CSE 242 - Homework 3

Abad, Ferdinand
fabad@ucsc.edu

Chen, Kejun
kchen158@ucsc.edu

29 October 2019

Problem 1. Consider using Naive Bayes to estimate if a student will be an honor student (**H**) or normal student (**N**) in college based on their high school performance. Each instances have two measurements: the student's high school GPA (a real number) and whether or not the student took any AP courses (a boolean value, yes=1, no=0). Based on the following training data, create (by hand and/or calculator) a Naive Bayes prediction rule using gaussians to estimate the conditional probability density of a high school GPAs given the class (**H** or **N**) and a Bernoulli distribution for the AP probability . (I know that Gaussians may not fit this problem well, but use them anyway).

Recall that Naive Bayes makes the simplifying assumption that the features are independent given the class (see pages 45-46 Bishop), so (for example)

$$\mathbf{P}[\text{GPA} = 3.2, \text{AP} = \text{yes}|\text{type} = \text{H}] = \mathbf{P}[\text{GPA} = 3.2|\text{type} = \mathbf{H}]\mathbf{P}[\text{AP} = \text{yes}|\text{type} = \text{H}].$$

class	AP	GPA
H	yes	4.0
H	yes	3.7
H	no	2.5
N	no	3.8
N	yes	3.3
N	yes	3.0
N	no	3.0
N	no	2.7
N	no	2.2

Use maximum likelihood estimation (do *not* us Laplace estimates or the unbiased variance) for the class probabilities and distribution of the two features conditioned on the two classes. Give the means and variance of the gaussians you found for the GPA (i.e. for GPA given H and for GPA give N).

Describe your prediction rule in the following form:

If AP courses are taken, then predict **H** if the GPA is between, . . . , and
If AP courses are not taken, then predict **H** if the GPA is between . . .

(It is probably easier to get this description if you take logarithms, 4 digits of precision should suffice.)

Solution : First, we consider the AP courses are taken, in order to solve the problem, we can calculate the balance point of GPA where the two Gaussians for predicting each class cross. It should satisfy the following equation:

$$P(\mathbf{H}|AP = \text{yes}, \text{GPA}) = P(\mathbf{N}|AP = \text{yes}, \text{GPA}).$$

Using Naive Bayes theory, it can be rewritten as:

$$P(\mathbf{H}) \cdot P(\text{AP} = \text{yes}|\mathbf{H}) \cdot P(\text{GPA}|\mathbf{H}) = P(\mathbf{N}) \cdot P(\text{AP} = \text{yes}|\mathbf{N}) \cdot P(\text{GPA}|\mathbf{N}).$$

According to the table, we can easily get the following equations:

$$\begin{aligned} P(\text{class} = \mathbf{H}) &= \frac{1}{3} \\ P(\text{class} = \mathbf{N}) &= \frac{2}{3} \\ P(\text{AP} = \text{yes}|\mathbf{H}) &= \frac{2}{3} \\ P(\text{AP} = \text{yes}|\mathbf{N}) &= \frac{1}{3} \\ P(\text{AP} = \text{no}|\mathbf{H}) &= \frac{1}{3} \\ P(\text{AP} = \text{no}|\mathbf{N}) &= \frac{2}{3}. \end{aligned}$$

According to the assumptions, the conditional probability density denoted f, satisfies the Gaussian Distribution, we can get:

$$\begin{aligned} P(\text{GPA}|\mathbf{H}) &= \frac{1}{\left(2\pi\sigma_{\mathbf{H}}^2\right)^{\frac{1}{2}}} \exp\Big\{ -\frac{1}{2\sigma_{\mathbf{H}}^2} \cdot (x - \mu_{\mathbf{H}})^2 \Big\} \\ P(\text{GPA}|\mathbf{N}) &= \frac{1}{\left(2\pi\sigma_{\mathbf{N}}^2\right)^{\frac{1}{2}}} \exp\Big\{ -\frac{1}{2\sigma_{\mathbf{N}}^2} \cdot (x - \mu_{\mathbf{N}})^2 \Big\}. \end{aligned}$$

Using maximum likelihood estimation, we can calculate each respective σ^2 and μ values for

