HW3-Week 5-Emotion Classification Using Naïve Bayes

(100 points) **Discussion Week 5**
Due Sunday May 7th, 11:59 pm PST

Ellen Yim

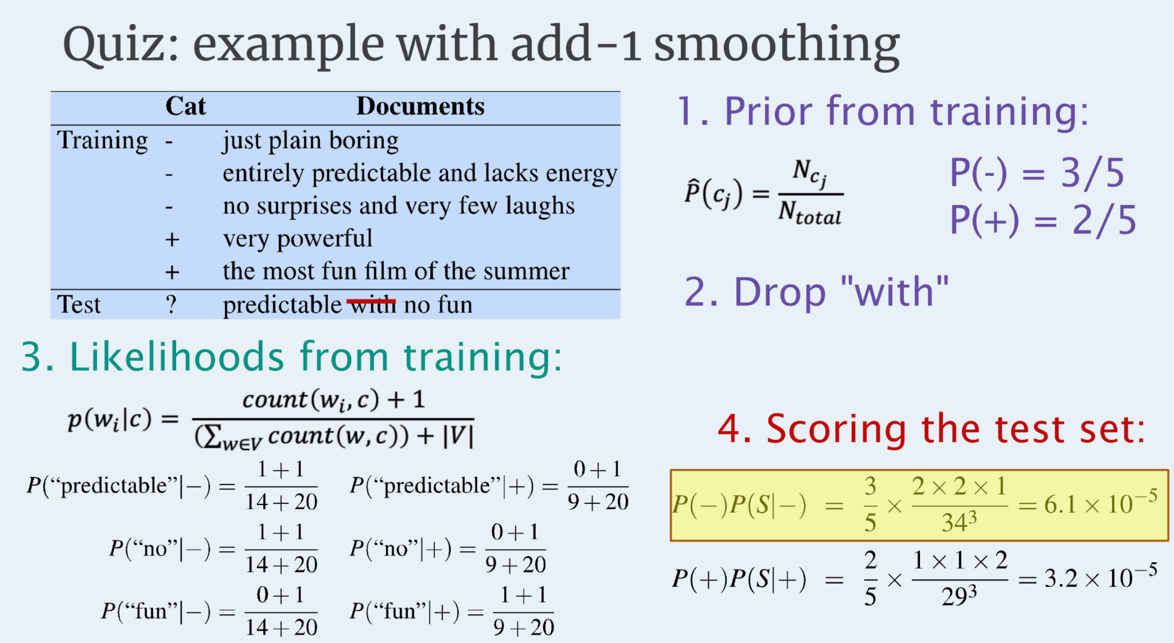
Objective: In this assignment, you will investigate the core principles of the Naïve Bayes algorithm, utilize it for an emotion detection task, and assess its effectiveness in identifying six emotion categories based on the dataset you and your peers created last week.

Instructions: The assignment is divided into four sections. Complete all sections and submit the following:

1. A PDF or word doc answering the below questions.
2. Source code files **(e.g., Colab)** containing your implementation of the Naïve Bayes classifier and data preprocessing steps.

Hint: You only need to construct your data with categories and sentences corresponding to the specific category, not the emotion lexicon. In the assignment, you have six classes rather than two. Sentences in combination categories should be included in each category. For example, a sentence in Sadness + Joy is a sentence in Sadness and also a sentence in Joy.

6 categories `{Fear, Anger, Surprise, Disgust, Sadness, Joy}.`



Quiz: example with add-1 smoothing

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable ~~with~~ no fun

1. Prior from training:

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}}$$

P(-) = 3/5
P(+) = 2/5

2. Drop "with"

3. Likelihoods from training:

$$p(w_i|c) = \frac{count(w_i, c) + 1}{(\sum_{w \in V} count(w, c)) + |V|}$$

$$P(\text{“predictable”}|-) = \frac{1+1}{14+20} \qquad P(\text{“predictable”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“no”}|-) = \frac{1+1}{14+20} \qquad P(\text{“no”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“fun”}|-) = \frac{0+1}{14+20} \qquad P(\text{“fun”}|+) = \frac{1+1}{9+20}$$

4. Scoring the test set:

$$P(-)P(S|-) = \frac{3}{5} \times \frac{2 \times 2 \times 1}{34^3} = 6.1 \times 10^{-5}$$

$$P(+)P(S|+) = \frac{2}{5} \times \frac{1 \times 1 \times 2}{29^3} = 3.2 \times 10^{-5}$$

Section 1: Understanding Naïve Bayes (10 points)

1.1: Research the Naïve Bayes algorithm and write a brief summary of the algorithm's principles, assumptions, and applications.

Naive Bayes algorithm is a classification technique using Bayes' theorem with an independence assumption among predictors where it assumes that a particular feature presence in a class is unrelated to another feature's presence. When the independence assumption is true the classifier performs better compared to many machine learning models. Some applications include Real-time prediction, text classification, and a recommendation system. [1]

1.2: Explain the role of conditional probability and independence in Naïve Bayes classification.

