LABSIN

An Introduction to Machine Learning and Deep Learning with R

HARPO (AKA Carlos A. Catania Ph.D)
LABSIN - Ingeniería - UNCuyo

@harpolabs
harpo@ingenieria.uncuyo.edu.ar

AGENDA Day 1: Machine Learning

- Basic concepts
- The Machine Learning Workflow
- Supervised vs Unsupervised
- The Caret Package
- LAB 1:
  - Wine Quality prediction
- LAB 2
  - Tree Inclination prediction

Materials Day 1: Machine Learning

- **Rstudio** desktop or server version 1.2
- **R** version >= 3.5
- **Rstudio** Notebook operational
- `install.packages(“caret”)`
- `install.packages(“tidyverse”)`

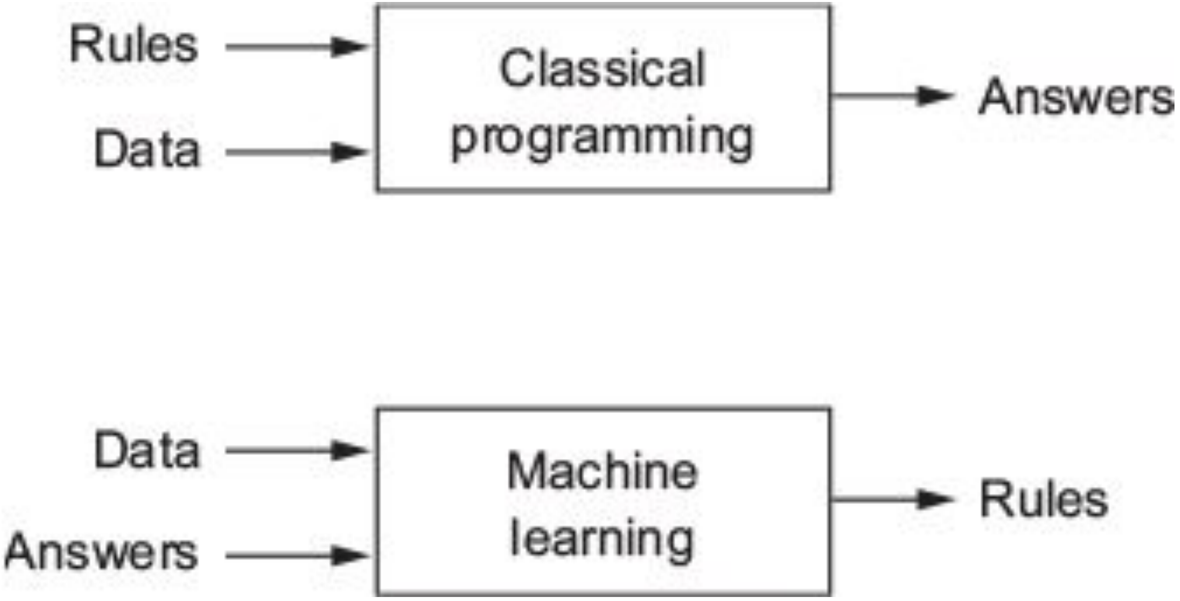
Machine Learning

Machine Learning: A definition

"The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience."

Tom Mitchell Circa 1997

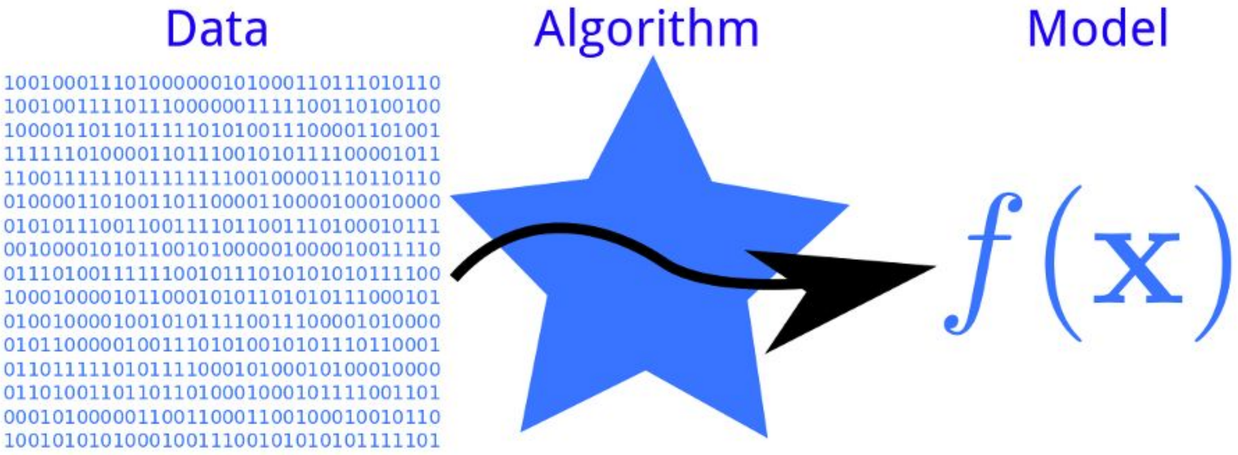
A Computer Science Perspective on Machine Learning



More Formally...