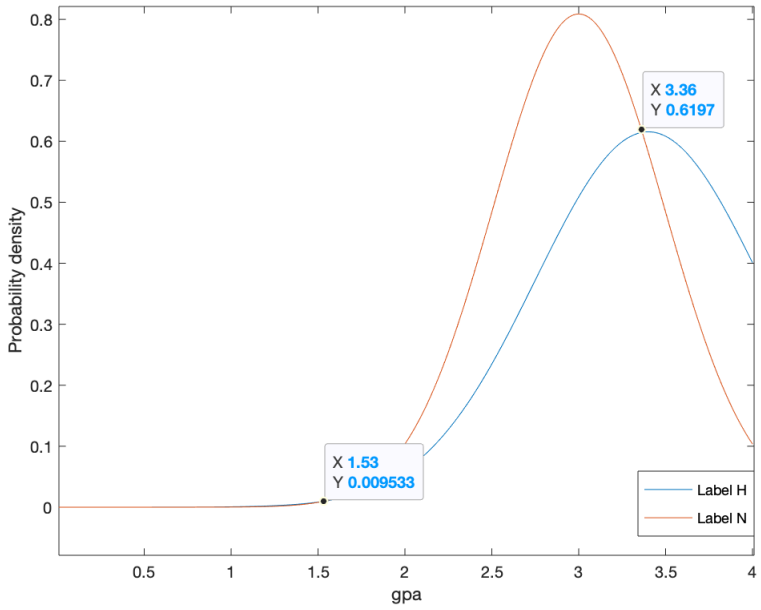
GPA and get the following

$$\mu_{\mathbf{H}} = \frac{(4+3.7+2.5)}{3} = 3.4000$$
$$\mu_{\mathbf{N}} = \frac{(3.8+3.3+3+3+2.7+2.2)}{6} = 3.0000$$
$$\sigma^2_{\mathbf{H}} = \frac{((4-3.4)^2+(3.7-3.4)^2+(2.5-3.4)^2)}{3} = 0.4200$$
$$\sigma^2_{\mathbf{N}} = \frac{((3.8-3)^2+(3.3-3)^2+0+0+(2.7-3)^2+(2.2-3)^2)}{6} = 0.2433.$$

Replace what we have known with the second equation, we can get:

$$P\text{GPA}|\mathbf{H}) = P(\text{GPA}|\mathbf{N})$$
$$\frac{1}{\left(2\pi\sigma^2_{\mathbf{H}}\right)^{\frac{1}{2}}}\exp\left\{-\frac{1}{2\sigma^2_{\mathbf{H}}}\cdot(x-\mu_{\mathbf{H}})^2\right\} = \frac{1}{\left(2\pi\sigma^2_{\mathbf{N}}\right)^{\frac{1}{2}}}\exp\left\{-\frac{1}{2\sigma^2_{\mathbf{N}}}\cdot(x-\mu_{\mathbf{N}})^2\right\}.$$
$$(\sigma^2_{\mathbf{H}}-\sigma^2_{\mathbf{N}})x^2 - 2(\sigma^2_{\mathbf{H}}\mu_{\mathbf{N}}-\sigma^2_{\mathbf{N}}\mu_{\mathbf{H}})x + \sigma^2_{\mathbf{H}}\mu^2_{\mathbf{N}} - \sigma^2_{\mathbf{N}}\mu^2_{\mathbf{H}} - \ln\frac{\sigma_{\mathbf{H}}}{\sigma_{\mathbf{N}}}\cdot 2\sigma^2_{\mathbf{N}}\sigma^2_{\mathbf{H}} = 0$$
$$x = \frac{2(\sigma^2_{\mathbf{H}}\mu_{\mathbf{N}}-\sigma^2_{\mathbf{N}}\mu_{\mathbf{H}}) \pm \sqrt{(2(\sigma^2_{\mathbf{H}}\mu_{\mathbf{N}}-\sigma^2_{\mathbf{N}}\mu_{\mathbf{H}}))^2 - 4(\sigma^2_{\mathbf{H}}-\sigma^2_{\mathbf{N}})(\sigma^2_{\mathbf{H}}\mu^2_{\mathbf{N}}-\sigma^2_{\mathbf{N}}\mu^2_{\mathbf{H}} - \ln\frac{\sigma_{\mathbf{H}}}{\sigma_{\mathbf{N}}}\cdot 2\sigma^2_{\mathbf{N}}\sigma^2_{\mathbf{H}})}}{2(\sigma^2_{\mathbf{H}}-\sigma^2_{\mathbf{N}})}$$

Solve the equation, we can get $x1 = 1.5331$ or $x2 = 3.3656$, we can know, if GPA fall inside (3.3656,4] or [0,1.5331), we will predict label honor. If GPA fall inside [1.5331,3.3656], we will predict normal.



