COMP47460

Naïve Bayes Classifier

Aonghus Lawlor
Deepak Anjwani

School of Computer Science
Autumn 2019

Overview

- Probability-based Learning
- Bayes Theorem
- Naïve Bayes Classifier
- Examples & Exercises

- Text Classification with Naïve Bayes
- Numeric Features
- Naïve Bayes in Weka

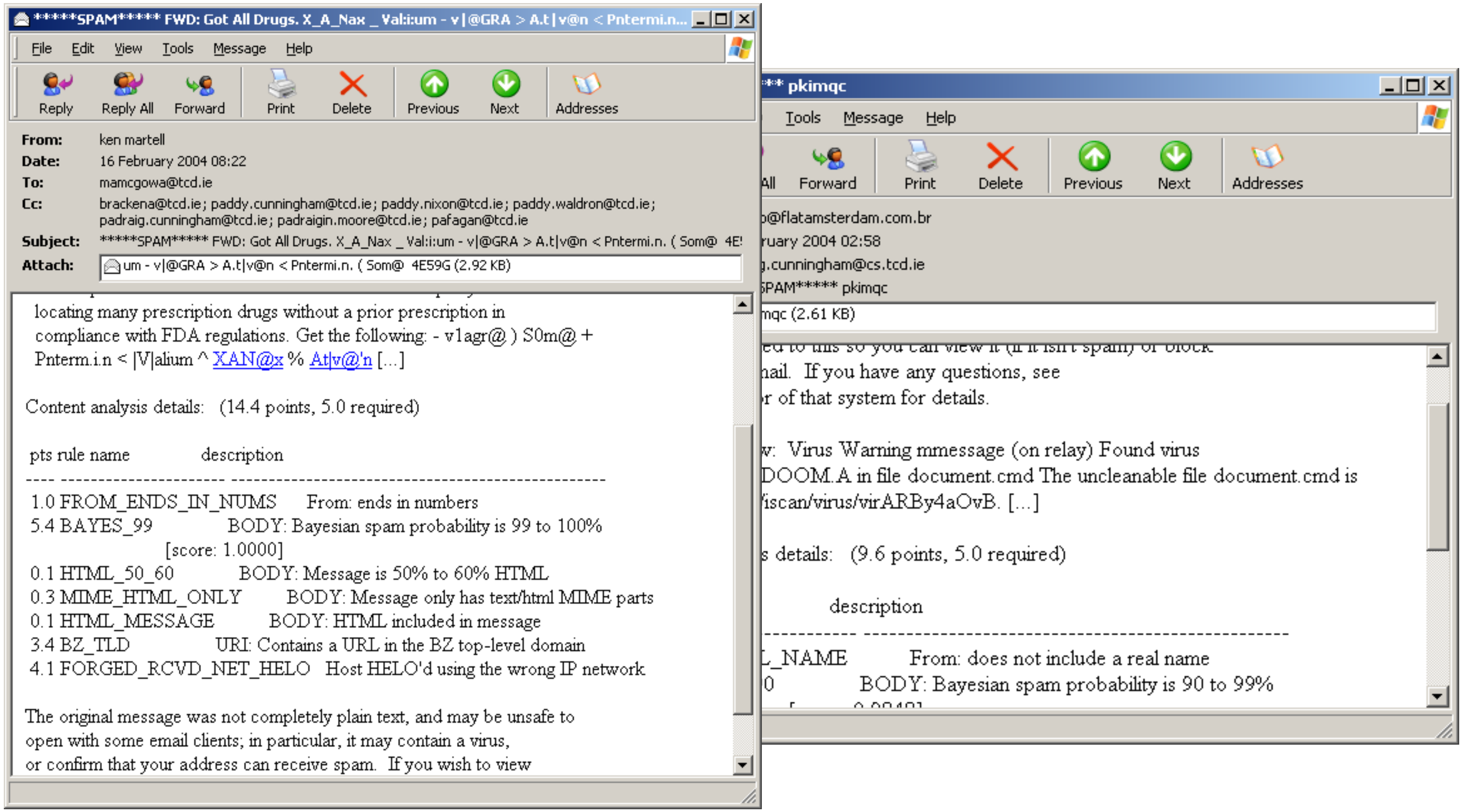
Probability-based Learning

- **Key Idea:** Use estimates of likelihoods to determine the most likely prediction which should be made (e.g. “the email X is more likely to be spam than non-spam”).
- Revise these predictions based on the data we collect.
- Most common probabilistic approach for classification is Naïve Bayes, an eager learning approach based on Bayes Theorem.

