Practical Machine Learning in R

Introduction

Lars Kotthoff[1][2]
larsko@uwyo.edu

[1] with slides from Bernd Bischl and Michel Lang

[2] slides available at http://www.cs.uwyo.edu/~larsko/ml-fac

What is Machine Learning?

- “gives computes the ability to learn without being explicitly programmed” (Wikipedia)

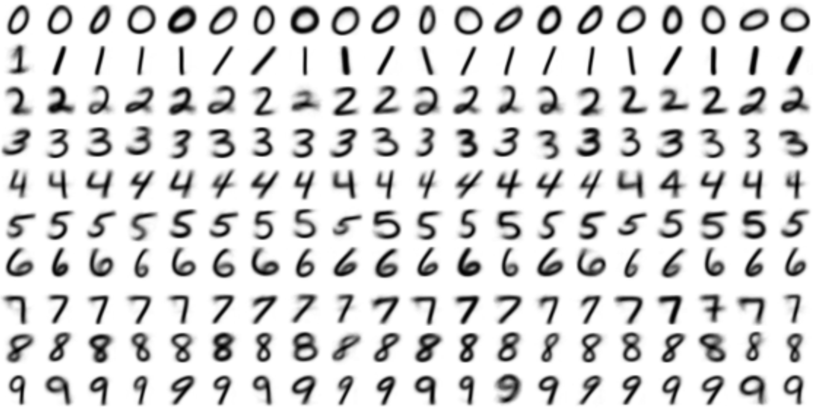
What is Machine Learning?

- “gives computes the ability to learn without being explicitly programmed” (Wikipedia)
- “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.” (Tom Mitchell)

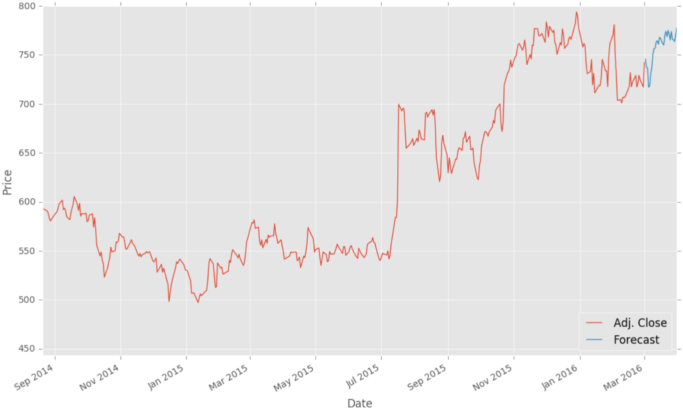
Examples

SPAM
PRODUIT DES E.-U.A.
VIANDE A LUNCH ENTIEREMENT CUITE
PRODUCT OF U.S.A.
LUNCHEON MEAT FULLY COOKED
FULLY COOKED LUNCHEON MEAT
50% LESS SODIUM
THAN OUR REGULAR LUNCHEON MEAT
Présentation Suggérée
Serving Suggestion
Hormel Foods
340 g

