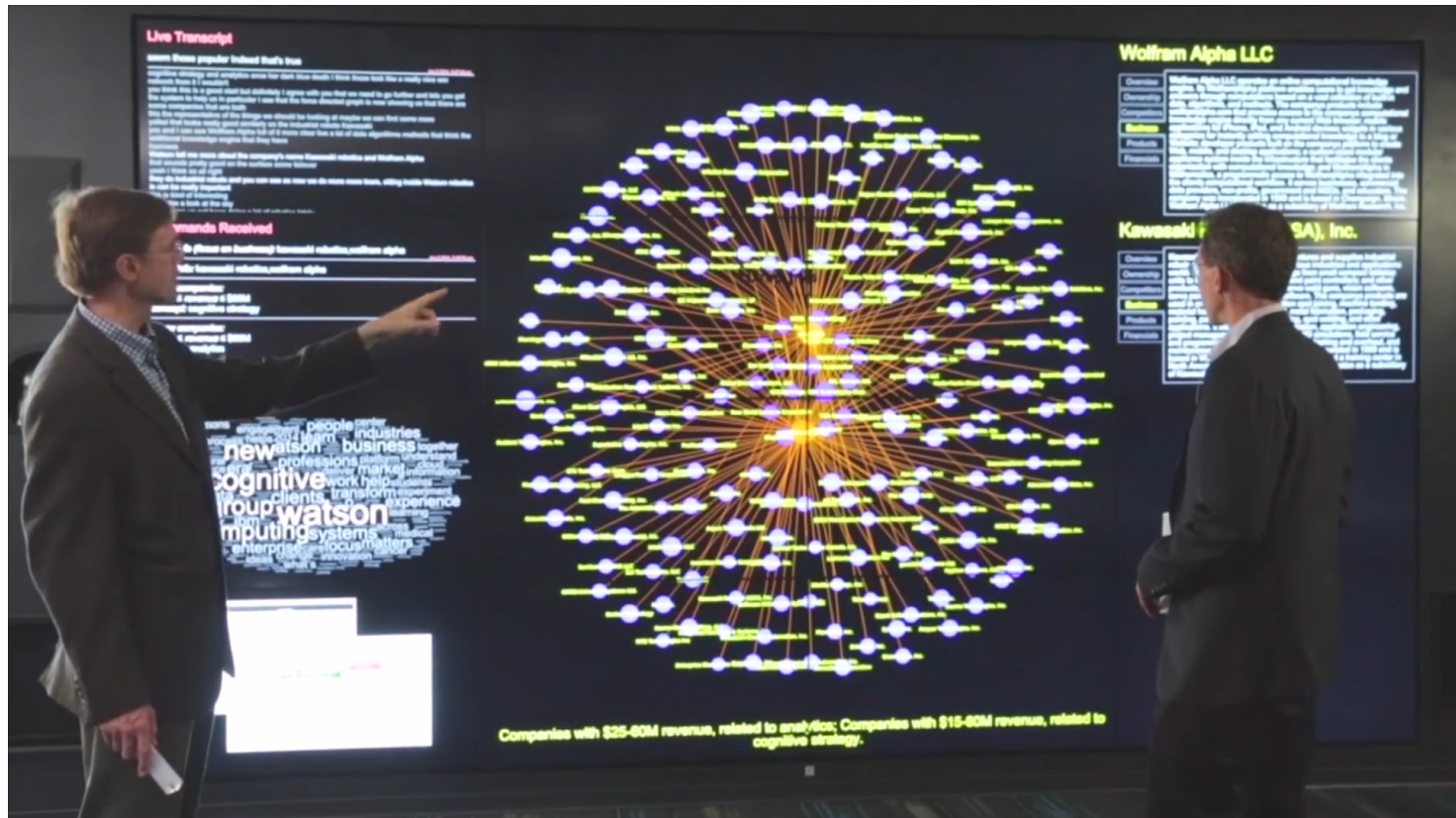


Grupo de R Madrid (10/05/2018) Machine Learning Automatizado

Presentación

- Físico. MBA y Máster BI y BigData
- Consultor Freelance
- Kaggle Máster
- Socio Asociación de Usuarios de R de España
- santiago_mota@yahoo.es
- <http://es.linkedin.com/in/santiagomota>

Dario Gil: Cognitive systems and the future of expertise TED (22/12/2014)