- **Why use a Naïve Bayes classifier?**
  - Intuitive and easy to implement.
  - Fast to train and to use as a classifier.
  - Suitable for moderate or large data sets with many features.
  - Can deal with missing features.

Application: Spam Filtering

Apache Spamassassin uses Naïve Bayes classification.

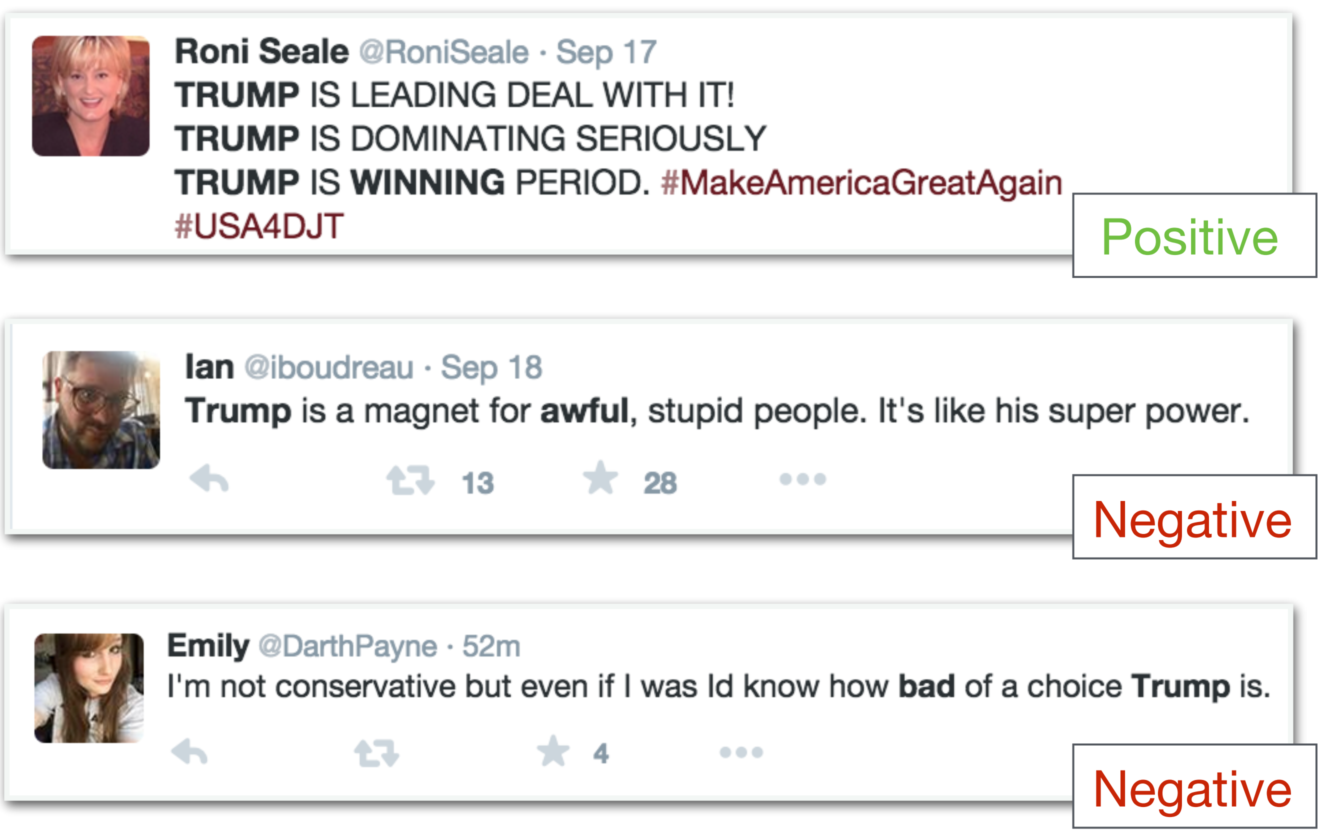


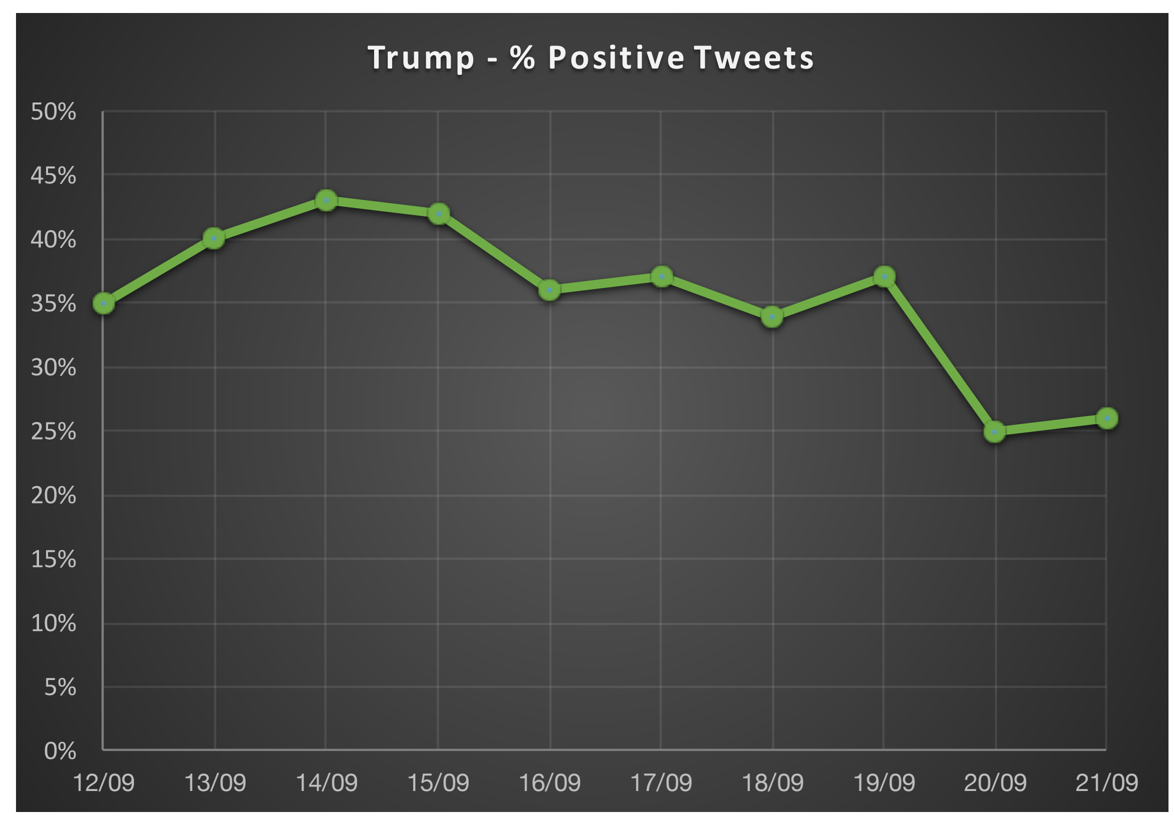
See: http://wiki.apache.org/spamassassin/BayesInSpamAssassin

Application: Sentiment Analysis

Task: Classify sentiment of tweets as “positive” or “negative”.

1. Crowdsource users to label a small subset of tweets as either “positive” or “negative” (i.e. training data).
2. Apply Naïve Bayes classifier to automatically label a much larger set of tweets on an ongoing basis.
3. Plot value of % of positive tweets over time.