Then we consider the second problem, when the AP courses are not taken.

$$P(\mathbf{H}|AP = \text{no}, \text{GPA}) = P(\mathbf{N}|AP = \text{no}, \text{GPA}).$$

Using Naive Bayes theory, it can be rewritten as:

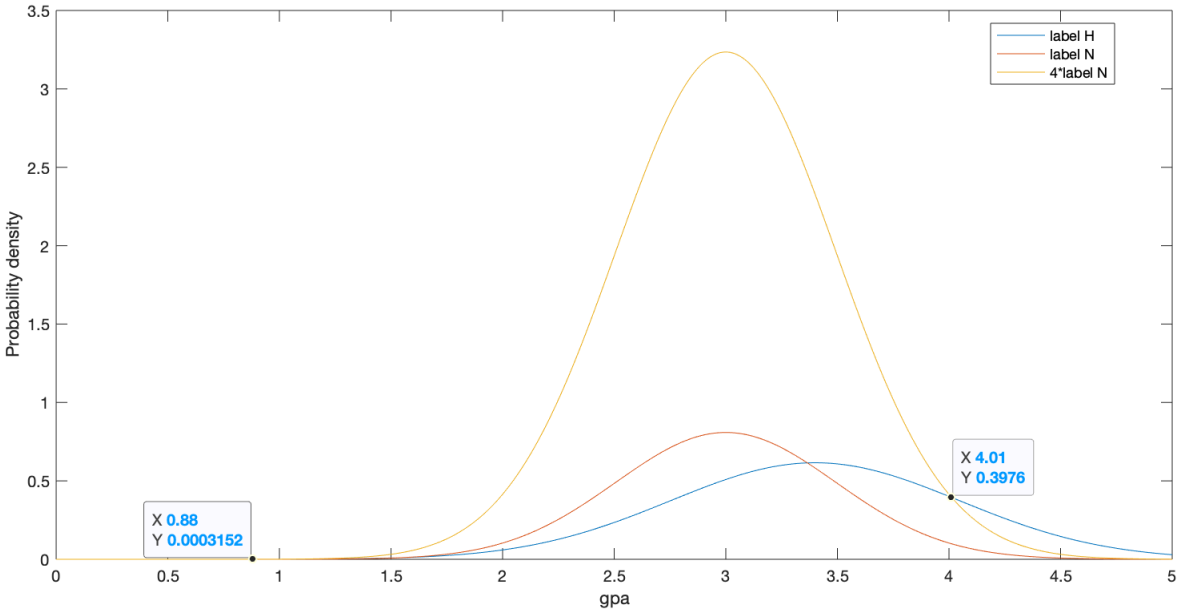
$$P(\text{GPA}|\mathbf{H}) = 4 \cdot P(\text{GPA}|\mathbf{N})$$

$$\frac{1}{\left(2\pi\sigma_{\mathbf{H}}^2\right)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma_{\mathbf{H}}^2} \cdot (x-\mu_{\mathbf{H}})^2\right\} = \frac{4}{\left(2\pi\sigma_{\mathbf{N}}^2\right)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma_{\mathbf{N}}^2} \cdot (x-\mu_{\mathbf{N}})^2\right\}.$$

$$(\sigma_{\mathbf{H}}^2 - \sigma_{\mathbf{N}}^2)x^2 - 2(\sigma_{\mathbf{H}}^2\mu_{\mathbf{N}} - \sigma_{\mathbf{N}}^2\mu_{\mathbf{H}})x + \sigma_{\mathbf{H}}^2\mu_{\mathbf{N}}^2 - \sigma_{\mathbf{N}}^2\mu_{\mathbf{H}}^2 - \ln\frac{4\cdot\sigma_{\mathbf{H}}}{\sigma_{\mathbf{N}}} \cdot 2\sigma_{\mathbf{N}}^2\sigma_{\mathbf{H}}^2 = 0$$

