The Problem

Given the occurence of certain words in an email, for example "viagra", "million dollars" and "get rich", what is the probability that the email is spam?

We can tackle this problem using the Naive Bayes Algorithm

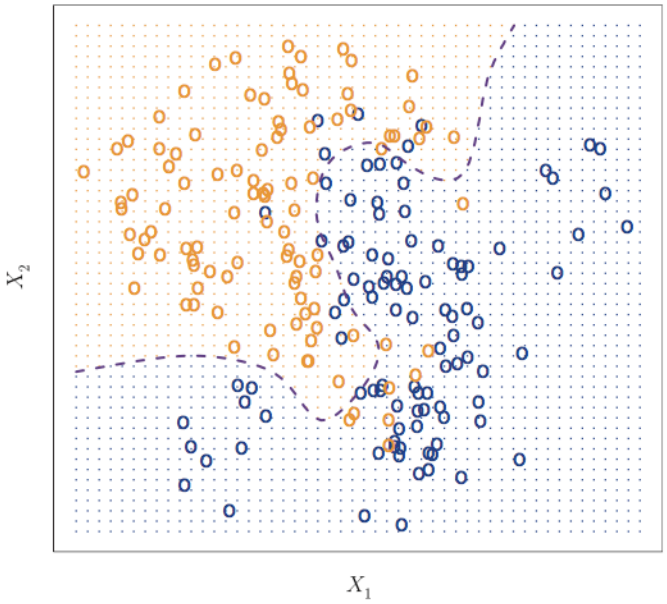
Thanks to RStudio Labs for the example. source: https://rstudio-pubs-static.s3.amazonaws.com/144238_29afd51da1bb46e1be952a190c772d27.html (https://rstudio-pubs-static.s3.amazonaws.com/144238_29afd51da1bb46e1be952a190c772d27.html)

The Naive Bayes Algorithm

- Based on conditional probability - given predictor values x, what is the probability of y?
- In our email example, given certain predictor words, what is the probability of the email being spam/not spam?
- P(Y = spam | X = x)

A visual illustration

- spam = orange and not spam = blue. The task of the Naive Bayes Classifier is to find a decision boundary separating the two classes



The Data

Toy datasets are great for building an intuition of how an algorithm works. For this example, we will create our own toy dataset to play with

In [3]:
```
train <- data.frame(class=c("spam","not spam","not spam","not spam"),
                    viagra=c("yes","no","no","yes"))
train
```

class	viagra
spam	yes
not spam	no
not spam	no
not spam	yes

Training Data

In [14]:
```
library(e1071)
NB <- naiveBayes(class ~ viagra, data = train)
NB
```

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
not spam     spam
    0.75     0.25

Conditional probabilities:
          viagra
Y                 no       yes
  not spam 0.6666667 0.3333333
  spam     0.0000000 1.0000000
```

Testing Data

In [16]:
```
test <- data.frame(viagra=c("yes"))
test$viagra <- factor(test$viagra, levels=c("no","yes"))
test
```

viagra
yes

Applying our predictive model to the test data

In [18]:
```
prediction <- predict(NB, test ,type="raw")
prediction
```

not spam	spam
0.5	0.5

Question: Why is Naive Bayes naive?

Let's investigate this question by considering two variables, "meet" and "viagra", instead of just "viagra".

In [19]:
```
train <- data.frame(type=c("spam","ham","ham","ham"),
                    viagra=c("yes","no","no","yes"),
                    meet=c("yes","yes","yes", "no"))
train
```

type	viagra	meet
spam	yes	yes
ham	no	yes
ham	no	yes
ham	yes	no

In [21]:
```
# train a classifier
NB2 <- naiveBayes(type ~ viagra + meet,train)
NB2
```

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
 ham spam
0.75 0.25

Conditional probabilities:
      viagra
Y             no       yes
  ham  0.6666667 0.3333333
  spam 0.0000000 1.0000000

      meet
Y             no       yes
  ham  0.3333333 0.6666667
  spam 0.0000000 1.0000000
```

In [23]:
```
# Create test data
test <- data.frame(viagra=c("yes"), meet=c("yes"))
test$viagra <- factor(test$viagra, levels=c("no","yes"))
test$meet <- factor(test$meet, levels=c("no","yes"))
test
```

viagra	meet
yes	yes

In [24]:
```
# Run our model on test data
prediction <- predict(classifier, test ,type="raw")
prediction
```

ham	spam
0.4	0.6

We see that the conditional probability of an email being spam, given the *meet* and the *viagra* variable is 0.6. However, if we were to compute the conditional probabilities by hand, we would find that the true conditional probability is 1. Our classifier is underestimating the true probability.

This is because Naive Bayes assumes that the "probability that a message contains 'viagra' given that it is spam is independent of whether or not the message contains 'meet'"

source:https://rstudio-pubs-static.s3.amazonaws.com/144238_29afd51da1bb46e1be952a190c772d27.html (https://rstudio-pubs-static.s3.amazonaws.com/144238_29afd51da1bb46e1be952a190c772d27.html)

Caveat: Naive Bayes assumes variables are independent

- This is worth considering before implementing a Naive Bayes classifier

Create your own toy dataset and classifier

In [25]:
```
## Your code goes here ##
```