ID	Rating	Year_Month	Reviewer_Location	Review_Text	Branch
0	670772142	4	2019-4	Australia	If you've ever been to Disneyland anywhere you...	Disneyland_HongKong
1	670682799	4	2019-5	Philippines	Its been a while since d last time we visit HK...	Disneyland_HongKong
2	670623270	4	2019-4	United Arab Emirates	Thanks God it wasn t too hot or too humid wh...	Disneyland_HongKong
3	670607911	4	2019-4	Australia	HK Disneyland is a great compact park. Unfortu...	Disneyland_HongKong
4	670607296	4	2019-4	United Kingdom	the location is not in the city, took around 1...	Disneyland_HongKong

- 42656 rows, 6 columns
- columns of interest are rating, and review_text

DATASET INFORMATION - PREPROCESSING

```
In [5]: df.drop_duplicates(keep='first', inplace=True)
        df
```

Out[5]:

	ï»¿Review_ID	Rating	Year_Month	Reviewer_Location	Review_Text	Branch
0	670772142	4	2019-4	Australia	If you've ever been to Disneyland anywhere you...	Disneyland_HongKong
1	670682799	4	2019-5	Philippines	Its been a while since d last time we visit HK...	Disneyland_HongKong
2	670623270	4	2019-4	United Arab Emirates	Thanks God it wasn t too hot or too humid wh...	Disneyland_HongKong
3	670607911	4	2019-4	Australia	HK Disneyland is a great compact park. Unfortu...	Disneyland_HongKong
4	670607296	4	2019-4	United Kingdom	the location is not in the city, took around 1...	Disneyland_HongKong
...	...	...	...	...	...	...
42651	1765031	5	missing	United Kingdom	i went to disneyland paris in july 03 and thou...	Disneyland_Paris
42652	1659553	5	missing	Canada	2 adults and 1 child of 11 visited Disneyland ...	Disneyland_Paris
42653	1645894	5	missing	South Africa	My eleven year old daughter and myself went to...	Disneyland_Paris
42654	1618637	4	missing	United States	This hotel, part of the Disneyland Paris compl...	Disneyland_Paris
42655	1536786	4	missing	United Kingdom	I went to the Disneyparis resort, in 1996, wit...	Disneyland_Paris

42644 rows × 6 columns

- 42644 rows after removing duplicates

DATASET INFORMATION - PREPROCESSING

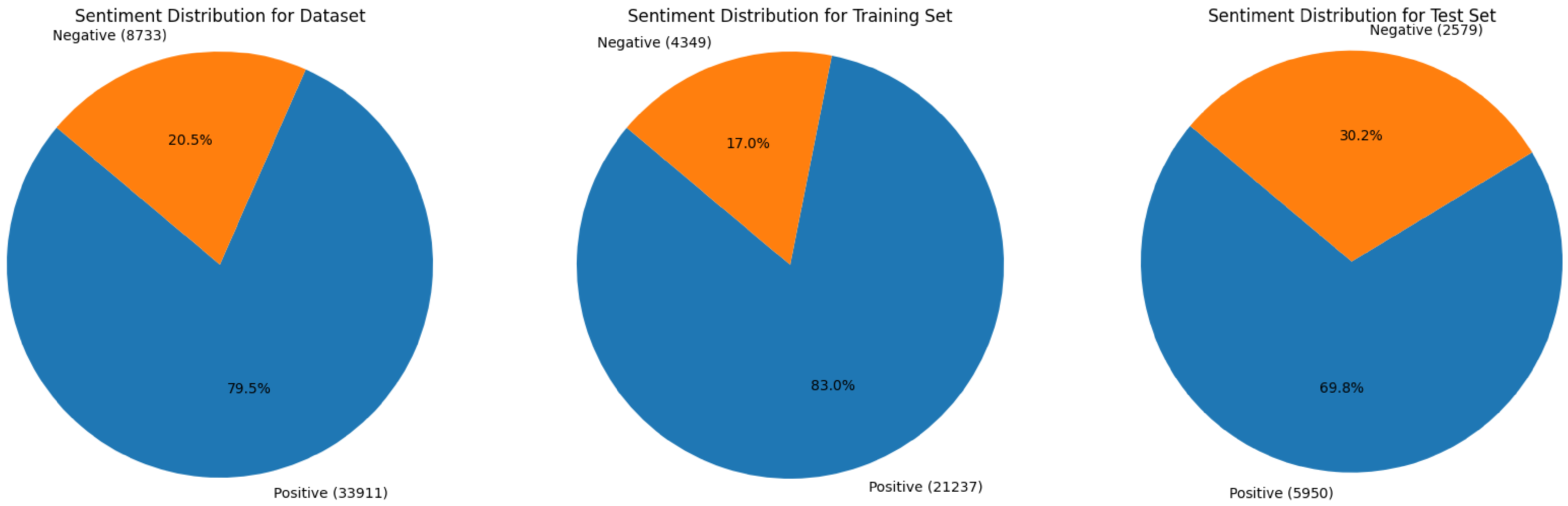
```
In [4]: df['Rating'].value_counts()

Out[4]: Rating
        5    23146
        4    10775
        3     5109
        2     2127
        1     1499
        Name: count, dtype: int64
```