Notation

$P(X)$	Probability of event *X* happening.
$P(X\|Y)$	Conditional probability of event *X* happening, given that event *Y* has happened.

What is the probability of a given hypothesis *h* being true ("the event"), given the observed training data *D* ("the evidence")?

Let h denote the hypothesis, D denote the data.

Prior probability of data

$P(D)$: Probability of the data D.

Prior probability of hypothesis - "initial beliefs"

$P(h)$: Probability of the hypothesis h.

Posterior probability

$P(h|D)$: Probability of the hypothesis h given the data D.

Bayes Classification

"The probability that an event has happened given a set of evidence for it is equal to the probability of the evidence being caused by the event by the probability of the event itself." (Kelleher et al, 2015)

- Bayes Theorem: Rule states that for each possible hypothesis *h*

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

$$\Pr(\text{spam}|\text{words}) = \frac{\Pr(\text{words}|\text{spam})\Pr(\text{spam})}{\Pr(\text{words})}$$

- For classification, each *h* corresponds to a possible class label.

  Q. What is the probability of a given example taking this class?

- If we knew $P(h|D)$ we could classify the data perfectly.
- Since we generally do not know $P(h|D)$, we try to estimate it from the data using Bayes Rule.

Bayes Classification

- We usually want the most likely hypothesis for our data.
- Formally, we are looking for the Maximum Aposteriori Hypothesis (MAP):

$$h_{MAP} = \arg\max_{h \in H} P(h|D)$$

- Example: Two competing hypotheses h_0 and h_1 for data set X

$$P(h_0|X) > P(h_1|X) \implies \text{choose } h_0$$

$$P(h_0|X) < P(h_1|X) \implies \text{choose } h_1$$

$$P(h_0|X) = P(h_1|X) \implies \text{choose either}$$

- In classification, we want to find the most likely class label for a given example among all possible class labels.

Example: Bayes Classification

- **Task:** Classify sentiment of tweets as “positive” or “negative”.

  $P(h_0)$ Probability of any tweet being classed “positive”.

  $P(h_1)$ Probability of any tweet being classed “negative”.

- Want to test hypothesis *h_0* - is a particular tweet *t* positive?

  $P(h_0|t)$ Probability of a positive class prediction for the tweet *t.* This is our target result.

  $P(t|h_0)$ Probability of the tweet *t*, given that it is positive. Calculated based on the data.

- We could rewrite the task with Bayes Theorem as follows:

$$P(h_0|t) = \frac{P(t|h_0)P(h_0)}{P(t)} = \frac{P(\text{tweet}|\text{positive})P(\text{positive})}{P(\text{tweet})}$$

Example: Bayes Classification

- Let's say that we know a-priori 60% of all tweets are positive and 40% of tweets are negative. $\Rightarrow P(positive) = 0.6$
- In addition, the probability of a tweet *t* is constant, so we can remove the denominator from the calculation:

$$P(positive|tweet) = \frac{P(\text{tweet}|\text{positive})P(\text{positive})}{P(\text{tweet})}$$

$$\Rightarrow \quad P(positive|tweet) = P(\text{tweet}|\text{positive}) \times 0.6$$

- But we still need some way of calculating the probability of a particular tweet (as described by its features), given the assumption that it has the class label "positive".

Definition: Bayes Classifier

Classifier Inputs:

A set of labels $V = \{v_1, v_2, \dots\}$

A set of examples $X = \{x_1, x_2, \dots\}$, each represented by features $\{f_1, f_2, \dots, f_n\}$

Classifier Objective:

Find the most probable class label v for x according to:

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j | f_1, f_2 \dots f_n)$$

$$\begin{aligned} v_{MAP} &= \arg\max_{v_j \in V} \frac{P(f_1, f_2 \dots f_n | v_j) P(v_j)}{P(f_1, f_2 \dots f_n)} \\ &= \arg\max_{v_j \in V} P(f_1, f_2 \dots f_n | v_j) P(v_j) \end{aligned}$$