Conditional probability used in Naive Bayes classification is used the same in Bayes' theorem where based on the occurrence of another event, it's equal to the likelihood of the second event

[1] https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

given the first event multiplied by probability of the first event. Independence is the assumption that features are independent of each other given the class. [2]

Section 2: Preprocessing the Data (10 points)

2.1: Download the dataset you created last week **[https://docs.google.com/spreadsheets/d/1FO779z232nz8pVEk2-lfA7pRX0BW1Nqlk1x-oPoxRmc/edit?usp=sharing]** . Please use the first 30 rows as training, the next 10 rows (30-40) as validation, and the next 10 rows (40-50) as testing set.

https://colab.research.google.com/drive/12H0pRN-Vv2XVkRSxSKuhFk-RPtuneCS6#scrollTo=Y7dkkP_iRT5Q

```
file_path = 'CS173-publishedsheet.xlsx'  # Replace with your Excel file path
df = pd.read_excel(file_path)
# Remove newline characters from all cells
df = df.replace({r'\n': ' '}, regex=True)
drop_columns = ['Sadness Lexicons', 'Joy Lexicons', 'Fear Lexicons', 'Anger Lexicons', 'Surprise Lexicons', 'Disgust Lexicons', 'Sadness + Joy Lexicons', 'Fear + Anger Lexicons', 'Surprise + Disgust Lexicons', 'Sadness + Joy + Fear Lexicons']
df = df.drop(drop_columns, axis=1).dropna()
df
training_set = df.iloc[:30]
#training_set
validation_set = df.iloc[31:41]
#validation_set
testing_set = df.iloc[42:]
#testing_set
```

2.2: Preprocess the data by tokenizing the text (make sure the end of sentence tokens, such as .!? are properly tokenized). Copy and paste one tokenized sentence for each emotion here.

```
Tokenized sadness sentence:
[['The', 'devastating', 'news', 'of', 'the', 'child', "'s", 'abduction',
'left', 'a', 'solemn', 'shadow', 'over', 'the', 'family', 'for', 'the', 'next',
'month', '.']]
```

[2] https://nlp.stanford.edu/IR-book/html/htmledition/properties-of-naive-bayes-1.html#:~:text=Naive%20Bayes%20is%20so%20called,true%20for%20terms%20in%20documents.

```
Joy sentence:
[['It', 'was', 'a', 'sunny', 'summer', 'morning', 'and', 'the', 'laughter',
'of', 'children', 'could', 'be', 'heard', 'from', 'the', 'pool', 'as', 'they',
'splashed', 'water', 'onto', 'each', 'other', '.']]

Fear sentence:
[['As', 'he', 'walked', 'in', 'the', 'dead', 'of', 'night', 'he', 'could',
'hear', 'sudden', 'footsteps', 'echoing', 'from', 'the', 'alleyway', ',', 'a',
'shiver', 'went', 'down', 'his', 'spine', '.']]

Anger sentence:
[['While', 'driving', 'his', 'family', 'to', 'a', 'restaurant', 'a', 'car',
'recklessly', 'changed', 'lanes', 'barely', 'missing', 'him', ',', 'his',
'face', 'flushed', 'red', 'with', 'fury', 'knowing', 'how', 'close', 'his',
'family', 'was', 'to', 'being', 'seriously', 'harmed', '.']]

Surprise sentence:
[['She', 'was', 'startled', 'unexpectedly', 'as', 'everyone', 'sprang', 'from',
'their', 'hiding', 'spot', 'to', 'celebrate', 'her', 'birthday']]

Disgust sentence:
[['She', 'had', 'forgotten', 'to', 'take', 'out', 'the', 'trash', 'before',
'leaving', 'for', 'holiday', ',', 'when', 'she', 'returned', 'she', 'was',
'repulsed', 'by', 'the', 'stench', 'of', 'decaying', 'food', '.']]

Sadness + Joy sentence:
[['When', 'I', 'visited', 'my', 'old', 'childhood', 'home', ',', 'I', 'felt',
'a', 'wave', 'of', 'bittersweet', 'nostalgia', 'as', 'I', 'remembered', 'the',
'pleasant', 'memories', 'in', 'that', 'house', ',', 'while', 'also', 'feeling',
'melancholy', 'as', 'I', '’', 'd', 'never', 'get', 'to', 'experience', 'those',
'moments', 'again', '.']]

Fear + Anger Sentence:
[['Tom', 'was', 'fearful', 'of', 'his', 'bully', 'and', 'remained', 'docile',
'while', 'being', 'picked', 'on', 'to', 'avoid', 'provoking', 'the', 'bully',
'into', 'a', 'fit', 'of', 'rage', 'where', 'Tom', 'may', 'be', 'harmed', '.']]

Surprise + Disgust Sentence:
[['He', 'opened', 'the', 'package', 'of', 'his', 'food', 'delivery', 'and',
'was', 'shocked', 'to', 'see', 'a', 'dead', 'insect', 'laying', 'on', 'top',
'his', 'food', ',', 'the', 'idea', 'of', 'accidentally', 'eating', 'it',
'revolted', 'him', '.']]

Sadness + Joy + Fear Sentence:
[['The', 'parents', 'watched', 'their', 'son', 'leave', 'for', 'college',
'and', 'were', 'proud', 'of', 'his', 'accomplishment', ',', 'at', 'the',
'sametime', 'they', 'were', 'sorrowful', 'that', 'he', 'would', 'no', 'longer',
'live', 'with', 'them', 'and', 'anxious', 'that', 'he', 'would', 'be', 'alone',
'to', 'face', 'his', 'challenges', '.']]
```