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.*

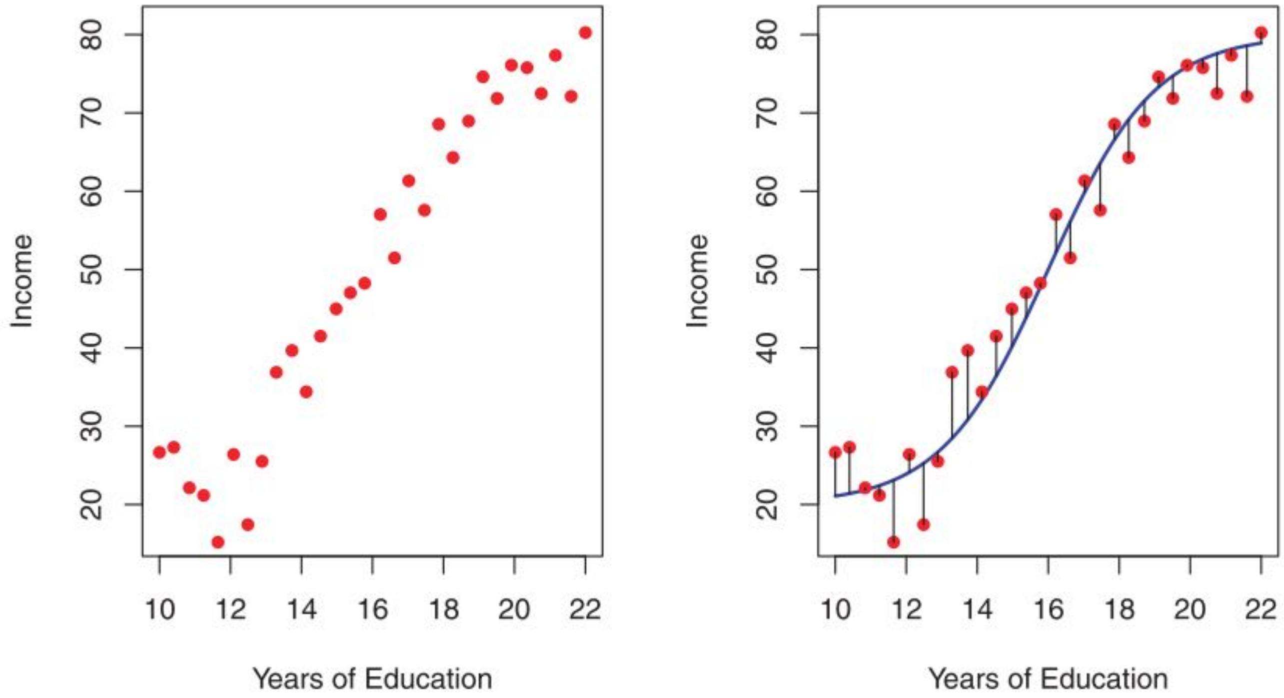
The general idea behind ML



More Formally... (again)

More generally, suppose that we observe a quantitative response Y and p different predictors, $X_1, X_2, \ldots, X_p$. We assume that there is some relationship between Y and $X = (X_1, X_2, \ldots, X_p)$, which can be written in the very general form

$$Y = f(X) + \epsilon.$$



A plot of income versus years of education for 30 individuals in the Income data set.

Why Estimate f?

Two main reasons:

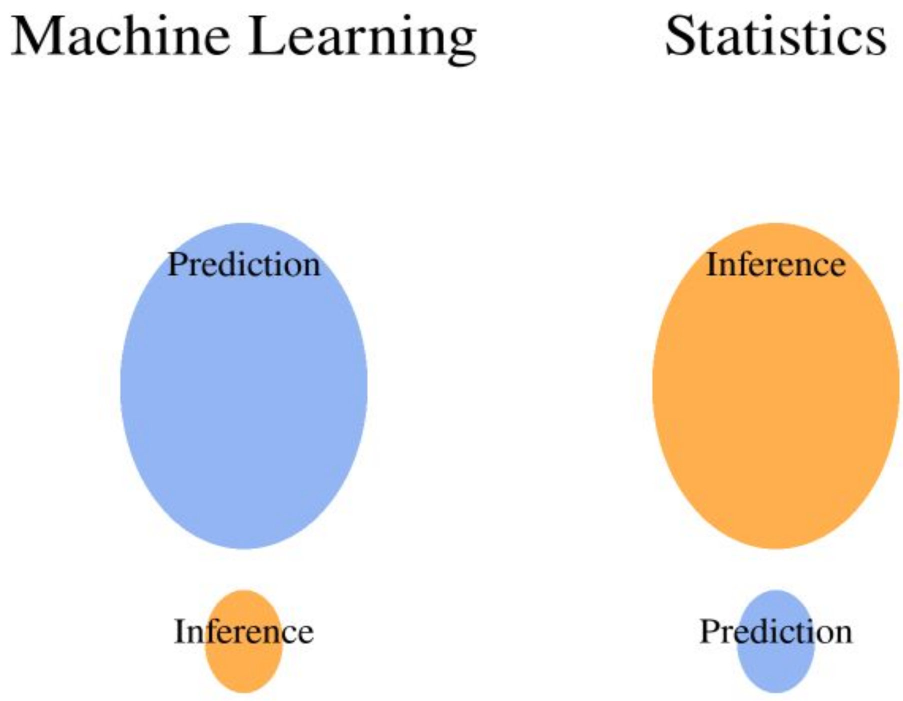
Prediction: Use the model to predict the outcomes for new data points. One **is not typically concerned with the exact form of f**, provided that it yields accurate predictions for Y.

$$\hat{Y} = \hat{f}(X),$$

Inference: We are often interested in understanding the way that Y is affected as $X_1,\ldots,X_p$ change.

In Machine Learning we mostly care about Prediction!

Differences between ML and Statistics



Differences between ML and Statistics

ML algorithms are often treated as black boxes.

Differences between ML and Statistics

- ML tends to deal with **large, complex datasets** (such as a dataset of millions of images, each consisting of tens of thousands of pixels)
- **Little mathematical theory**—maybe too little—
- ML **is engineering oriented**. ideas are proven empirically more often than theoretically.