Problem: Difficult to estimate $P(f_1, f_2 \dots f_n | v_j)$

Naïve Bayes Classifier

- Key Idea: Apply Bayes Theorem with the "naïve" assumption that all features in the data are *conditionally independent*:

$$P(f_1, f_2 \ldots f_n | v_j) = \prod_i P(f_i | v_j)$$

i.e. the value of a particular feature is unrelated to the presence or absence of any other feature, given class label v_j

- Based on this assumption, the objective of the Naïve Bayes classifier becomes:

Find the most probable class label v for x according to:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(f_i | v_j)$$

i.e. (Class Probability) x (Product of Class-Feature Probabilities)

Example: Swimming

Q. "Will we go swimming today?"
Binary classification task (Swimming = {Yes,No}), with examples described by 5 categorical weather features:

Example	Rain Recently (RR)	Rain Today (RT)	Temp (T)	Wind (W)	Sunshine (S)	Swimming
1	Moderate	Moderate	Warm	Light	Some	Yes
2	Light	Moderate	Warm	Moderate	None	No
3	Moderate	Moderate	Cold	Gale	None	No
4	Moderate	Moderate	Warm	Light	None	Yes
5	Moderate	Light	Cold	Light	Some	No
6	Heavy	Light	Cold	Moderate	Some	Yes
7	Light	Light	Cold	Moderate	Some	No
8	Moderate	Moderate	Cold	Gale	Some	No
9	Heavy	Heavy	Warm	Moderate	None	Yes
10	Light	Light	Cold	Light	Some	No

X0	Moderate	Moderate	Cold	Light	Some	???

➡ How can we use a Naïve Bayes Classifier to predict for *X0*?

Example: Swimming

- To use a Naïve Bayes Classifier, the first step is to construct a contingency table (probability table) of conditional and prior probabilities.
- That is, calculate the probability of each possible feature value given a class, and the overall probability of each class.
- e.g. If we look at the 4 training examples for Swimming=Yes:

Example	Rain Recently (RR)	Rain Today (RT)	Temp (T)	Wind (W)	Sunshine (S)	Swimming
1	Moderate	Moderate	Warm	Light	Some	Yes
2	Light	Moderate	Warm	Moderate	None	No
3	Moderate	Moderate	Cold	Gale	None	No
4	Moderate	Moderate	Warm	Light	None	Yes
5	Moderate	Light	Cold	Light	Some	No
6	Heavy	Light	Cold	Moderate	Some	Yes
7	Light	Light	Cold	Moderate	Some	No
8	Moderate	Moderate	Cold	Gale	Some	No
9	Heavy	Heavy	Warm	Moderate	None	Yes
10	Light	Light	Cold	Light	Some	No

Class Probability

P(Yes) = 4/10

Feature: Rain Recently

P(L_RR|Yes) = 0/4

P(M_RR|Yes) = 2/4

P(H_RR|Yes) = 2/4

Feature: Rain Today

P(L_RT|Yes) = 1/4

P(M_RT|Yes) = 2/4

P(H_RT|Yes) = 1/4

...

Example: Swimming

Construct full contingency table for all features on both classes:

Swimming	Yes	No
Rain Recently=light	0/4	3/6
Rain Recently=moderate	2/4	3/6
Rain Recently=heavy	2/4	0/6
Rain Today=light	1/4	3/6
Rain Today=moderate	2/4	3/6
Rain Today=heavy	1/4	0/6
Temp=Cold	1/4	5/6
Temp=Warm	3/4	1/6
Wind=Light	2/4	2/6
Wind=Moderate	2/4	2/6
Wind=Gale	0/4	2/6
Sunshine=Some	2/4	4/6
Sunshine=None	2/4	2/6
Class Probabilities (Priors)	4/10	6/10

Example: Swimming

Test new input example for hypothesis 1: *Swimming=Yes*

Example	Rain Recently (RR)	Rain Today (RT)	Temp (T)	Wind (W)	Sunshine (S)	Swimming
X0	Moderate	Moderate	Cold	Light	Some	???

