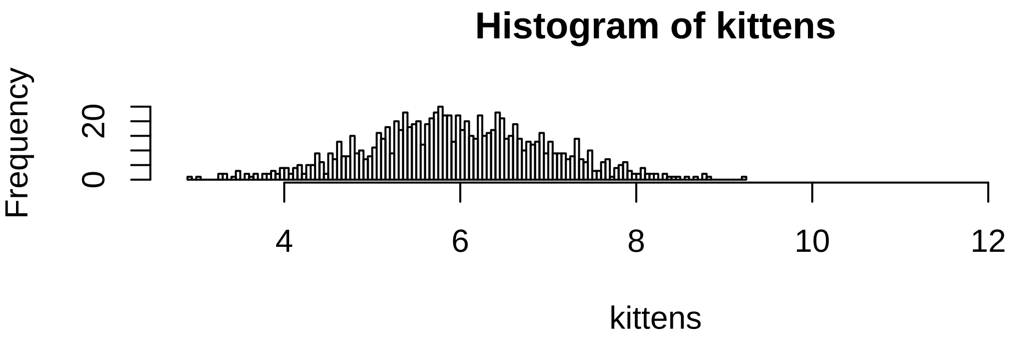
Simple Example of Supervised Learning

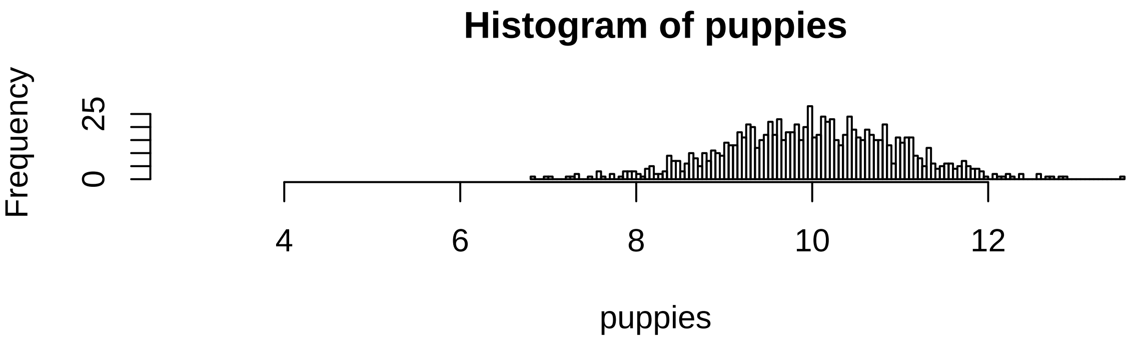
Jon Kinsey

Sun Nov 30 14:38:08 2014

```r
#
# This is a simple example of telling kittens from puppies using
# some fake weight data for 1000 puppies and kittens.  This is
# supervised learning because we know which is which.
#
# Machine learning is awesome - it's great for:
# 1.  Telling two things apart
# 2.  Finding hidden patterns
# 3.  Predicting future values
# 4.  Finding anomalies.

# There are two types of learning:
#    Regression: the response Y is quantitative.
#    Classification: the response is qualitative, or categorical.

# Machine learning falls into two main camps:
# - Supervised:  We have labels for
#    what we want to learn.
# - Unsupervised:  We don't have labels
#    but we kind of want to learn them automatically.

# The path to machine learning:
# 1.  Turn the world into numbers ("features")
# 2.  Model those numbers.
# 3.  See how they did.

# NOTE: Beware of "overfitting".
# It's a problem where you fit your model too close to your training data
# and it  doesn't generalize to the world well.
# In machine learning, a concept called "cross-validation" is used to test
# the model, in which the training data is chopped up into two sets - training
# and testing data that we use to see how well the data will generalize.

kittens <- rnorm(1000, 6, 1)
puppies <- rnorm(1000, 10, 1)

weight.data <- data.frame(weight=c(kittens, puppies), type=c(rep(1, 1000), rep(0,1000)))

# This data is "training data" because we're going to use it to "train" (build)
# a model that we'll then use on future data (called "testing data").

# So given this data, how would we model our two classes?
# Simplest model:  a line dividing the two classes.  Animals with weight below
# our line will be kitties, animals with weight above our line will be puppies.

# Where should we draw this line though?

par(mfrow=c(2,1))
```

```
hist(kittens, breaks=100, xlim=c(min(kittens),max(puppies)))
hist(puppies, breaks=100, xlim=c(min(kittens),max(puppies)))
```





```
# Let's start at 7.  What if we use 7, how well does this work ?
predictions <- as.numeric(weight.data$weight < 7)
correct <- sum(predictions == weight.data$type)
correct / nrow(weight.data)
```

```
## [1] 0.9215
```

```
# Hey hey, not bad!  Can we do better?
# Let's try a bunch of values and see the minimum error:

MSE <- vector()
values <- seq(min(puppies), max(kittens), 0.1)

for (i in values) {
  predictions <- weight.data$weight < i
  error <- mean((as.numeric(predictions)-weight.data$type)^2) #MSE
  MSE <- c(MSE, error)
}

min(MSE)
```

```
## [1] 0.022
```

```r
which(MSE == min(MSE)) # When I first ran this, the 12th value is smallest
```

```
## [1] 14
```

```r
values[12]  # My value is 8.045236.  Cool.
```

```
## [1] 7.926
```

```r
threshold <- values[12]

# So this is my first classifier. Lets try it out:

test.kitties <- rnorm(100, 6, 1)
test.puppies <- rnorm(100, 10, 1)

test.data <- data.frame(weight=c(test.kitties, test.puppies), type=c(rep(1,100), rep(0,100)))

# How'd we do?
predictions <- test.data$weight < threshold
sum(abs(predictions == test.data$type)/nrow(test.data))
```

```
## [1] 0.96
```