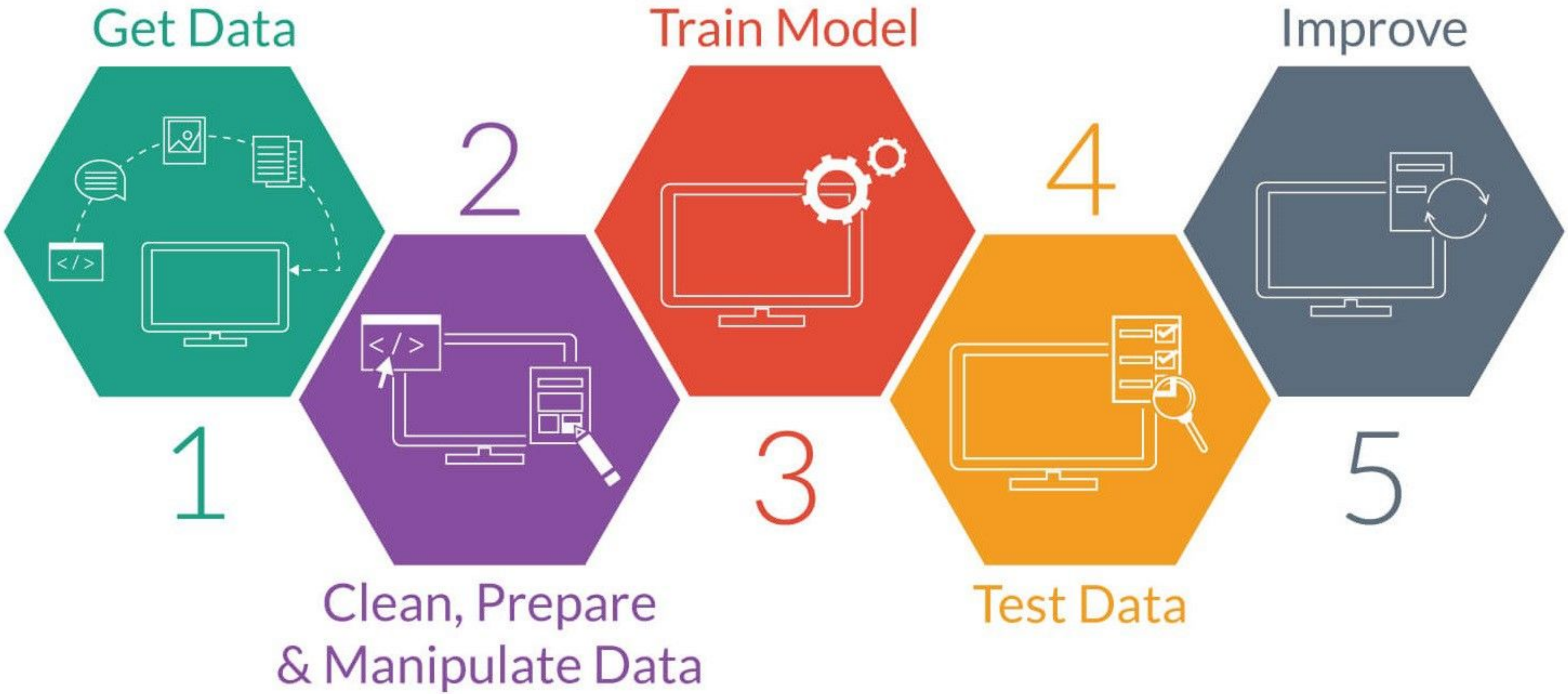
Machine Learning Algorithms

The algorithms and method come from areas such as:

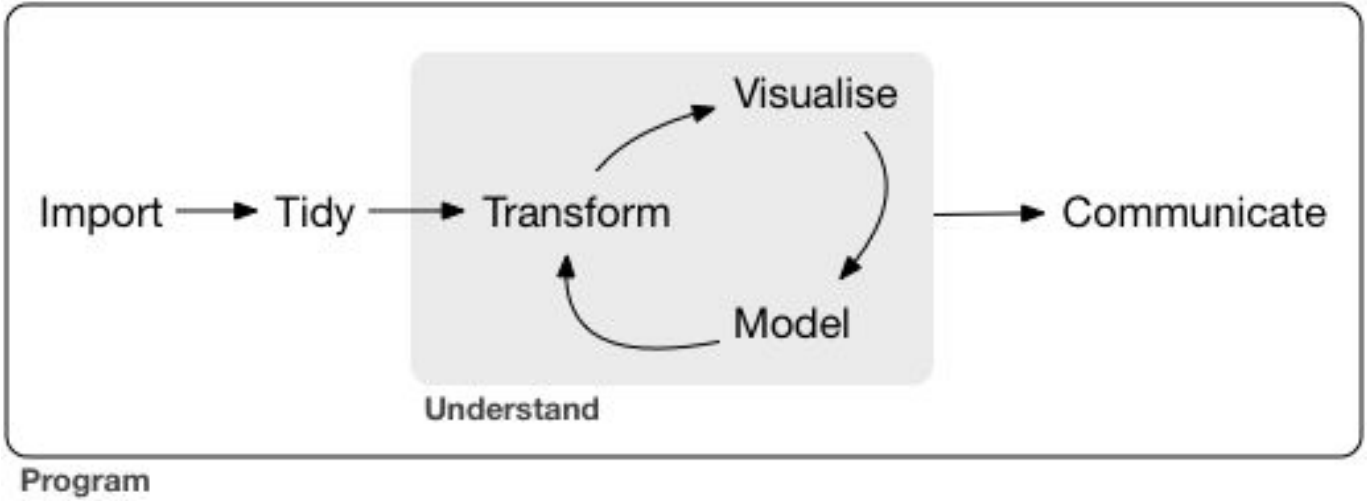
- Pattern Recognition
- Applied Statistics
- Artificial Intelligence

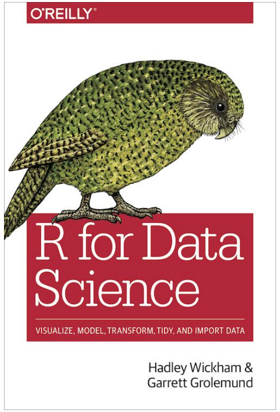
The borderline between disciplines has become diffuse.

The machine learning workflow



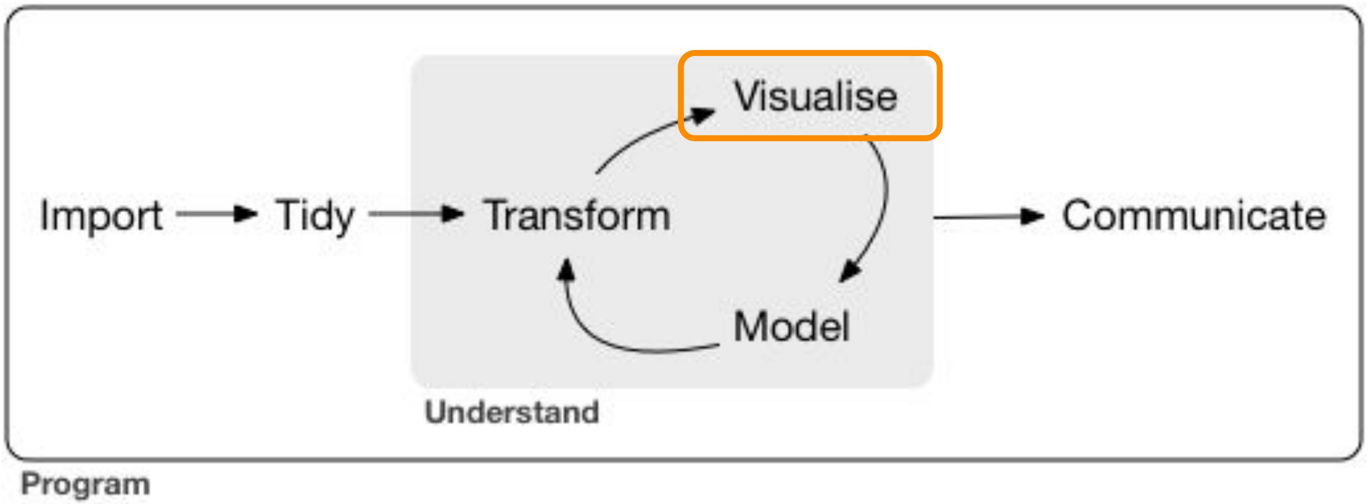
In R we have our own way to do it. (Actually, Hadley's way)