(Product of Class-Feature Probabilities) x (Class Probability)

```
P(Yes) = (2/4 x 2/4 x 1/4 x 2/4 x 2/4) x 4/10
P(Yes) = 0.00625
```

Test new input example for hypothesis 2: *Swimming=No*

```
P(No) = (3/6 x 3/6 x 5/6 x 2/6 x 4/6) x 6/10
P(No) = 0.028
```

We usually normalise probabilities to sum to 1:

$$P(Yes)' = \frac{0.00625}{0.00625+0.028} = 0.18$$

$$\boxed{P(No)' = \frac{0.028}{0.00625+0.028} = 0.82}$$

Output: *Swimming=No*

Handling Numeric Features

- How to classify when features take numeric values?

Example	Rain Recently (RR)	Rain Today (RT)	Temp (T)	Wind (W)	Sunshine (S)	Swimming
X0	Moderate	Moderate	9	Light	Some	???

- **Option 1:** Discretise the feature to take fixed number of values. e.g. Temp = {cool, mild, hot}

- **Option 2:** Assume that the feature fits to some distribution. e.g. for a Normal Distribution:
  1. For numeric feature f_i, store mean μ_i and standard deviation σ_i for each class v_j
  2. When classifying, find the probability that the feature value fits the distribution $N(\mu_i, \sigma_i^2)$

Text Classification

- **Naïve Bayes for text:** Each word in the vocabulary of a collection of documents is a feature; assume independence between word occurrences.
- Input: Examples *X* (set of documents), *V* (class labels)

LEARN_NB_TEXT(X, V):

- $Vocabulary \leftarrow$ set of all unique words in X
- FOR EACH $v_j \in V$
  - $Docs_j \leftarrow$ subset of documents from X with class label v_j
  - $P(v_j) \leftarrow \frac{|Docs_j|}{|X|}$
  - $Text_j \leftarrow$ concatenation of all text from $Docs_j$
  - $n \leftarrow$ total number of word positions in $Text_j$
  - FOR EACH word $w_k \in Vocabulary$
    - $n_k \leftarrow$ number of occurrences of word w_k in $Text_j$
    - $P(w_k|v_j) \leftarrow \frac{n_k}{n}$

Text Classification

- Once we have computed word probabilities for each class, we can use these to predict the class of a new input document *Doc.*
- Words not present in *Vocabulary* are not considered.

CLASSIFY_NB_TEXT(Doc):