Section 3: Implementing Naïve Bayes Classifier (60 points)

3.1: Implement and calculate the priors for each emotion category. Copy and paste the priors here. (10 points)

```
P(EMO)where EMO \in {Fear, Anger, Surprise, Disgust, Sadness, Joy}

P(Fear) =  0.2
P(Anger) =  0.13333333333333333
P(Surprise) =  0.13333333333333333
P(Disgust) =  0.13333333333333333
P(Joy) =  0.2
P(Sadness) =  0.2
```

3.2: Implement and calculate the likelihoods of each word given an emotion category using Laplace smoothing. Indicate which sections in your Colab are corresponding to this question and provide necessary explanations to TA on how to run it. (20 points)

https://colab.research.google.com/drive/12H0pRN-Vv2XVkRSxSKuhFk-RPtuneCS6#scrollTo=n936UA_mwGqu

- My code for likelihood of each word given an emotion category is labeled under Section 3.2
- To run this section, Need to run the cell before Section 3 (Stopwords, lower function, and the cell for finding the vocabulary (# of unique words in whole corpus)
- Make sure the excel file is correct format when uploading excel to Colab

3.3: Implement the Naïve Bayes classifier to predict the emotion of a given text `S=`

```
As she hugged her daughter goodbye on the first day of college, she
felt both sad to see her go and joyful knowing that she was embarking
on a new and exciting chapter in her life.
```

https://colab.research.google.com/drive/12H0pRN-Vv2XVkRSxSKuhFk-RPtuneCS6#scrollTo=GZT3Wx7ELEHD

- Code is under Section 3.3

Sentence was classified as Surprise emotion

Indicate which sections in your Colab are corresponding to this question and provide necessary explanations to TA on how to run it. Copy and paste the prediction probability here. (30 points)

```
P(S|EMO)where EMO \in {Fear, Anger, Surprise, Disgust, Sadness, Joy}
```

P(S|Sadness) = 3.596292277669418e-51
P(S|Joy) = 1.595782839409849e-41
P(S|Fear) = 5.2261677903748786e-36
P(S|Anger) = 2.1089115698742628e-33
P(S|Surprise) = 3.163166525815443e-27
P(S|Disgust) = 2.5745837274344748e-30

https://colab.research.google.com/drive/12H0pRN-Vv2XVkRSxSKuhFk-RPtuneCS6#scrollTo=GZT3Wx7ELEHD

- Need to run all cells (data preprocessing, section 3.1, and section 3.2) to run section 3.3
- Under Section 3.3, second printed value is the score on test sentence (given an emotion)

Section 4: Evaluating the Classifier on Test Set (20 points + 10 bonus points)

4.1: Generate a confusion matrix (6x6) and analyze the results on the **test dataset**. You can use the existing Python package. Copy and paste your confusion matrix here. (10 points)

```
[[1 1 0 1 0 1]
 [0 0 0 0 0 0]
 [0 2 0 0 2 4]
 [0 0 0 0 2 2]
 [2 0 0 2 0 0]
 [0 1 0 0 1 2]]
```

4.2: Calculate the accuracy, precision, recall, and F1-score for the Naïve Bayes classifier of the **Joy** category on the test dataset. Copy and paste your results here and indicate sections of code in colab for this computation. (10 points)

Joy category on test dataset

```
Accuracy: 0.22

Micro Precision: 0.22
Micro Recall: 0.22
Micro F1-score: 0.22
```

```
Macro Precision: 0.20
Macro Recall: 0.04
Macro F1-score: 0.07

Weighted Precision: 1.00
Weighted Recall: 0.22
Weighted F1-score: 0.36
```

Code in Colab (Labeled under Section 4.2)
https://colab.research.google.com/drive/12H0pRN-Vv2XVkRSxSKuhFk-RPtuneCS6#scrollTo=t8Is5YVzDEvU&uniqifier=1

4.3: Compare the performance of your Naïve Bayes classifier to an existing naive bayes implementation (e.g.,https://scikit-learn.org/stable/modules/naive_bayes.html) for *question 4.1. (bonus 10 points)*