- **Predictor variable is “Review_Text”.**
  - Lowercasing and stop word removal applied
- **Response variable is “Sentiment”**
  - Sentiment is based on rating.
    - Positive sentiment = 4-5
    - Negative sentiment = 1-3

LABEL DISTRIBUTIONS FOR DATASET

LABEL DISTRIBUTION



- Training set is first 80% of data
- Test set is last 20% of data

MODEL INFORMATION

PSEUDOCODE

1. Clean Data – drop duplicates, remove stopwords, apply lowercasing to reviews
2. Split test and training sets – training set is first 80% of data, test set is last 20% of data
3. Get vocabulary from the entire dataframe
4. Using binary bag-of-words technique, set presence of a token in a review to 1 for each review in the training set
5. Using binary bag-of-words technique, set presence of token in a review to 1 for each review in the test set
6. Calculate probability of having a positive sentiment and a negative sentiment in the training set.
   - E.g. P(positive) = count of positive sentiment in training set/number of rows in training set
7. Calculate probability of each token given each sentiment. Use +1 smoothing
   - Formula is ((Count x= X, y = Y) + 1)/(count(number of words when y = Y) + number of words in Vocabulary)
     - x is the token, y is the sentiment
8. Predict sentiments for the test set's reviews
9. Compare predicted sentiments with actual sentiments to calculate metrics

PSEUDOCODE – PREDICTION

```
predictions = []
log_probability_negative = log(probability review is negative) #step 6
log_probability_positive = log(probability review is positive) #step 6
log_probabilities_token_negative = log(probabilities of each token given negative sentiment) #step 7
log_probabilities_token_positive = log(probabilities of each token given positive sentiment) #step 7
For each review to predict
        log_probability_this_review_negative = log_probability_negative
        log_probability_this_review_positive = log_probability_positive
        For each word in review
                log_probability_this_review_negative = log_probabilities_token_negative[this_token]
                log_probability_this_review_positive = log_probabilities_token_positive[this_token]
        prediction = 1 if log_probability_this_review_positive > log_probability_this_review_negative else 0
        push prediction into predictions
```

PSEUDOCODE- ALTERNATIVE MODEL

1. Clean Data – drop duplicates, remove stopwords, apply lowercasing to reviews
2. Split test and training sets – training set is first 80% of data, test set is last 20% of data
3. Undersample training set so that training and test sets are the same size
4. Get vocabulary from the entire dataframe
5. Using binary bag-of-words technique, set presence of a token in a review to 1 for each review in the training set
6. Using binary bag-of-words technique, set presence of token in a review to 1 for each review in the test set
7. Calculate probability of having a positive sentiment and a negative sentiment in the training set.
   - P(positive) = count of positive sentiment in training set/number of rows in training set
7. Calculate probability of each token given each sentiment. Use +1 smoothing
   - Formula is ((Count x= X, y = Y) + 1)/(count(number of words when y = Y) + number of words in Vocabulary)
     - x is the token, y is the sentiment
8. Predict sentiments for the test set's reviews
9. Compare predicted sentiments with actual sentiments to calculate metrics

PACKAGES

- Numpy – for array operations
- Pandas – to store dataframe
- Matplotlib – for plotting data
- Nltk – tokenize reviews and remove stopwords
- Scipy – for sparse matrix to store binary bag of words as the data set is large
- Sklearn – for confusion matrix calculation, ROC curve calculation and resampling training set in the alternative model
- Seaborn – for confusion matrix visualization

- Calculating binary bag of words and probabilities, and predicting sentiment are done manually without any packages for the calculations