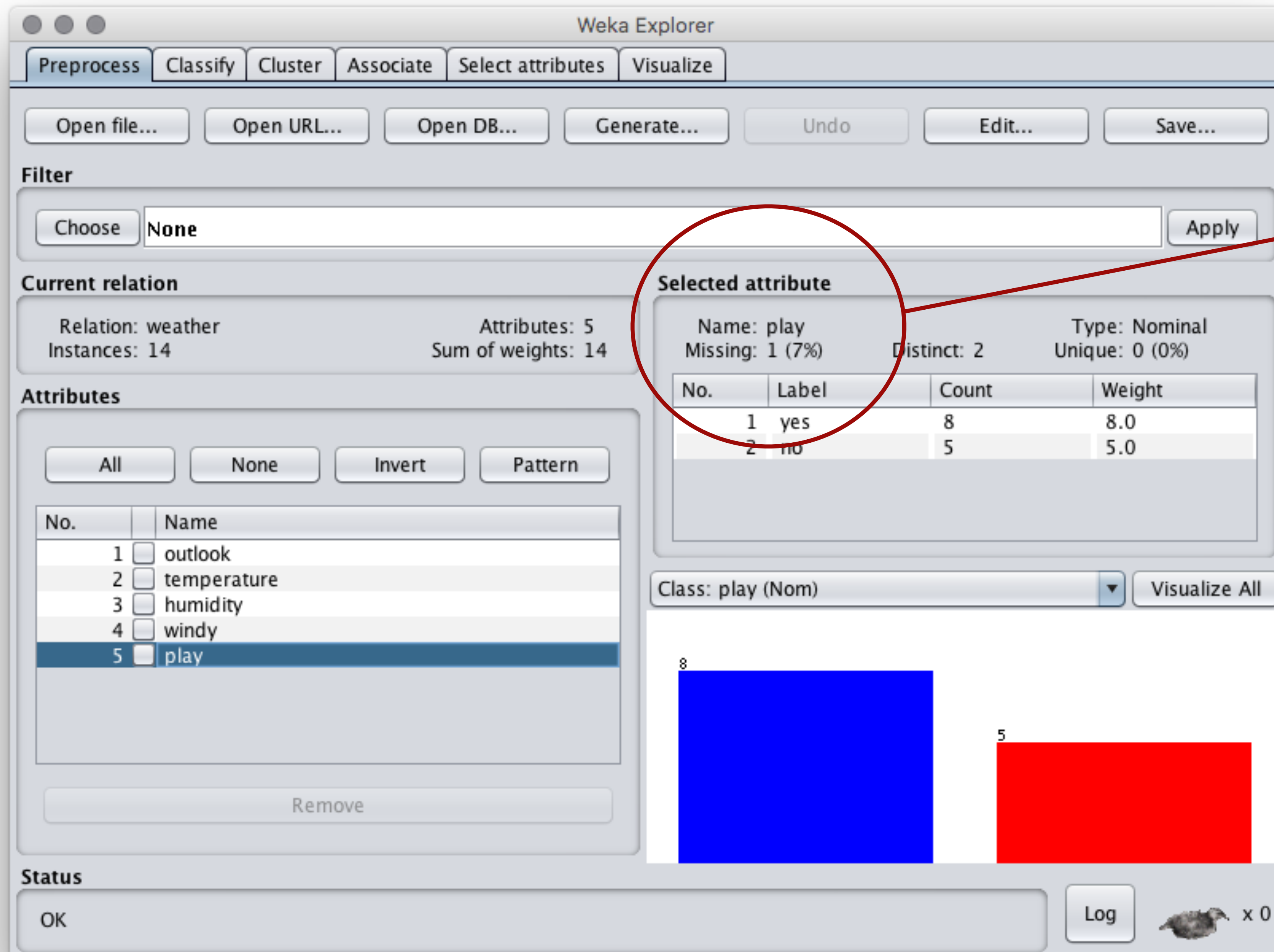
- $Positions \leftarrow$ all word positions in Doc with words from $Vocabulary$
- Return class label v_{NB} such that:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_{i \in Positions} P(w_i | v_j)$$

- But… words are not independent of one another (e.g. United + States, Barack + Obama, Enda + Kenny).
- Often the conditional independence assumption is violated. Despite this, in practice Naïve Bayes classifiers perform well.

