All of the Netflix slides from last Wednesday’s lecture are presented here.
Predicting user TV preferences

Will a user like the Pokémon TV series?

Observe indicator variables $X = (X_1, X_2)$:

- $X_1 = 1$: “likes Star Wars”
- $X_2 = 1$: “likes Harry Potter”

Output $Y$ indicator:

- $Y = 1$: “likes Pokémon”

Predict $\hat{Y} = \arg\max_{Y \in \{0, 1\}} P(Y | X)$

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Spring 2021
Predicting user TV preferences

Which probabilities do you need to estimate? How many are there?

- Brute Force Bayes (strawman, without NB assumption)
- Naïve Bayes

During training, how to estimate the prob $\hat{P}(X_1 = 1, X_2 = 1 | Y = 0)$ with MLE? with Laplace?

- Brute Force Bayes
- Naïve Bayes
Predicting user TV preferences

Which probabilities do you need to estimate?
How many are there?

• Brute Force Bayes (strawman, without NB assumption)

• Naïve Bayes

During training, how to estimate the prob $P(X_1 = 1, X_2 = 1 | Y = 0)$ with MLE? with Laplace?

• Brute Force Bayes

• Naïve Bayes
(strawman brute force) Multinomial MLE and MAP

Model: Multinomial, \( m \) outcomes:
\( p_j \) probability of outcome \( j \)

Observe:
\( n_j = \# \) of trials with outcome \( j \)
Total of \( \sum_{j=1}^{m} n_j \) trials

**MLE**
\[
\hat{p}_j = \frac{n_j}{\sum_{j=1}^{m} n_j}
\]

**MAP**
\[
\hat{p}_j = \frac{n_j + 1}{\sum_{j=1}^{m} n_j + m}
\]

\( \mathbb{P}(X_1 = 1, X_2 = 1 \mid Y = 1) \)

<table>
<thead>
<tr>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( Y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Training data

\( \mathbb{P}(X_1 = 1, X_2 = 1 \mid Y = 0) = \frac{\# \text{count}(X_1 = 1 \land X_2 = 1 \land Y = 1)}{\# \text{count}(Y = 0)} \)

\( \mathbb{P}(X_1 = 1, X_2 = 1 \mid Y = 0) = \frac{\# \text{count}(X_1 = 1 \land X_2 = 1 \land Y = 0)}{\# \text{count}(Y = 0) + 4} \)

\( \mathbb{P}(X_1 = 1, X_2 = 1 \mid Y = 0) \Rightarrow \# \text{count}(Y = 1) \)
(Naïve Bayes) Multinomial MLE and MAP

Model: Multinomial, $m$ outcomes: $p_j$ probability of outcome $j$

Observe: $n_j = \#$ of trials with outcome $j$
Total of $\sum_{j=1}^{m} n_j$ trials

\[ \hat{p}_j = \frac{n_j}{\sum_{j=1}^{m} n_j} \]

MLE

\[ \hat{p}_j = \frac{n_j + 1}{\sum_{j=1}^{m} n_j + m} \]

Laplace estimate (MAP w/Laplace smoothing)

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Training data

\[ \hat{P}(X_1 = 1 \mid Y = 0) \]
\[ \hat{P}(X_2 = 1 \mid Y = 0) \]

\[ \hat{P}(X_1 = 1 \cap Y = 0) \]
\[ \hat{P}(X_1 = 0 \cap Y = 0) \]
NETFLIX

and Learn

naively
Ex 1. Naïve Bayes Classifier (MLE)

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)
\]

∀i: \( \hat{P}(X_j = 1|Y = 0), \hat{P}(X_j = 0|Y = 0) \), \ Use MLE or Laplace (MAP) \\
\( \hat{P}(X_j = 1|Y = 1), \hat{P}(X_j = 0|Y = 0), \) \\
\( \hat{P}(Y = 1), \hat{P}(Y = 0) \)

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)
\]
Training: Naïve Bayes for TV shows (MLE)