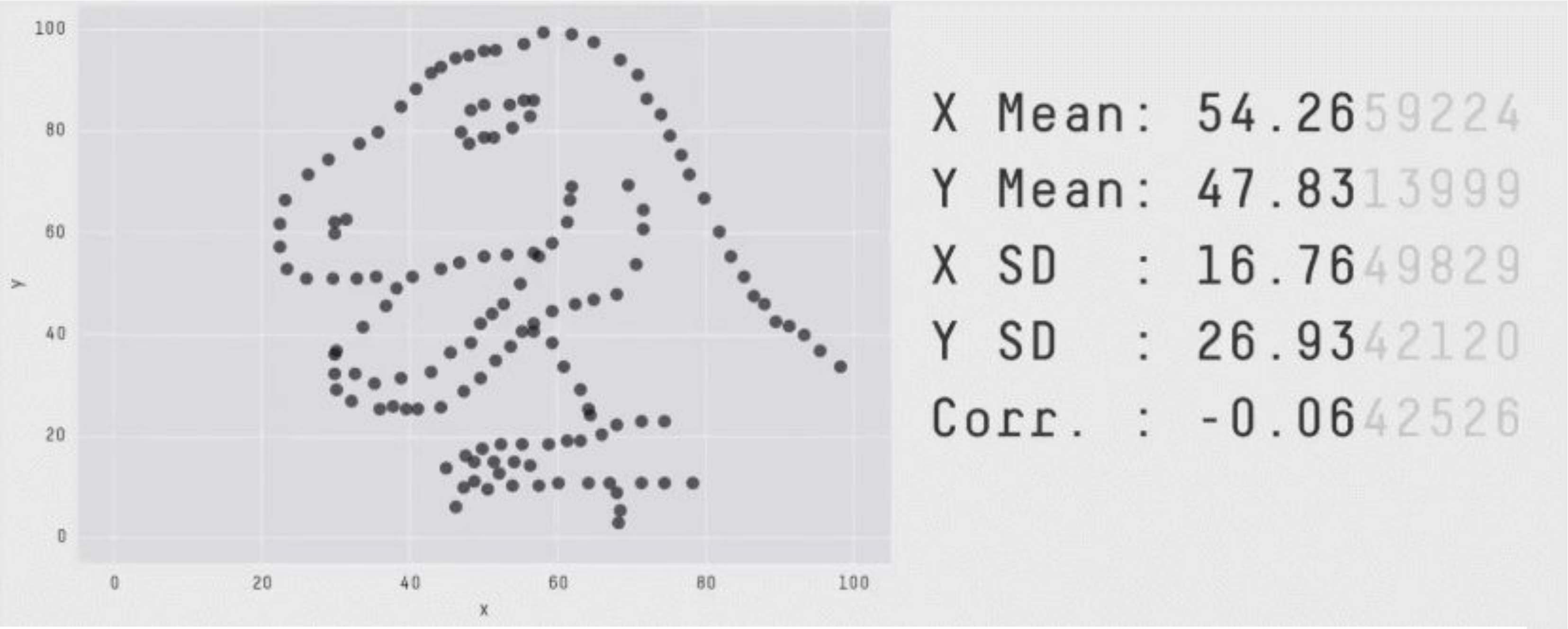


https://r4ds.had.co.nz

In R we have our own way to do it. (Actually, Hadley's way)



Mom said, you should always visualize your dataset



Justin Matejka, George Fitzmaurice (2017) Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing

Supervised vs. Unsupervised Learning

SUPERVISED:
For each observation of the predictor measurement(s) x_i, $i = 1, \ldots, n$ there is an associated response measurement y_i. We wish to fit a model that relates the response to the predictors.

UNSUPERVISED:
For every observation $i = 1, \ldots, n$, we observe a vector of measurements x_i but no associated response y_i. We seek to understand the relationships between the variables or between the observations.

SUPERVISED VS UNSUPERVISED

SUPERVISED LEARNING

All data has been labeled (supervised) by an expert. Thanks to this labeling process, we can help the network to realise the difference between classes (even though sometimes this does not happen).

Some techniques: NNs, SVM, etc.

UNSUPERVISED LEARNING

Our data are not labeled. Unsupervised algorithms consider confidence measures among samples in order to create homogeneous clusters.

Most famous technique: Clustering (k-means, hierarchical etc.)