Naïve Bayes in Weka

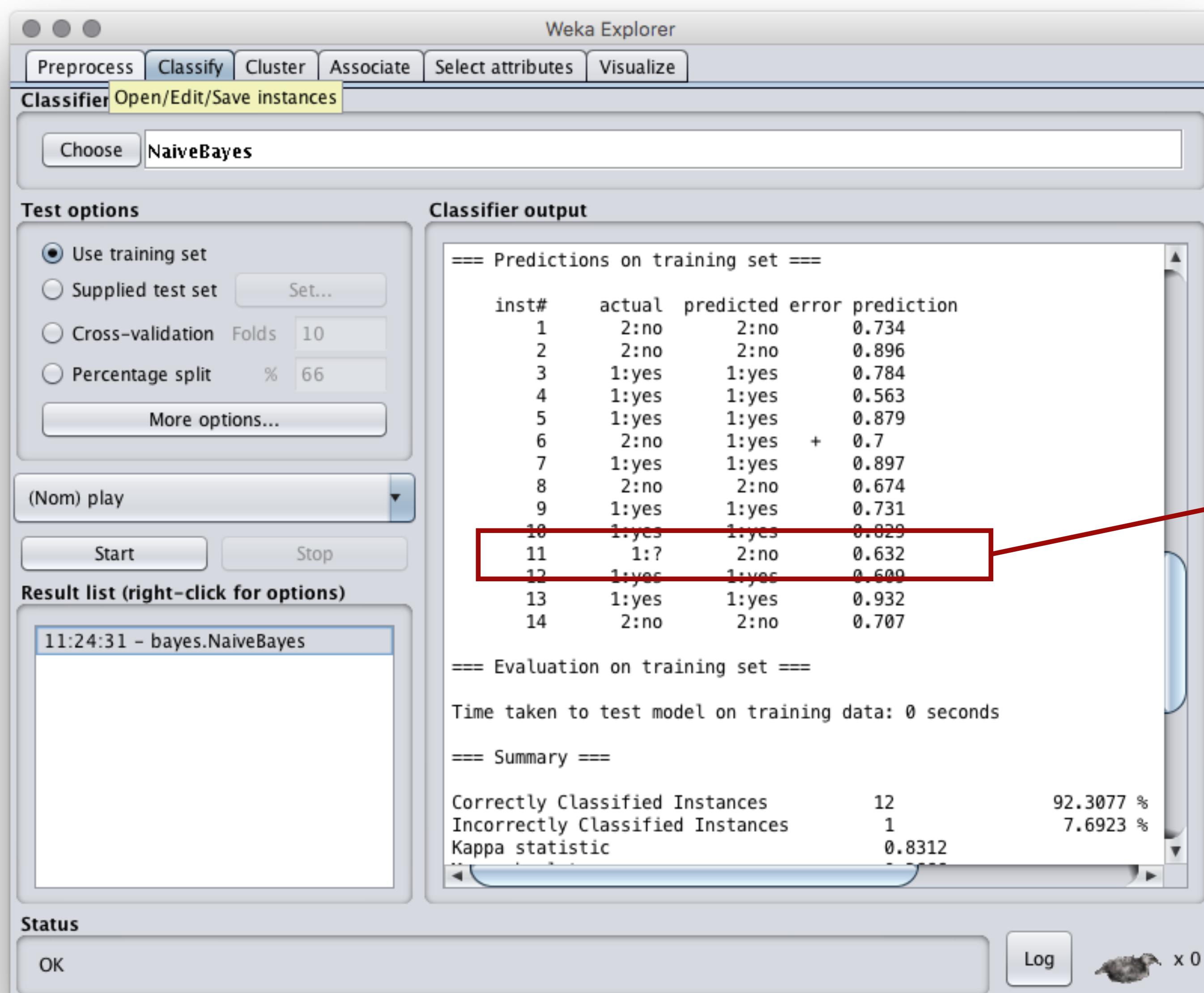
1. Launch the WEKA application, click on the *Explorer* button.
2. *Open File* - weather-prediction.arff



Note we are missing a class label (Play) for one of the examples i.e. it is unlabelled

Naïve Bayes in Weka

3. In *Classify* tab, click *Choose* and choose *Bayes→NaiveBayes*
4. Set *Test Options* to *Use Training Set*
5. Click *More Options* button, set *Output Predictions* to *PlainText*.
6. Choose *(Nom) Play* as class label, then click *Start.*



Predicted class for unlabelled Example 11 is "No"

Summary

- Naïve Bayes Classifier
  - Probabilistic approach to classification.
  - Based on key independence assumption. This assumption is often violated, but still works.
- Handling Numeric Features
  - Make feature discrete or assume a distribution.
- Text Classification with Naïve Bayes
  - Learning: Calculate word probabilities for vocabulary
  - Classifying: Find product of word probabilities in the new document.

References

- J. D. Kelleher, B. Mac Namee, A. D'Arcy. "Fundamentals of Machine Learning for Predictive Data Analytics", 2015.
- T. Mitchell. "Machine Learning". McGraw-Hill, 1997. pp. 55–58.
- Witten, I., Frank, E., Hall, M. "Data Mining: Practical Machine Learning Tools and Techniques, 3rd Ed".
- Lewis, D. D. "Naive (Bayes) at forty: The independence assumption in information retrieval". Proceedings of ECML 1998.