Examples

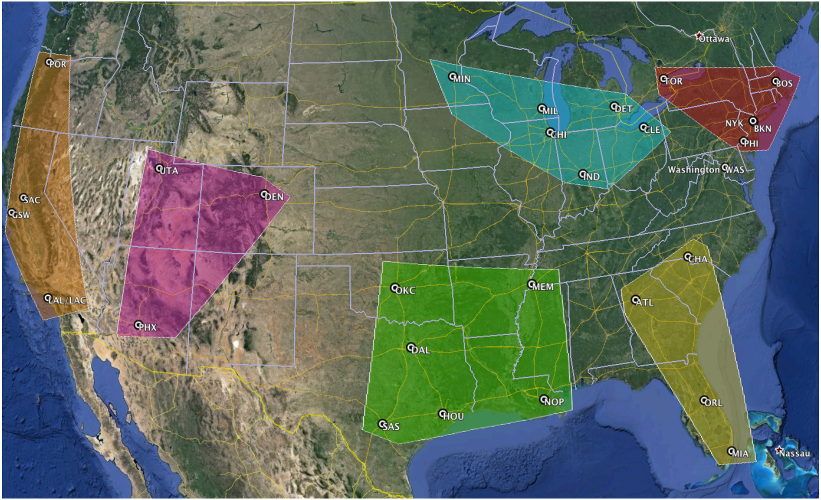


Examples



https://pythonprogramming.net/forecasting-predicting-machine-learning-tutorial/

Examples



https://squared2020.com/2015/09/09/redefining-nba-divisions-by-clustering/

Supervised Learning

- learn the relationship between input x and output y
- training data with labels available – y known for given x
- can see this as function approximation – find an f such that

$$y \approx f(x)$$

Supervised Learning

- x are features or attributes
- y is the ground truth
- denote predictions $f(x) = \hat{y}$
- loss function $L(y, \hat{y})$ measures how good predictions are, e.g.

$$L(y, \hat{y}) = (y - \hat{y})^2$$

- want to minimize loss given training data $X_{\text{train}} = \{(x_i, y_i)\}^n$:

$$\arg\min \sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

Supervised Learning

- want to learn a general function that is predictive on new data
- second set X_{test} that is not used in training to test generalization performance:

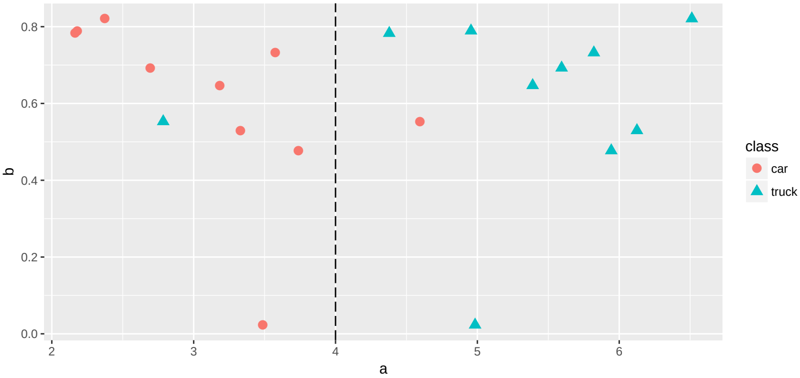
$$\sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

- usually full data set X is split into non-overlapping train and test sets:

$$X_{train} \cup X_{test} = X$$

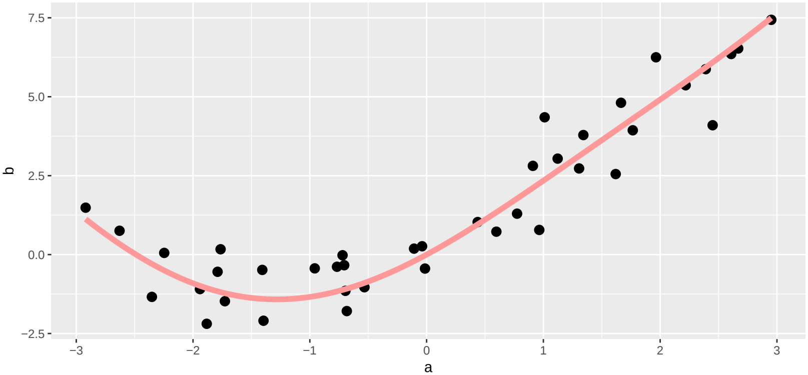
$$X_{train} \cap X_{test} = \varnothing$$