For doing ML we need:

- *Input data points*
- *Examples of the expected output*
- *A way to measure whether the algorithm is doing a good job*

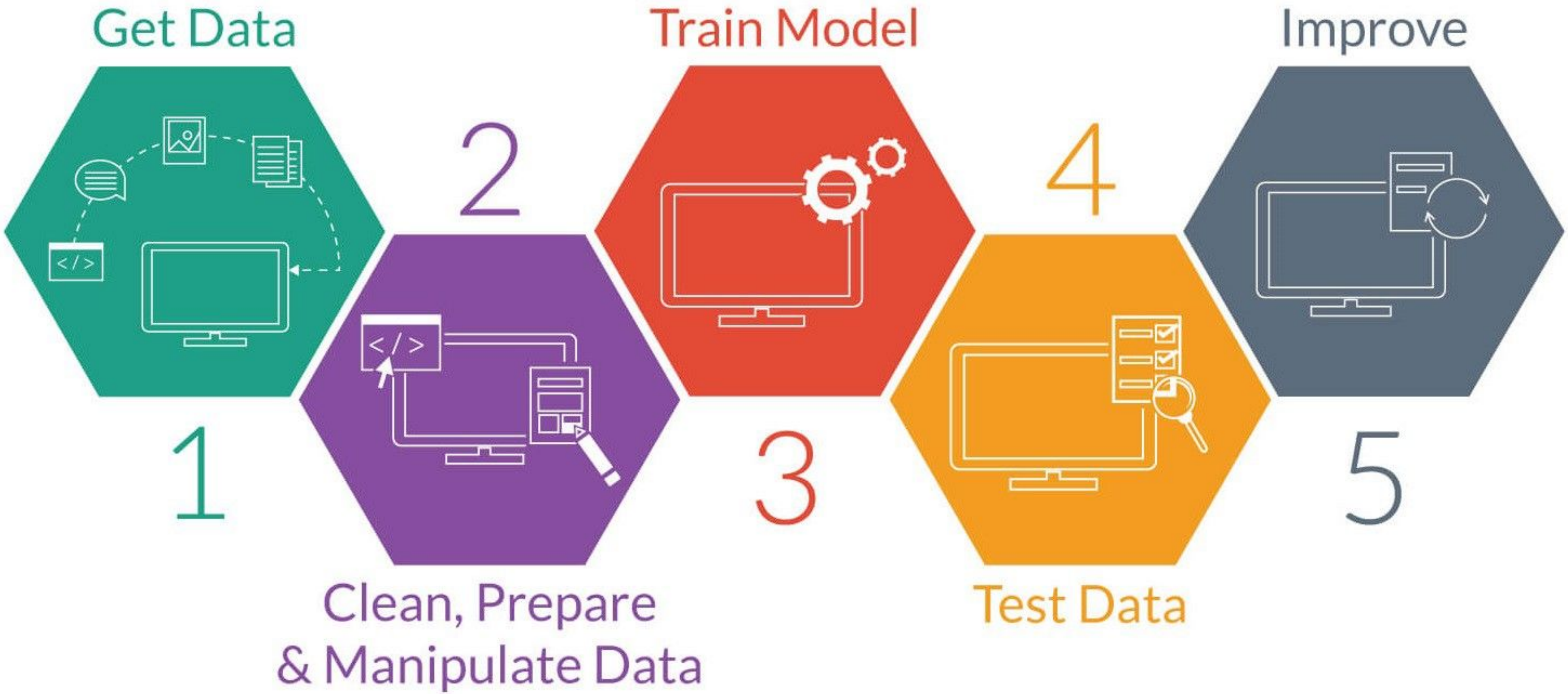
The Caret Package.

The **caret** package (short for Classification And REgression Training) is a set of functions that attempt to streamline the process for creating predictive models.

The package contains tools for:

- Data splitting
- Pre-processing
- Feature selection
- Model tuning using resampling
- Variable importance estimation

The machine learning workflow



LAB I: Wine quality Dataset

The two datasets are related to red and white variants of the Portuguese "Vinho Verde" wine. For more details, consult the reference [Cortez et al., 2009]. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

These datasets can be viewed as **classification** or **regression** tasks. The classes are ordered and not balanced (e.g. there are much more normal wines than excellent or poor ones).

Available at https://archive.ics.uci.edu/ml/datasets/wine+quality

GET the Data: Dataset Available at github

https://github.com/harpomaxx/intro-mldl-r/archive/master.zip

Or via git

```
git clone https://github.com/harpomaxx/intro-mldl-r.git
```

```r
winedataset_blanco <- read_csv("data/blanco_train.csv.gz")
winedataset_red <- read_csv("data/tinto_train.csv.gz")

# Create a new feature for the type
winedataset_blanco$type="white"
winedataset_red$type="red"

# Merge both datasets into one.
winedataset<-rbind(winedataset_blanco,winedataset_red)
```

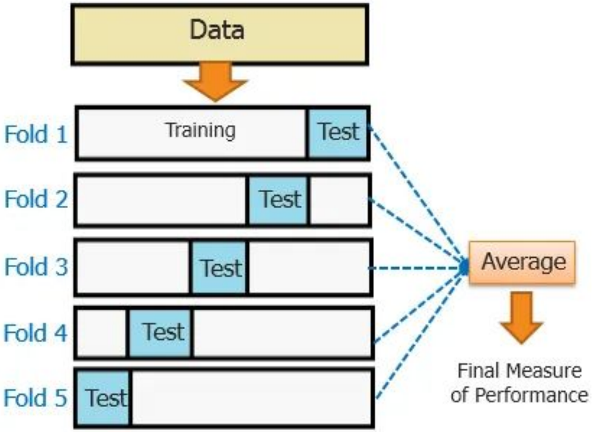
TRAIN THE MODEL: Split the dataset

```
# TRAIN THE MODEL
## Split train and test
```{r}

trainIndex <- createDataPartition(as.factor(trainset$quality), p=0.80, list=FALSE)
data_train <- trainset[ trainIndex,]
data_test <-  trainset[-trainIndex,]
colnames(data_train) <- make.names(colnames(data_train))
colnames(data_test) <- make.names(colnames(data_test))
```
```

TRAIN THE MODEL: The train control object

```
ctrl_fast <- trainControl(method="cv",
                     repeats=1,
                     number=5,
                   #  summaryFunction=twoClassSummary,
                     verboseIter=T,
                     classProbs=F,
                     allowParallel = TRUE)
```



TRAIN THE MODEL: train()

