1. Goal: implement a classifier that can distinguish btwn Economist + onion articles tht have been pre-processed.
V = Set of all words found in the articles (vocabulary)
$|V|$ = size of vocabulary
Article$^{(i)}$ = $\langle X_1, \ldots X_{|V|} \rangle$ where $X_i = \{0,1\}$
1 if i^{th} word appears in dictionary
0 otherwise
Label space = $\{1,2\}$
Article$^{(i)}$ = $\begin{cases} 1 \text{ if economist} \\ 2 \text{ if onion} \end{cases}$

Applying Naive Bayes:
2 assumptions
1) data is i.i.d
2) for each pair of features X_i, X_j
w/ $i \neq j$ we assume conditional independence given label Y (this is the Naive Bayes assumption)

Naive Bayes classification rule:
$\hat{Y} = \underset{y}{\text{argmax}}\ P(Y=y|X=x)$

Can be rewritten using Bayes Rule
$\hat{Y} = \underset{y}{\text{argmax}}\ \frac{P(X=x|Y=y)P(Y=y)}{P(X=x)}$ (Bayes Rule)
$\hat{Y} = \underset{y}{\text{argmax}}\ P(X=x|Y=y)P(Y=y)$
$= \underset{y}{\text{argmax}}\ P(X=X_1 \cdots X_{|V|}|Y=y)\ P(Y=y)$
$= \underset{y}{\text{argmax}} \left(\prod_{i=1}^{|V|} P(X=X_i|Y=y)\right) P(Y=y)$ (conditional indep)

A note about # params needed:
- Without conditional indep assumptions
$2(2^{|V|}-1)$ params
w/ conditional independence assumptions
$\hookrightarrow 2n$

a) Compute the log Product(x) where x is a vector of numbers in log space. and returns the product of those numbers in log space
↳ The product of numbers who are already in log space is simply the sum of all vector components
i.e $\prod_i p_i = \sum_i \log p_i$

Training NB.
b). Compile func[D] = NB_XGiven4Y(XTrain, yTrain)
Output θ: $2 \times |V|$ matrix where for any word index $w \in \{1, \ldots |V|\}$ and class index $y \in \{1,2\}$ the entry $\theta(y,w)$ is the map estimate of $P(X_w=1|Y=y)$ w/ a Beta(2,1) distribution
$\hat{\theta}_{map} = \frac{\#(\text{observed } i\text{'s} \wedge Y=y) + \text{hallucinated } i\text{'s}}{\#(Y=y) + \text{hallucinated } i\text{'s} + \text{hallucinated } 0\text{'s}}$

J 1). Seperate the XTrain examples by classes
J 2) $P(X_w=1|Y=1) = \frac{\text{sum}(X\text{Train}_{y_1} \text{ examples}) + 1}{\text{size of } X\text{Train}_{y_1} \text{ examples} + 1}$
$P(X_w=1|Y=2) = \frac{\text{sum}(X\text{Train}_{y_2} \text{ examples}) + 1}{\text{size of } X\text{Train}_{y_2} \text{ examples} + 1}$

c Theta for $p(y=1)$ is the empirical count of labels, $y=1$ / size of vector y.
$\frac{\#\{Y=1\}}{|Y|}$

d) Create the function [yHAT] = NB_Classify(D, p, X)
X is a $m \times V$ matrix containing m feature vectors
$\hat{y}$ = $m \times 1$ vector of predicted class labels
$\hat{y}^{(i)}$ = the predicted label of feature vector x

The NB optimal classifier is
$\hat{Y} = \underset{y \in Y}{\text{argmax}}\ P(Y|D)$
$= \underset{y \in Y}{\text{argmax}}\ P(D|Y)P(Y)$ (Bayes Rule)
$= \underset{y \in Y}{\text{argmax}}\ P(X_1 \cdots X_{|V|}|Y=y)\ P(Y=y)$
$= \underset{y \in Y}{\text{argmax}} \prod_{i=1}^{|V|} P(X_i|Y=y)\ P(Y=y)$ (NB assumption: conditional independencies)
$= \underset{y \in Y}{\text{argmax}} \sum_{i=1}^{|V|} P(X_i|Y=y) + P(Y=y)$ (Log transformation)

1) We've previously calculated $P(X_i|Y=y)$
2) We apply PMF Bernoulli$(X_i|\theta)$ where $\theta = P(X_i|Y=y)$
3) We take the log Prod of the log of the result of above and the log of the prior
4). $\hat{y}^{(i)}$ = the label w/ maximum probability

e) Classification error: $\frac{\#\text{ of labels that agree}}{\text{number of test examples}}$

g) The test error is higher than the train error. This is because we trained the classifier on the training data & Testing the classifier using the training data effectively amounts to having memorized the data.

h). Retraining the classifier using the smaller training set & attempting to reclassify the test samples produces a lower success fail classification rate than using the larger training set. This is intuitive yet we look at the params of the classifier
$P(X_i|Y=y)$ and $P(Y=y)$
certainly larger data sets are more likely to produce accurate estimates of the 2 params. Also since we are using the MLE to estimate the params we are likely to get inaccurate measurements. As a result, the error has the potential to skew the classifier towards the incorrect classifier.

i) Find the m=5 most popular words per label:
Looking at the results for the m=5 most popular words for each label we see
economist most popular words are more economic in nature (e.g reform, Servant, member)
This is appropriate for articles that mostly deal with topics related to the political economy.