USC Marshall
School of Business

DSO 599 – Text Analytics & Natural Language Processing
Instructor: Yu Chen
Midterm Exam
April 9th, 2019

WRITE ALL ANSWERS ON SEPARATE SHEETS OF PAPER

Part A (recommended 45 minutes)

1. Naïve Bayes (recommended 15 minutes)

Your marketing firm is managing the social media marketing campaign for a theatrical release called Shazam! Each comment has already been tagged with either **Intent to Buy** (indicating a willingness to see the film) or **No Intent to Buy** (no interest in seeing the film)

Intent to Buy Tickets:

1. Love this movie. Can't wait!
2. I want to see this movie so bad.
3. This movie looks amazing.

No Intent to Buy Tickets:

1. Looks bad.
2. Hard pass to see this movie.
3. So boring!

Stopwords to remove: *to, this*

You **do not need to perform stemming or lemmatization, and can disregard punctuation / case-sensitivity** (ie. *Can't = can't*).

A. Calculate the following probabilities:

i. .25 pts - P(x = "so" | y = Intent to Buy)
ii. .25 pts - P(x = "see")
iii. .25 pts - P(x1 = "see", x2 = "movie")
iv. .25 pts - P(y = No Intent | x = "bad")

B. Are the words "love" and "movie" independent in terms of sentence occurrence? Prove why or why not (1 pts).

C. A new comment is posted: This looks bad. Assume a Naïve Bayes Bag of Words model with conditional independence. Calculate the following:

i. .50 pts - **prior probabilities** for Intent to Buy and No Intent to Buy
ii. .50 pts - **evidence**
iii. .50 pts - **likelihood** for this new comment
iv. .50 pts - **posterior probabilities** for Intent to Buy and No Intent to Buy after observing this new comment

2. Vectorization and Similarity (recommended 15 minutes)

Your company has **an inventory of products tagged with keywords** and a search feature that uses vectorization and similarity to match user queries with desired products based upon these keywords. Assume that part of your company's inventory is described below:

Product A: trendy wood chair **Product B:** old discount blue jeans old navy **Product C.** discount old blue trendy sweater

a. Generate the following document vectors (you may write them as a matrix or table):
   - **i.** word count (count vectorized) **(0.25 pts)**
   - **ii.** one-hot encoded vectors (**0.25 pts)**
   - **iii.** TF-IDF **(1 pt) –** calculate IDF for each of the words, then term frequency (TF) for each of document – word combinations.

b. A new user query is submitted via your company's website: **old jeans.** Assuming **word count vectorized** documents and **Euclidean distance**, would your backend search engine recommend **Product B** or **Product C?** (**1 pt**)

c. A second user queries for **discount chair**. Assuming **TF-IDF** vectors are used with **cosine similarity,** would **Product A or Product B be recommended**? (**1 pt**)

To avoid repetitive computations, you may use the following term frequency (TF) and inverse document frequency (IDF) functions (these are simply the same functions from the textbook, but without the log or square root):

TF = n (t,d) ie. *number of times term **t** appears in document **d***

$$\textbf{IDF} = 1 + \frac{N}{df(t)+1}$$

d. Explain **two reasons** why cosine similarity/distance is often preferred over Euclidean distance when calculating similarity between documents within NLP (**0.5 pts**).

3. Classification & Model Evaluation (recommended 15 minutes):

A security company has built a cybersecurity machine learning model that monitors text conversations on the internet to determine if the comment/post is a potential public security threat. It has been tested on a **set of 15 unseen internet posts.**

	Actual Outcome	**Predicted Outcome**
1.	Threat	Threat
2.	No Threat	No Threat
3.	No Threat	No Threat
4.	No Threat	No Threat
5.	No Threat	No Threat
6.	Threat	No Threat
7.	No Threat	No Threat
8.	No Threat	Threat
9.	No Threat	No Threat
10.	Threat	Threat
11.	Threat	No Threat
12.	No Threat	No Threat
13.	No Threat	No Threat
14.	No Threat	No Threat
15.	No Threat	Threat

a. Compute the following:
   i. Accuracy (0.25 pts)
   ii. Precision (0.25 pts)
   iii. Recall (0.25 pts)
   iv. F1 Score (0.25 pts)

b. Construct the confusion matrix for this model's test results. Make sure to label whether the columns/rows are actual or predicted results (**1 pt**).

c. Which of the following metrics calculated in part a. is **most relevant** for this particular use case? Explain why (**1 pt)**

d. What is a **baseline level of accuracy** that you should expect (ie. any accuracy below this threshold means the model is useless)? Explain why (**1 pt**).

Part B (recommended 45 minutes)

1. Vectorization and Classification (7pts - recommended 25 minutes):

You are a member of Amazon's Product Analytics team. A product manager for the Amazon Fire product suite has approached you with a dataset called **amazon_fire_reviews.csv** and some task:

- Provide her a **list of the most similar pair of reviews for each type of product** (so if review A and review B are identified as the most similar, they should both be rating the same product).
- The Fire product suite is always extremely popular with customers. However, the product manager wants a way to quickly flag new negative comments she finds on other forums. She wants you to **build a classification model that performs better than baseline accuracy**. She considers a positive review to be a 5 rating, and a negative review to be less than or equal to 4.
- **Report in no more than 2 sentences** your evaluation of the model's performance using relevant performance metrics.

In terms of technical implementation, she has a few requirements:

1. Only use features that **do not include numbers**, and keep only tokens that are 4 or more characters
2. It is up to you to decide which n-gram works the best (1-gram, 2-gram, 3-gram, etc...)
3. Only use up to **1000** features
4. Use TF-IDF vectors.

2. N-Gram Language Model (5pts - recommended 20 minutes):

1. Find the probability that a word appears in a sentence in the corpus.

2. Find the probability that a word STARTS a sentence.

3. Calculate the probability of a sequence of two words (bi-gram).

4. Calculate the probability of the following test sentences:

A. He said they went home

B. They want to go home

5. Find the perplexity of these two test sentences' probabilities. If you could not find the probabilities from Question 4, you may still earn full credit on this question - make up any two valid probabilities (between 0 and 1) to substitute in their place.