Observe indicator vars. $X = (X_1, X_2)$:
- $X_1$: “likes Star Wars”
- $X_2$: “likes Harry Potter”

Predict $Y$: “likes Pokémon”

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$Y$</th>
<th>$X_2$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>3</th>
<th>10</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>13</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$X_1$ = 0</th>
<th>$X_1$ = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y = 0$</td>
<td>$Y = 1$</td>
</tr>
</tbody>
</table>

| $P(X_1 = 0 | Y = 0)$ | $P(X_1 = 1 | Y = 0)$ |
|-----------|-----------|
| $P(X_1 = 0 | Y = 1)$ | $P(X_1 = 1 | Y = 1)$ |

1. How many datapoints ($n$) are in our training data?
2. Compute MLE estimates for $\hat{P}(X_1 | Y)$:
### Training: Naïve Bayes for TV shows (MLE)

Observe indicator vars. \( X = (X_1, X_2) \):
- \( X_1 \): “likes Star Wars”
- \( X_2 \): “likes Harry Potter”

Predict \( Y \): “likes Pokémon”

<table>
<thead>
<tr>
<th>( Y )</th>
<th>( X_1 )</th>
<th>( X_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>13</td>
</tr>
</tbody>
</table>

Training data counts

<table>
<thead>
<tr>
<th>( Y )</th>
<th>( X_1 )</th>
<th>( X_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>17</td>
<td>10</td>
</tr>
</tbody>
</table>

1. How many datapoints (\( n \)) are in our training data?

2. Compute MLE estimates for \( \hat{P}(X_1 | Y) \):

<table>
<thead>
<tr>
<th>( Y )</th>
<th>( X_1 )</th>
<th>( X_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 ( \frac{3}{13} ) ( \frac{10}{13} )</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1 ( \frac{4}{17} ) ( \frac{13}{17} )</td>
<td>1</td>
</tr>
</tbody>
</table>
Training: Naïve Bayes for TV shows (MLE)

Observe indicator vars. \( X = (X_1, X_2) \):
- \( X_1 \): “likes Star Wars”
- \( X_2 \): “likes Harry Potter”

Predict \( Y \): “likes Pokémon”

<table>
<thead>
<tr>
<th>( X_1 ) ( Y )</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( X_2 ) ( Y )</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

Training data counts

<table>
<thead>
<tr>
<th>( X_1 ) ( Y )</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5/13 ( \approx 0.38 )</td>
<td>8/13 ( \approx 0.62 )</td>
</tr>
<tr>
<td>1</td>
<td>7/17 ( \approx 0.41 )</td>
<td>10/17 ( \approx 0.59 )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( X_2 ) ( Y )</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>13/30 ( \approx 0.43 )</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>17/30 ( \approx 0.57 )</td>
<td></td>
</tr>
</tbody>
</table>

\[ \frac{3}{12} \approx 0.23 \]
Training: Naïve Bayes for TV shows (MLE)

Observe indicator vars. $X = (X_1, X_2)$:
- $X_1$: “likes Star Wars”
- $X_2$: “likes Harry Potter”

Predict $Y$: “likes Pokémon”

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>Y</th>
<th>$X_2$</th>
<th>Y</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.23</td>
<td>0</td>
<td>0.38</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.24</td>
<td>1</td>
<td>0.41</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0.77</td>
<td>0.62</td>
<td></td>
<td>0.43</td>
</tr>
<tr>
<td>0.76</td>
<td></td>
<td>0.59</td>
<td></td>
<td>0.57</td>
</tr>
</tbody>
</table>

Now that we’ve trained and found parameters, it’s time to classify new users!
Ex 1. Naïve Bayes Classifier (MLE)

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)
\]

Training:
\[
\forall i: \hat{P}(X_j = 1|Y = 0), \hat{P}(X_j = 0|Y = 0), \quad \text{Use MLE or Laplace (MAP)}
\]
\[
\hat{P}(X_j = 1|Y = 1), \hat{P}(X_j = 0|Y = 0), \quad \hat{P}(Y = 1), \hat{P}(Y = 0)
\]