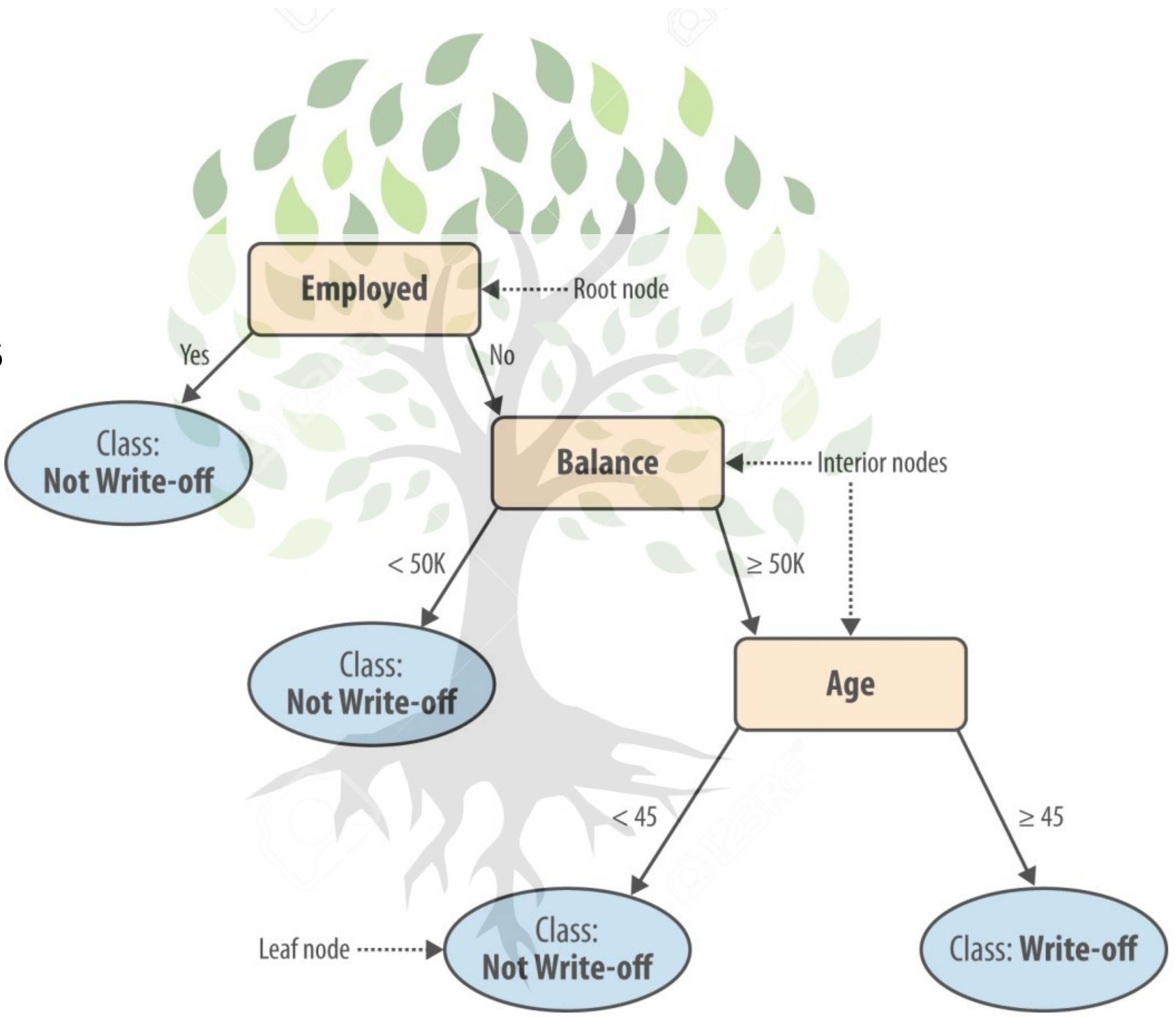
```
ctrl_fast <- trainControl(method="cv",
                          repeats=1,
                          number=5,
                        #  summaryFunction=twoClassSummary,
                          verboseIter=T,
                          classProbs=F,
                          allowParallel = TRUE)
```

```
rfFitupsam<- train(train_formula,
                   data = data_train,
                   trControl = ctrl_fast,
                   method="rpart")
```

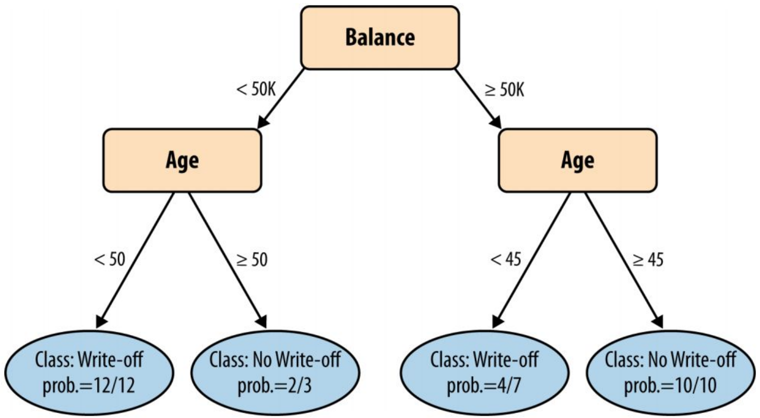
Decisión Trees

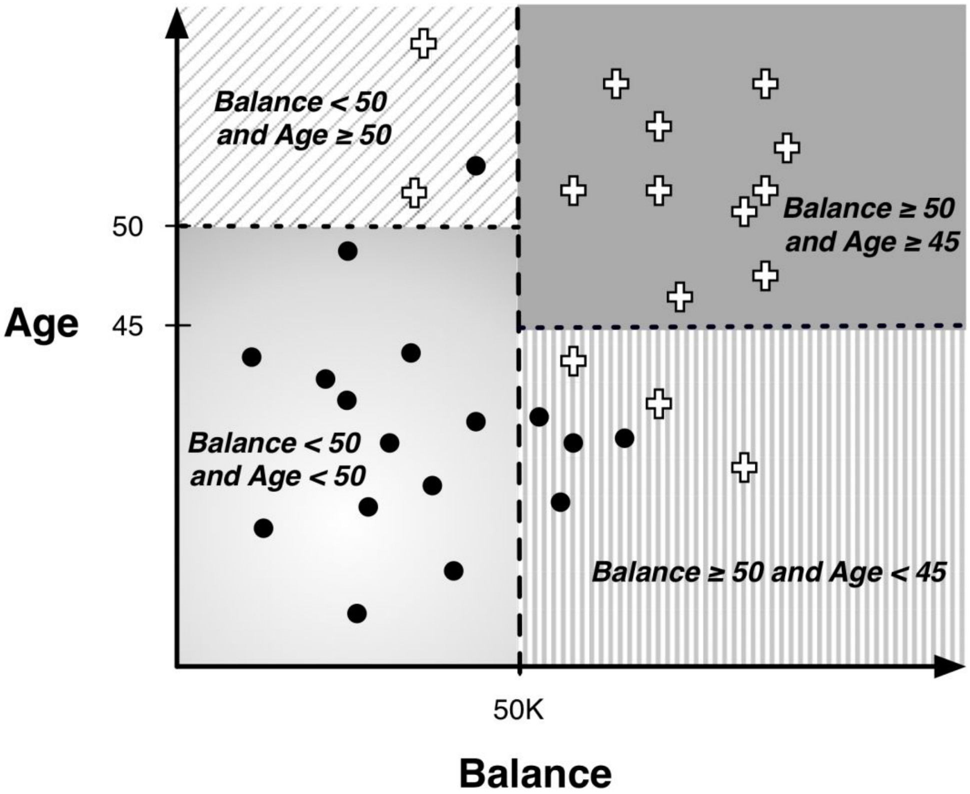
Try to segment the population into subgroups that have different values for the target variable.