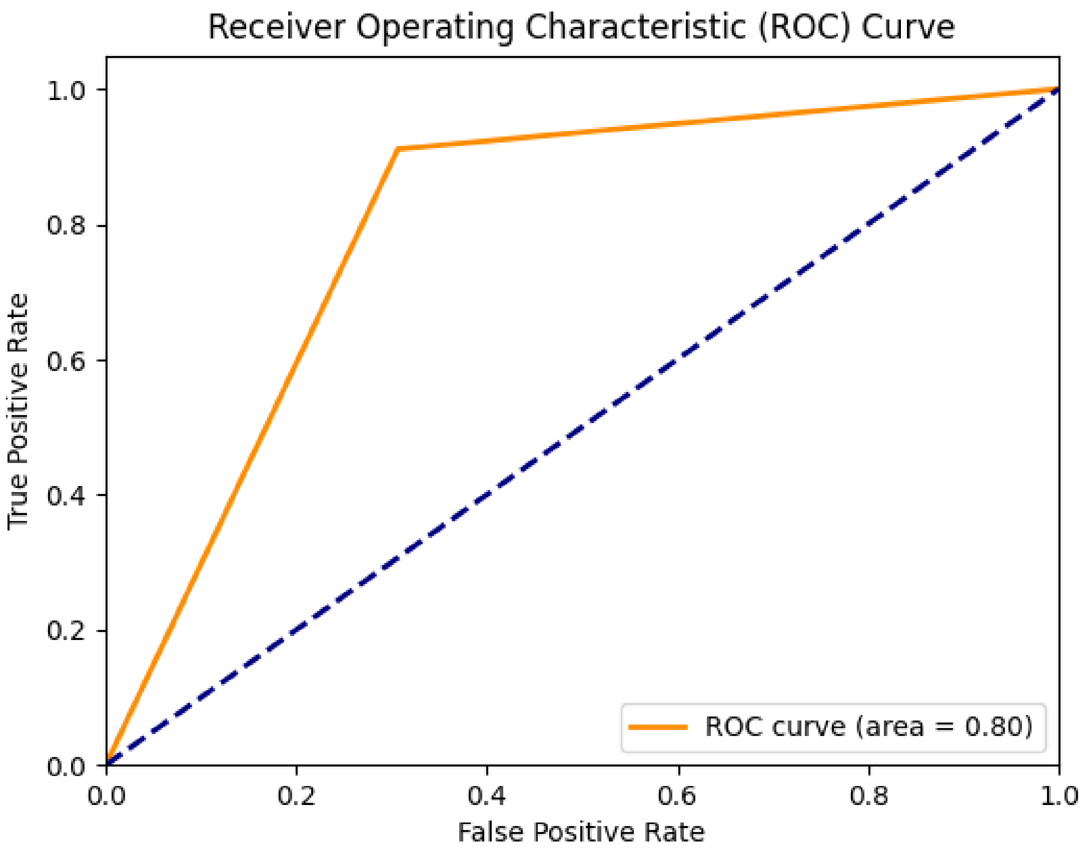
MODEL EVALUATION

MODEL METRICS

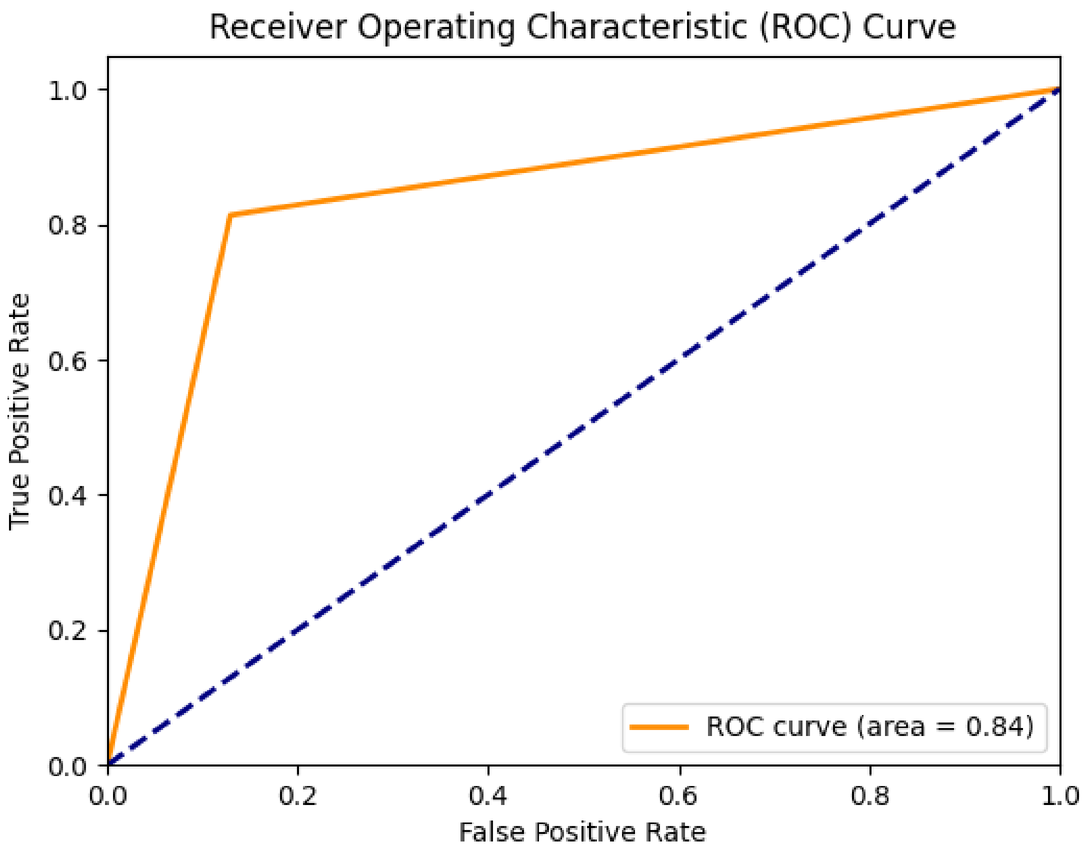
METRIC	MODEL	ALTERNATE MODEL
Number of True Positives	2245	1788
Number of True Negatives	4840	5421
Number of False Positives	334	791
Number of False Negatives	1110	529
Sensitivity (recall)	0.6692	0.7717
Specificity	0.9355	0.8727
Precision	0.8705	0.6933
Negative Predictive Value	0.8134	0.9111
Accuracy	0.8307	0.8452
F-score	0.7567	0.7304