$$x = \frac{2(\sigma_{\mathbf{H}}^2\mu_{\mathbf{N}} - \sigma_{\mathbf{N}}^2\mu_{\mathbf{H}}) \pm \sqrt{(2(\sigma_{\mathbf{H}}^2\mu_{\mathbf{N}} - \sigma_{\mathbf{N}}^2\mu_{\mathbf{H}}))^2 - 4(\sigma_{\mathbf{H}}^2 - \sigma_{\mathbf{N}}^2)(\sigma_{\mathbf{H}}^2\mu_{\mathbf{N}}^2 - \sigma_{\mathbf{N}}^2\mu_{\mathbf{H}}^2 - \ln\frac{4\cdot\sigma_{\mathbf{H}}}{\sigma_{\mathbf{N}}} \cdot 2\sigma_{\mathbf{N}}^2\sigma_{\mathbf{H}}^2)}}{2(\sigma_{\mathbf{H}}^2 - \sigma_{\mathbf{N}}^2)}$$

Solve the equation, we can get $x3 = 0.8865$ or $x4 = 4.0122$, we can know, if GPA fall inside [0,0.8865), we will predict label honor. If GPA fall inside [0.8865,4], we will predict normal.



Problem 2. Nearest neighbor calculation. Assume that examples are drawn from the uniform density on the unit square and the label t is "+" with probability $\frac{2}{3}$ and labelled "" with probability $\frac{1}{3}$, independent of the location drawn. The Bayes-optimal hypothesis (minimizing the probability of a mistake) always predicts "+" and has error rate $\frac{1}{3}$. Assume that we draw a large sample from a continuous density (so we can ignore the chance that any point is repeated, or there are two points the same distance away from the point being predicted on) and use 3-Nearest-Neighbor to predict (i.e. to predict on a new point, first find the three nearest points in the sample and use the most common label of the those three training point as the prediction).

What is the (average over samples) error of this 3-Nearest-Neighbor algorithm? What is the average error rate when the noise rate (probability that labels are $-$) is $\frac{1}{10}$?

In more detail, the statistical experiment defining the probabilities to be analyzed is:

- Draw the locations of the training sample
- Set the labels (t-values) for each training point
- Draw a single test location to predict on and label t to be predicted

Note that the labels of points in the sample are independent of their locations, and that the error rate is the chance that the predicted doesn't match the label t of the test example.

Solution ccording to the problem description, we have known the probability of label " + " and " − " are independent of the location drawn and the examples are drawn from density on the unit square. Then we can calculate the probability of the following situation:
1) When the three nearest neighbor points are all labeled " + ", the test point will be labeled " + ".

$$P(+++) = \frac{2}{3} \cdot \frac{2}{3} \cdot \frac{2}{3} = \frac{8}{27}$$

2) When the two of nearest neighbor points are all labeled " + " and one of nearest neighbor point are labeled " − ", the test point will be labeled " + ".

$$P(++-) = \frac{2}{3} \cdot \frac{2}{3} \cdot \frac{1}{3} \cdot 3 = \frac{12}{27}$$

3)When the one of nearest neighbor points are all labeled " + " and two of nearest neighbor point are labeled " − ", the test point will be labeled " − ".

$$P(+--) = \frac{2}{3} \cdot \frac{1}{3} \cdot \frac{1}{3} \cdot 3 = \frac{6}{27}$$

4) When the three nearest neighbor points are all labeled " − ", the test point will be labeled " _ "

$$P(---) = \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{3} = \frac{1}{27}$$

then we can get the probability table:

Label	Probability	Prediction
$+++$	$\frac{8}{27}$	$+$
$++-$	$\frac{12}{27}$	$+$
$+--$	$\frac{6}{27}$	$-$
$---$	$\frac{1}{27}$	$-$

The prediction is wrong in two situations. First, the test point labeled "$+$" but predicted "$-$". Second, the test point labeled "$-$" but predicted "$+$". When test point has label "$+$" with probability $\frac{2}{3}$ and label "$-$" with probability $\frac{1}{3}$, note that we draw test data and training data independent, we can get the error rate:

$$\begin{aligned} P(error) &= P(\text{label is "}+\text{"}) \cdot P(\text{label predicted "}-\text{"}) + P(\text{label is "}-\text{"}) \cdot P(\text{label predicted "}+\text{"}) \\ &= \frac{2}{3} \cdot (\frac{6}{27} + \frac{1}{27}) + \frac{1}{3} \cdot (\frac{8}{27} + \frac{12}{27}) \\ &= \frac{34}{81} \approx 0.4198 \end{aligned}$$