Supervised Classification



Goal: Predict a class (discrete quantity), or membership probabilities

Supervised Regression

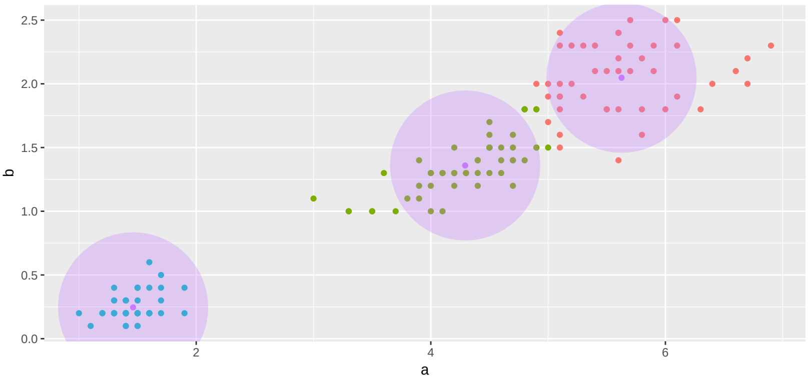


Goal: Predict a continuous quantity

Unsupervised Learning

- no ground truth y available
- determine group membership or assign labels
- loss function measures properties of groups, e.g. homogeneity wrt. features
- still want to minimize loss given training data and generalize

Unsupervised Clustering



Goal: Group data by similarity, or estimate membership probabilities

In this Course

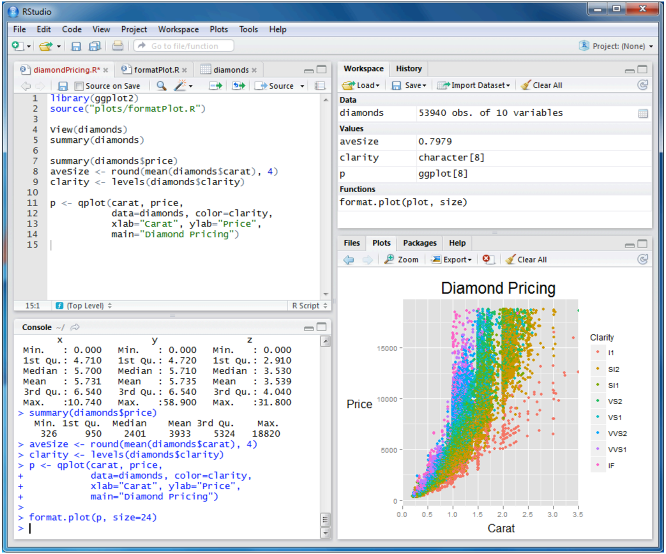
- classification
- regression
- clustering
- data preprocessing (missing values, dimensionality reduction)
- performance evaluation
- parameter tuning

Not in this Course

- R tutorial
- details on particular methods
- deep learning
- time series
- Big Data

What you'll need

Install RStudio



```r
library(ggplot2)
source("plots/formatPlot.R")

View(diamonds)
summary(diamonds)

summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)

p <- qplot(carat, price,
           data=diamonds, color=clarity,
           xlab="Carat", ylab="Price",
           main="Diamond Pricing")
```

Data

diamonds	53940 obs. of 10 variables

Values

aveSize	0.7979
clarity	character[8]
p	ggplot[8]

Functions

format.plot(plot, size)

```
        x               y                z
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
 Median : 5.700   Median : 5.710   Median : 3.530
 Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
 3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
 Max.   :10.740   Max.   :58.900   Max.   :31.800
> summary(diamonds$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    326     950    2401    3933    5324   18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+            data=diamonds, color=clarity,
+            xlab="Carat", ylab="Price",
+            main="Diamond Pricing")
>
> format.plot(p, size=24)
> |
```

https://www.rstudio.com/products/rstudio/download/

Install mlr

- on the R console:

```
install.packages("mlr")
```

- or see http://derekogle.com/IFAR/supplements/installations/InstallPackagesRStudio.html
- extensive tutorial available: https://mlr-org.github.io/mlr-tutorial/devel/html/

Format

- meetings roughly every week
- half lecture, half practical exercises
- happy to discuss specific problems