ROC CURVE

Model

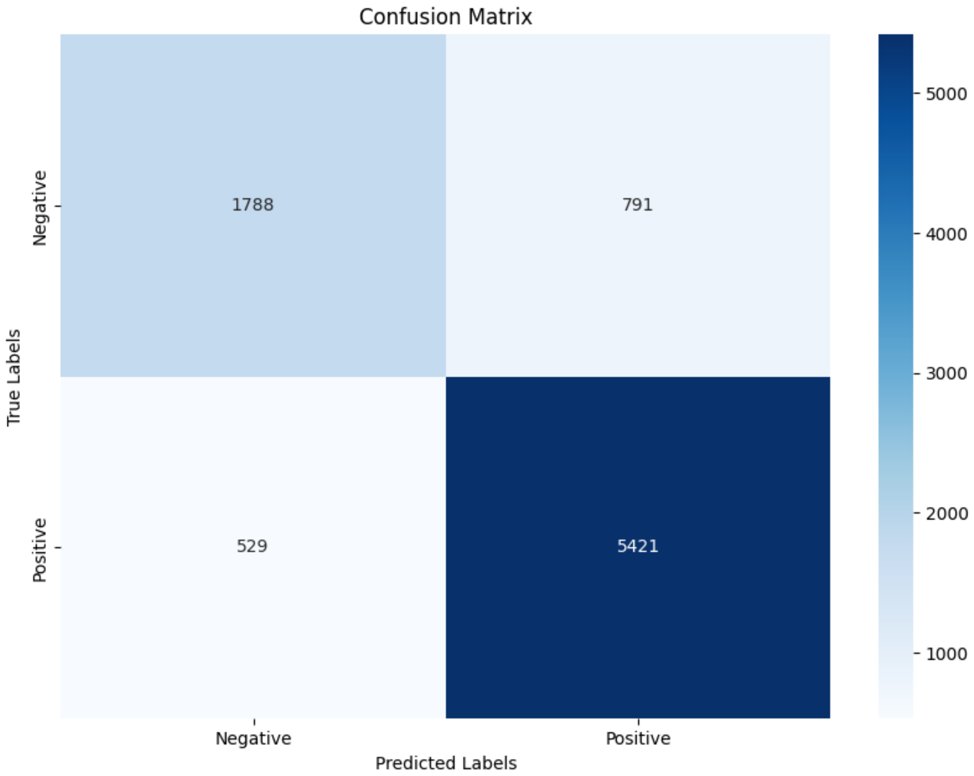


Alternate Model

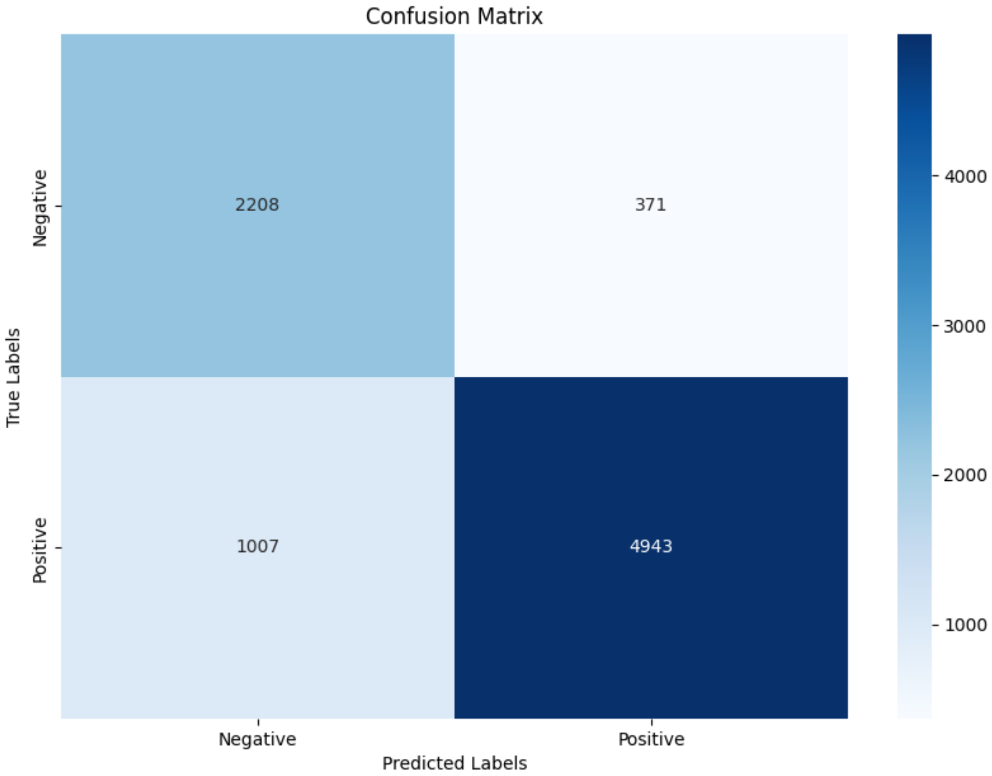


CONFUSION MATRIX

Model



Alternate Model



OBSERVATIONS AND SUMMARY

SUMMARY

- Alternate model performed better. This is expected as the imbalance data was dealt with
- Surprisingly, the word “sucks” is positive in the model. However, it is negative in the alternate model

Model Testing

```
     disney sucked

    was classified as POSITIVE.
    P(POSITIVE | S) = 4.968894897491188e-08
    P(NEGATIVE | S) = 2.2667890287655082e-08
Do you want to enter another sentence? [Y/N] y
Please enter your sentence: sucks
Sentence S:

     sucks

    was classified as POSITIVE.
    P(POSITIVE | S) = 9.411756646693865e-06
    P(NEGATIVE | S) = 6.312675167725326e-06
```

Alternate Model Testing

```
     disney sucked

    was classified as NEGATIVE.
    P(POSITIVE | S) = 4.980958520801664e-08
    P(NEGATIVE | S) = 6.642274074938692e-08
Do you want to enter another sentence? [Y/N] y
Please enter your sentence: sucks
Sentence S:

     sucks

    was classified as NEGATIVE.
    P(POSITIVE | S) = 7.349899306379499e-06
    P(NEGATIVE | S) = 1.799058274768534e-05
```

SUMMARY

- To improve performance
    - Data can be shuffled when splitting into training and test sets
    - Dictionary of positive and negative words can be provided
    - Use cross validation techniques
    - Use a different technique (not Naïve Bayes)