Testing:
\[
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)
\]
Testing: Naïve Bayes for TV shows (MLE)

Observe indicator vars. $X = (X_1, X_2)$:

<table>
<thead>
<tr>
<th></th>
<th>$X_1$</th>
<th></th>
<th>$X_2$</th>
<th></th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.23</td>
<td>0.77</td>
<td>0.38</td>
<td>0.62</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.24</td>
<td>0.76</td>
<td>0.41</td>
<td>0.59</td>
<td>1</td>
</tr>
</tbody>
</table>

Predict $Y$: “likes Pokémon”

Suppose a new person “likes Star Wars” ($X_1 = 1$) but “dislikes Harry Potter” ($X_2 = 0$). Will they like Pokemon? Need to predict $Y$:

$$\hat{Y} = \arg\max_{y=\{0, 1\}} \hat{P}(X|Y)\hat{P}(Y) = \arg\max_{y=\{0, 1\}} \hat{P}(X_1|Y)\hat{P}(X_2|Y)\hat{P}(Y)$$

If $Y = 0$:

$$\hat{P}(X_1 = 1|Y = 0)\hat{P}(X_2 = 0|Y = 0)\hat{P}(Y = 0) = 0.77 \cdot 0.38 \cdot 0.43 = 0.126$$

If $Y = 1$:

$$\hat{P}(X_1 = 1|Y = 1)\hat{P}(X_2 = 0|Y = 1)\hat{P}(Y = 1) = 0.76 \cdot 0.41 \cdot 0.57 = 0.178$$

Since term is greatest when $Y = 1$, predict $\hat{Y} = 1$
Ex 2. Naïve Bayes Classifier (MAP)

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)
\]

∀i: \( \hat{P}(X_j = 1|Y = 0) \), \( \hat{P}(X_j = 0|Y = 0) \), \( \hat{P}(X_j = 1|Y = 1) \), \( \hat{P}(X_j = 0|Y = 0) \), \( \hat{P}(Y = 1) \), \( \hat{P}(Y = 0) \)

Use MLE or Laplace (MAP)

Testing

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)
\]

(note the same as before)
Training: Naïve Bayes for TV shows (MAP)

Observe indicator vars. $X = (X_1, X_2)$:
- $X_1$: “likes Star Wars”
- $X_2$: “likes Harry Potter”

Predict $Y$: “likes Pokémon”

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y$</td>
<td>10</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$X_2$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y$</td>
<td>8</td>
<td>5</td>
</tr>
</tbody>
</table>

Training data counts

What are our MAP estimates using Laplace smoothing for $\hat{P}(X_j | Y)$?

A. $\frac{\#(X_j=x, Y=y)}{\#(Y=y)}$
B. $\frac{\#(X_j=x, Y=y) + 1}{\#(Y=y) + 2}$
C. $\frac{\#(X_j=x, Y=y) + 1}{\#(Y=y) + 4}$
D. other
Training: Naïve Bayes for TV shows (MAP)

Observe indicator vars. $X = (X_1, X_2)$:
- $X_1$: “likes Star Wars”
- $X_2$: “likes Harry Potter”

Predict $Y$: “likes Pokémon”

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$Y$</th>
<th>$X_2$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Training data counts:

<table>
<thead>
<tr>
<th>X1</th>
<th>Y</th>
<th>X2</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td>7</td>
</tr>
</tbody>
</table>

$\hat{Y} = \arg \max_{Y \in \{0, 1\}} \left( \prod_{i=1}^{m} \hat{P}(X_i|Y) \right) \hat{P}(Y)$

In practice:
- We use Laplace for $\hat{P}(X_j|Y)$ in case some events $X_j = x_j$ don’t appear
- We don’t use Laplace for $\hat{P}(Y)$, because all class labels should appear reasonably often

\[\frac{3}{13} + \frac{3+1}{15+2} = \frac{4}{15} \approx 0.27\]

\[\frac{5}{13} = \frac{6}{15}\]
Naïve Bayes Model is a Bayesian Network

\[
P(X|Y) = \prod_{j=1}^{m} P(X_j|Y)
\]

Which Bayesian Network encodes this conditional independence?