The second problem is when the noise rate is $\frac{1}{10}$, and we can do similarly as the first problem:
1) When the three nearest neighbor points are all labeled "$+$", the test point will be labeled "$+$".

$$P(+++) = \frac{9}{10} \cdot \frac{9}{10} \cdot \frac{9}{10} = \frac{729}{1000}$$

2) When the two of nearest neighbor points are all labeled "$+$" and one of nearest neighbor point are labeled "$-$", the test point will be labeled "$+$".

$$P(++-) = \frac{9}{10} \cdot \frac{9}{10} \cdot \frac{1}{10} \cdot 3 = \frac{243}{1000}$$

3)When the one of nearest neighbor points are all labeled "$+$" and two of nearest neighbor point are labeled "$-$", the test point will be labeled "$-$".

$$P(+--) = \frac{9}{10} \cdot \frac{1}{10} \cdot \frac{1}{10} \cdot 3 = \frac{27}{1000}$$

4) When the three nearest neighbor points are all labeled " $-$ ", the test point will be labeled " $-$ "

$$P(---) = \frac{1}{10} \cdot \frac{1}{10} \cdot \frac{1}{10} = \frac{1}{1000}$$

then we can get the probability table:

Label	Probability	Prediction
$+++$	$\frac{729}{1000}$	$+$
$++-$	$\frac{243}{1000}$	$+$
$+--$	$\frac{27}{1000}$	$-$
$---$	$\frac{1}{1000}$	$-$

The prediction is wrong in two situations. First, the test point labeled " $+$ " but predicted " $-$ ". Second, the test point labeled " $-$ " but predicted " $+$ ". When test point has label " $+$ " with probability $\frac{9}{10}$ and label " $-$ " with probability $\frac{1}{10}$, note that we draw test data and training data independent, we can get the error rate:

$$\begin{aligned} P(error) &= P(\text{label is " + "}) \cdot P(\text{label predicted " - "}) + P(\text{label is " - "}) \cdot P(\text{label predicted " + "}) \\ &= \frac{9}{10} \cdot (\frac{27}{1000} + \frac{1}{1000}) + \frac{1}{10} \cdot (\frac{729}{1000} + \frac{243}{1000}) \\ &= 0.1224 \end{aligned}$$

As we can see, from the first problem, the error rate is much larger than Bayes-optimal hypothesis, but when the noise rate is $\frac{1}{10}$, the error is still higher than Bayes-optimal hypothesis but they are close.

Problem 3. Test classification. Download the Enron-spam dataset from `http://www.aueb.gr/users/ion/data/enron-spam/`. Use the pre-processed datasets Enron1, ..., Enron5 as traiing data and Enron6 as test data. Implement Naive Bayes algorithm that we discussed in class (one reference is `http://nlp.stanford.edu/IRbook/pdf/13bayes.pdf`), or become familiar with a tool (like that provided by scikit-learn) and use that. Remember to perform all calculations on the log scale to prevent underflow. For the base question

- Report the accuracy on the test set
- How do you account for the different prior probabilities for spam and ham?
- Implement versions with and without Laplace smoothing (adding a fictitious observation of each word to each class)? How does the performance of the classifier change when Laplace smoothing is added?
- What are the most discriminative words based on the learned probabilities?

Advanced exploration (optional, 1pt): try additional input processing, like removing common stop-words (see for examples `https://www.ranks.nl/stopwords`), trying stemming, or other tech

Solution : To solve the problem, we solve the problem based on Naive Bayes theory and then we mainly used Sklearn and NLTK packet in Python to put it into practice.

First, we will introduce how to classify a new email. The two labels in the data set are spam and ham. For the words in the training set, it is easy to get the individual frequencies for each word in the emails. We also assume their appearance is independent for simplicity. In the test set, they are denoted as w_{test}, which may contain words in the training set, denoted w_i, $i = 1, 2, 3 \ldots$, and out of the training set. It it is not in the training set, the posterior probability is zero, which is not helpful. To avoid this, we add Laplace smoothing and compared it with the results without Laplace smoothing. Besides, we can also know the prior probability about spam and ham according to the training set, that is the differences in the labels therefore the prior is just the probability of each class. Some words in the test set may appear several times, and we can also know their repeat times in the test set. After ignoring the denominator which is a constant and won't affect our final prediction results, our most basic equation is:

$$P(ham|w_{test}) = P(w_1|ham)^{repeat1} \cdot P(w_2|ham)^{repeat2} \cdot \ldots \cdot P(w_n|ham)^{repeat_n} \cdot P(ham)$$
$$P(spam|w_{test}) = P(w_1|spam)^{repeat1} \cdot P(w_2|spam)^{repeat2} \cdot \ldots \cdot P(w_n|spam)^{repeat_n} \cdot P(spam)$$