Information gain (IG) measures how much an attribute improves (decreases) entropy over the whole segmentation it creates



Decision Trees: Selecting informative attributes



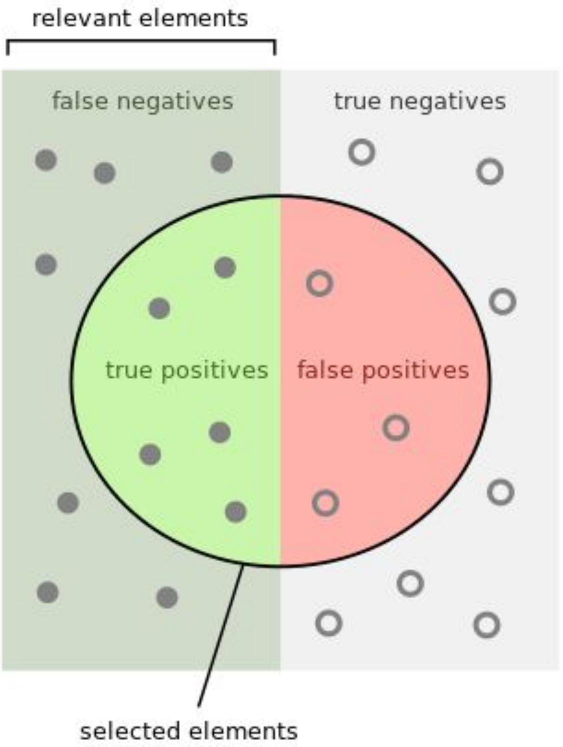


Decision Trees are Interpretables as rules

http://harpomaxx.shinyapps.io/dtdemo

Performance Metrics for Discrete classes

		Predicted		
		Yes	No	
Actual	Yes	2 (True +ve)	1 (False -ve)	2/(2+1)=⅔ **Recall (Sensitivity)**
	No	2 (False +ve)	3 (True -ve)	3/(3+2)=⅗ **(Specificity)**
		2/(2+2)=50% **(Precision)**		**Accuracy=** (2+3)/(2+1+2+3)= **5/8**



```
Prediction high low medium
    high     29   0     20
    low       0   0      0
    medium  175  40    774

Overall Statistics

               Accuracy : 0.7736
                 95% CI : (0.7469, 0.7987)
    No Information Rate : 0.7649
    P-Value [Acc > NIR] : 0.2682

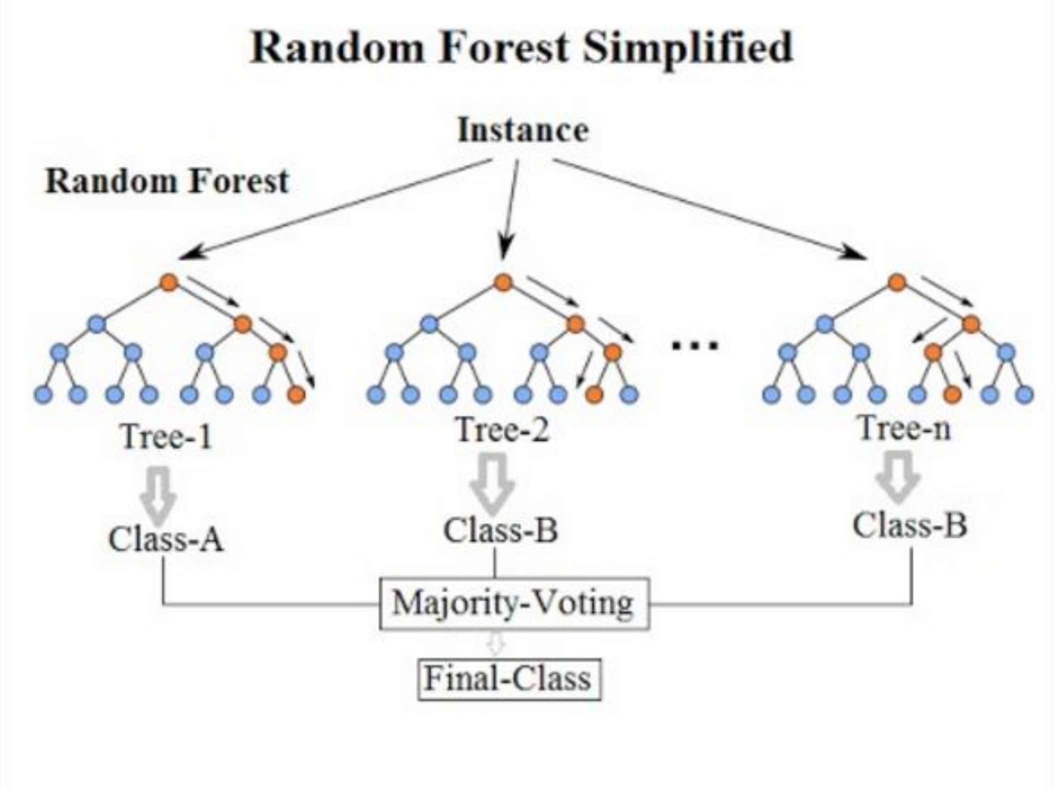
                  Kappa : 0.1356

 Mcnemar's Test P-Value : NA

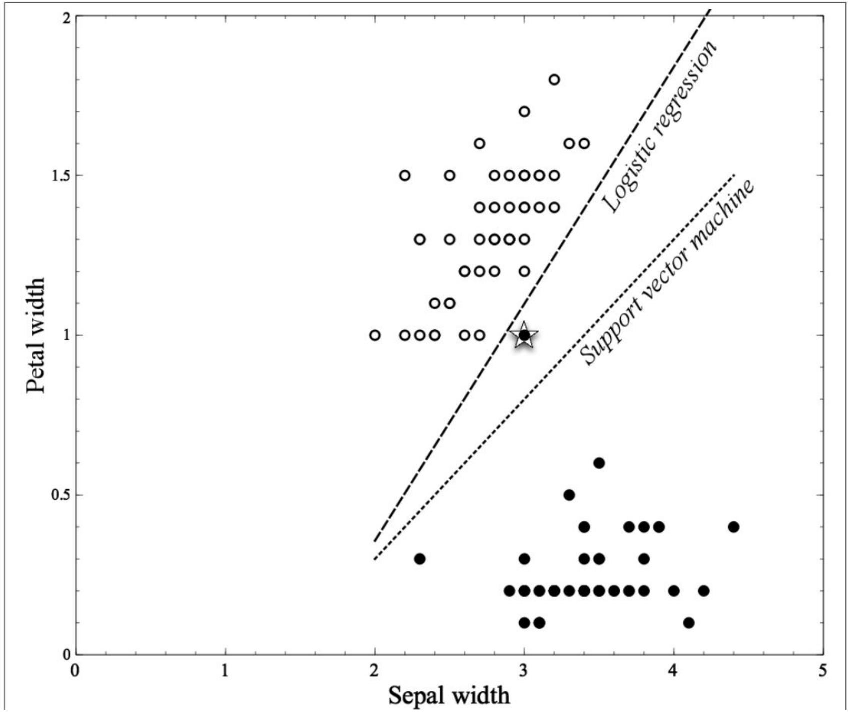
Statistics by Class:

                     Class: high Class: low Class: medium
Sensitivity              0.14216    0.00000        0.9748
Specificity              0.97602    1.00000        0.1189
```