A. \[
\begin{array}{c}
Y \\
X_1 \quad X_2 \quad \ldots \quad X_n
\end{array}
\]

B. \[
\begin{array}{c}
X_1 \quad X_2 \quad \ldots \quad X_n \\
Y
\end{array}
\]

\(X_i\) are conditionally independent given \(Y\)
**Naïve Bayes Model is a Bayesian Network**

Naïve Bayes Assumption

\[
P(X|Y) = \prod_{i=1}^{m} P(X_i|Y) \quad \Rightarrow \quad P(X, Y) = P(Y) \prod_{j=1}^{m} P(X_j|Y)
\]

Which Bayesian Network encodes this conditional independence?

A. \( Y \)
   \( \cdots \)
   \( X_1 \)  \( X_2 \)  \( \cdots \)  \( X_n \)

B. \( X_1 \)
   \( \downarrow \)
   \( \cdots \)
   \( \downarrow \)
   \( Y \)

\( X_i \) are conditionally independent given parent \( Y \)
and Learn naively
What is Bayes doing in my mail server?

Let’s get Bayesian on your spam:

Content analysis details:  (49.5 hits, 7.0 required)
0.9 RCVD_IN_PBL    RBL: Received via a relay in Spamhaus PBL  
[93.40.189.29 listed in zen.spamhaus.org]
1.5 URIBL_WS_SURBL Contains an URL listed in the WS SURBL blocklist  
[URIs: recragas.cn]
5.0 URIBL_JP_SURBL Contains an URL listed in the JP SURBL blocklist  
[URIs: recragas.cn]
5.0 URIBL_OB_SURBL Contains an URL listed in the OB SURBL blocklist  
[URIs: recragas.cn]
5.0 URIBL_SC_SURBL Contains an URL listed in the SC SURBL blocklist  
[URIs: recragas.cn]
2.0 URIBL_BLACK Contains an URL listed in the URIBL blacklist  
[URIs: recragas.cn]
8.0 BAYES_99    BODY: Bayesian spam probability is 99 to 100%  
[score: 1.0000]
Ex 3. Naïve Bayes Classifier $(m, n \text{ large})$

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{i=1}^{m} \hat{P}(X_i|Y) \right) \hat{P}(Y)
\]

**Training**

\[
\forall i: \hat{P}(X_i) = 0 \quad \text{or} \quad \hat{P}(X_i|Y) = 0, \quad \text{Use MLE or Laplace (MAP)}
\]

**Testing**

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{i=1}^{m} \hat{P}(X_i|Y) \right) \hat{P}(Y)
\]

What changes are necessary?
Email classification

**Goal**
Based on email content \( X \), predict if email is spam or not.

**Features**
Consider a lexicon \( m \) words (for English: \( m \approx 100,000 \)).

\[
X = (X_1, X_2, ..., X_m), \ m \text{ indicator variables}
\]

\( X_j = 1 \) if word \( j \) appeared in document

**Output**
\( Y = 1 \) if email is spam

Note: \( m \) is huge. Make Naïve Bayes assumption:

\[
P(X_{\text{spam}}) = \prod_{j=1}^{m} P(X_j | \text{spam})
\]

Appearances of words in email are conditionally independent given the email is spam or not.

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Spring 2021
Training: Naïve Bayes Email classification

Train set

\( n \) previous emails \((x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})\)

\( x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_m^{(i)}) \) for each word, whether it appears in email \( i \)

\( y^{(i)} = 1 \) if spam, 0 if not spam

Note: \( m \) is huge.

Which estimator should we use for \( \hat{P}(X_j|Y) \)?