We need compare the two results, and choose the bigger one as our prediction results. In practice, we use the $log[P()]$ instead of $P()$ to calculate the probability to prevent underflow when dealing with small numbers.

Next, we will introduce how can we use python to put this process into practice.

First, we need to load the .txt file from the website and change them into the regular format we need for learning. For simplicity the ham and spam files from enron1-enron5 were combined into one training directory. Our program will read the emails and create a Pandas data frame table of our training data. Each row will represent each individual email. The table will have two columns where the first one corresponds to the contents of the email and the second column will represent the true label for that file.

Second, we consider preprocessing to remove punctuation, capital letters and stop words. Stop word may appear many time in the text, but it doesn't make much sense for our prediction, such as and, the, or .... Python provides strong packet named NLTK for us to deal with stop words in NLP. Later, we find we get quite different top 10 discriminative words in two situations where we keep the stop words or choose to remove them while training the models. Now we need consider what is the most discriminative words? We shouldn't only consider the term frequency in each class. For example, the word appears most in the spam may also appear many times in the ham. In this case, when the word appear in the test email, it is hard to say it belongs to spam. Considering an extreme situation, a word never appear in the spam email and only appear in the ham email, so whenever it appears, the system will predict it into a ham email. To be more specific, the most discriminative words are given by the biggest difference in posterior instead of the posterior itself. We will show the most frequency words and most discriminative words later.

Third, it is the most important part because we should learn the conditional probability in the training set. That is to say, we need draw the features in the training set, to be specific, we need to know the probability of certain word appear in each class, in fact, we need to count the number of the words in the training text. We use CountVectorizer function form Sklearn in the python, it will return both the word dictionary and the frequency.

Fourth, we need classify the test email and get the prediction error. We use Naive Bayes to make prediction so we use the MultinomialNB classifier function in the Sklearn Module. To begin with, we need do the same data preprocess as the training data. Then, we can get the prior probability.

Finally, we get the following results:

Accuracy With Stop Words with Laplace Smoothing: 0.9830

Accuracy With Stop Words without Laplace Smoothing: 0.9792

Accuracy No Stop Words with Laplace Smoothing: 0.9822

Accuracy No Stop Words without Laplace Smoothing: 0.9782

We can know with Laplace Smoothing, our prediction accuracy increases. Note the error consists with two types: ham but predicted spam and spam but predicted ham. In fact the costs are quite different for the two types of error, but here we only focus on the overall error.

We can see from our result that removing stop words barely changed our model's accuracy. This is because since our vocabulary dictionary is made up of only $136,031$ non-stop words. We need a bigger dictionary of non-stop words in order for our accuracies to improve more.

Prior probability:

$$\begin{aligned} P(\text{ham}) &= \frac{15045}{27716} \\ &= 0.5428 \\ P(\text{spam}) &= \frac{12671}{27716} \\ &= 0.4571 \end{aligned}$$

Now we can analyze the word frequencies of the words in our corpus. First, we will look at term frequency for each word and get the following results:

Ham	Spam
the	the
to	to
and	and
of	of
a	a
in	you
enron	in
for	your
on	for
i	this

Table 1: Top 10 Frequent Words With Stop Words

Ham	Spam
enron	com
etc	1
hou	3
2001	company
2000	2
1	http
please	e
would	email
company	information
com	5

Table 2: Top 10 Frequent Words Without Stop Words

From our code, the results show that the top 10 most discriminative words are the same whether the training set contains stop words or not. This is because stop words are frequent

in both the ham and spam emails therefore the difference of the log of their conditional probabilities will be extremely low have Thus, we get the following table:

Ham	Spam
enron	pills
kaminski	viagra
dynegy	computron
ect	cialis
ees	nbsp
ena	photosho
dbcaps	width
hourahead	href
fastow	voip
mmbtu	paypal

Table 3: Top 10 Discriminative Words