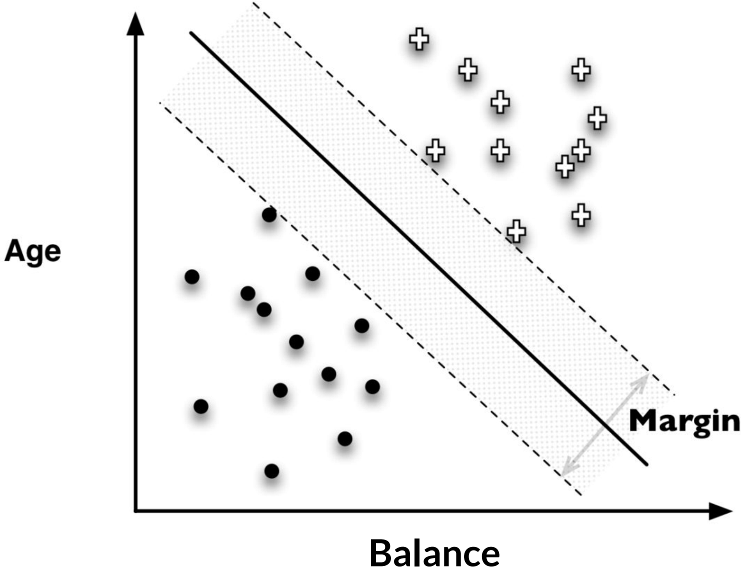
Random Forest, the first choice for tabular data



What about linear discriminants...?



Maximize the margin



CARET available models

https://topepo.github.io/caret/available-models.html

6 Available Models

The models below are available in `train`. The code behind these protocols can be obtained using the function `getModelInfo` or by going to the github repository.

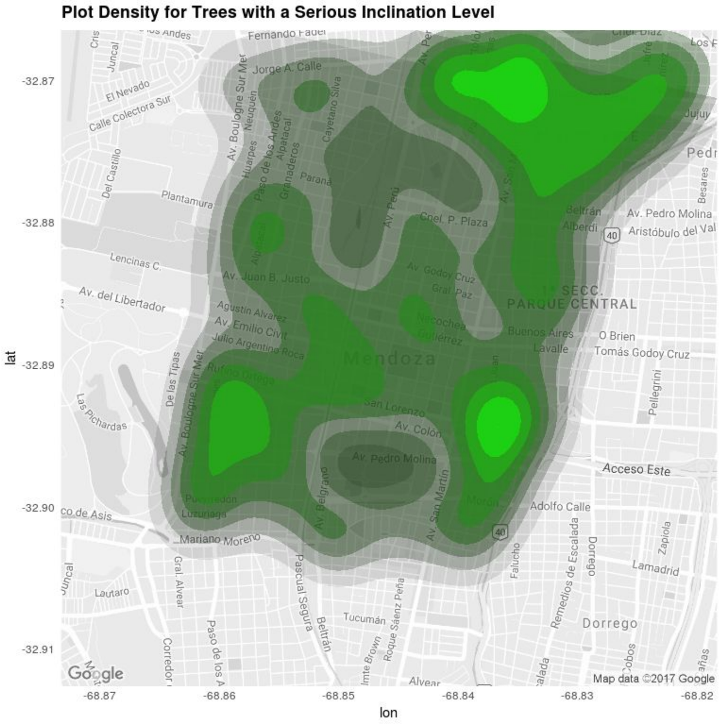
Show 238 entries

Search:

Model	*method* Value	Type	Libraries	Tuning Parameters
AdaBoost Classification Trees	adaboost	Classification	fastAdaboost	nIter, method
AdaBoost.M1	AdaBoost.M1	Classification	adabag, plyr	mfinal, maxdepth, coeflearn
Adaptive Mixture Discriminant Analysis	amdai	Classification	adaptDA	model
Adaptive-Network-Based Fuzzy Inference System	ANFIS	Regression	frbs	num.labels, max.iter

LAB II: Prediction of the inclination of trees in Mendoza City.

A Kaggle Challenge: http://bit.ly/kaggle-tree-2019



Carlos A. Catania (PhD)
(AKA **Harpo**)
LABSIN - Ingeniería - UNCuyo
@harpolabs
harpo@ingenieria.uncuyo.edu.ar

LABSIN

http://labsin.org