A. MLE
B. Laplace estimate (MAP)
C. Other MAP estimate
D. Both A and B
Training: Naïve Bayes Email classification

Train set

\[ n \text{ previous emails } (x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)}) \]

\[ x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_m^{(i)}) \]

\[ y^{(i)} = 1 \text{ if spam, } 0 \text{ if not spam} \]

Note: \( m \) is huge.

Which estimator should we use for \( \hat{P}(X_j|Y) \)?

A. MLE
B. Laplace estimate (MAP)
C. Other MAP estimate
D. Both A and B

Many words are likely to not appear at all in the training set!
**Ex 3. Naïve Bayes Classifier \((m, n \text{ large})\)**

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)
\]

**Training**

\[
\forall i: \ \hat{P}(X_j = 1|Y = 0), \ \hat{P}(X_j = 0|Y = 0), \ \hat{P}(X_j = 1|Y = 1), \ \hat{P}(X_j = 0|Y = 0), \ \hat{P}(Y = 1), \ \hat{P}(Y = 0)
\]

Use MLE or Laplace (MAP)

**Testing**

\[
\hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)
\]

Laplace (MAP) estimates avoid estimating 0 probabilities for events that don’t occur in your training data.
Testing: Naïve Bayes Email classification

For a new email:
• Generate $X = (X_1, X_2, ..., X_m)$
• Classify as spam or not using Naïve Bayes assumption

Note: $m$ is huge.
Suppose train set size $n$ also huge (many labeled emails).
Can we still use the below prediction?

$$\hat{Y} = \arg \max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)$$
Testing: Naïve Bayes Email classification

For a new email:

- Generate $X = (X_1, X_2, ..., X_m)$
- Classify as spam or not using Naïve Bayes assumption

Note: $m$ is huge.

Suppose train set size $n$ also huge (many labeled emails).
Can we still use the below prediction?

$$\hat{Y} = \arg\max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)$$

Will probably lead to underflow!
Ex 3. Naïve Bayes Classifier \((m, n \text{ large})\)

\[
\hat{Y} = \arg\max_{y=\{0,1\}} \left( \prod_{j=1}^{m} \hat{P}(X_j|Y) \right) \hat{P}(Y)
\]

\(\forall i: \hat{P}(X_j = 1|Y = 0), \hat{P}(X_j = 0|Y = 0), \hat{P}(X_j = 1|Y = 1), \hat{P}(X_j = 0|Y = 1), \hat{P}(Y = 1), \hat{P}(Y = 0)\)

Use sums of log-probabilities for numerical stability.

\[
\hat{Y} = \arg\max_{y=\{0,1\}} \left( \log \hat{P}(Y) + \sum_{j=1}^{m} \log \hat{P}(X_j|Y) \right)
\]
How well does Naïve Bayes perform?

After training, you can test with another set of data, called the test set.

- Test set also has known values for $Y$ so we can see how often we were right/wrong in our predictions $\hat{Y}$.

Typical workflow:

- Have a dataset of 1789 emails (1578 spam, 211 ham)
- Train set: First 1538 emails (by time)
- Test set: Next 251 messages

Evaluation criteria on test set:

<table>
<thead>
<tr>
<th></th>
<th>Spam Prec.</th>
<th>Spam Recall</th>
<th>Non-spam Prec.</th>
<th>Non-spam Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words only</td>
<td>97.1%</td>
<td>94.3%</td>
<td>87.7%</td>
<td>93.4%</td>
</tr>
<tr>
<td>Words + addtl features</td>
<td>100%</td>
<td>98.3%</td>
<td>96.2%</td>
<td>100%</td>
</tr>
</tbody>
</table>
What are precision and recall?

Accuracy (# correct)/(# total) sometimes just doesn’t cut it.

**Precision**: Of the emails you predicted as spam, how many are *truly* spam? Measure of false positives

**Recall**: Of the emails that are truly spam, how many did you predict? Measure of false negatives