<https://www.youtube.com/watch?v=0heqP8d6vtQ>

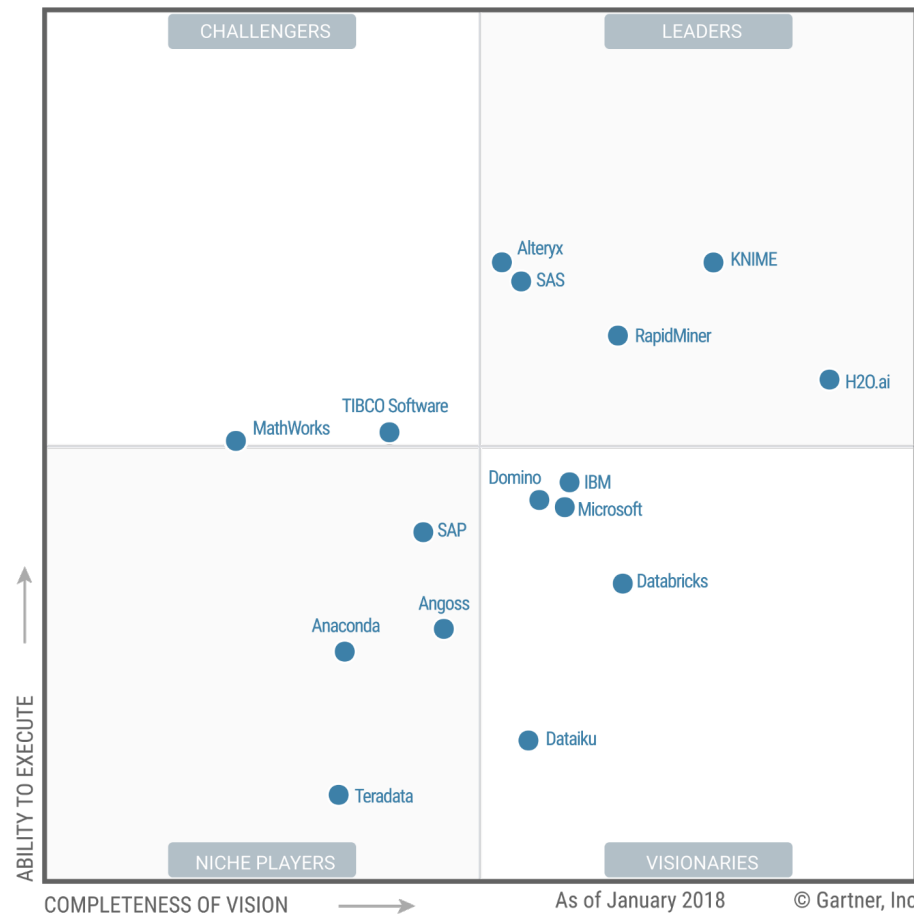
ML Automatizado / <https://github.com/santiagomota>



IBM anuncia Watson Analytics, un servicio de analítica cognitiva de negocio (19/09/2014)

- *“IBM Watson Analytics automatiza, además, algunos pasos del análisis, como la preparación de los datos, el análisis predictivo y la visualización...”*
- *“Diálogo en lenguaje natural: el nuevo producto entiende el lenguaje natural, por lo que solo es necesario teclear lo que al usuario le gustaría ver...”*
- *“Analítica predictiva “guiada”: el servicio es capaz de guiar al usuario en patrones y resultados de los datos en los que el usuario tradicionalmente no se fijaría”.*

Gartner. Data Science y ML Platforms



<https://www.gartner.com/doc/reprints?id=1-4RQ3VEZ&ct=180223&st=sb>

ML Automatizado / <https://github.com/santiagomota>



Tools that Data Scientists actually use



https://thomaswdinsmore.com/2018/02/26/notes-on-gartners-2018-data-science-and-machine-learning-mq/?lipi=urn%3Ali%3Apage%3Ad_flagship3_feed%3BvtLwdqeRTBCwlam84Qf%2BOW%3D%3D

ML Automatizado / <https://github.com/santiagomota>



Proyecto de datos

id	superficie_sq_ft	tipo	parcela_acres	habitaciones	banos	precio_venta
1	719	Casa	1,64	1	1	88.000
2	2.017	Apartamento		3	2	164.000
3	697	Apartamento		1	1	72.000
4	948	Casa	1,02	2	3	85.000
5	3.375	Apartamento		3	4	271.000
6	3.968	Apartamento		4	4	482.000
7	790	Apartamento		1	2	88.000
8	1.341	Casa	0,66	3	3	128.000
9	2.379	Apartamento		3	3	235.000
10	2.495	Casa	0,21	3	4	309.000
11	1.356	Apartamento		1	1	163.000
12	3.361	Casa	1,64	3	4	375.000
13	1.060	Casa	0,05	1	1	98.000
14	582	Casa	0,61	1	1	50.000
15	1.640	Apartamento		2	3	145.000
16	3.546	Casa	0,40	4	4	394.000
17	903	Apartamento		2	2	82.000
18	1.096	Casa	0,40	3	4	105.000
19	1.280	Casa	0,15	2	2	129.000
20	1.139	Apartamento		1	1	106.000

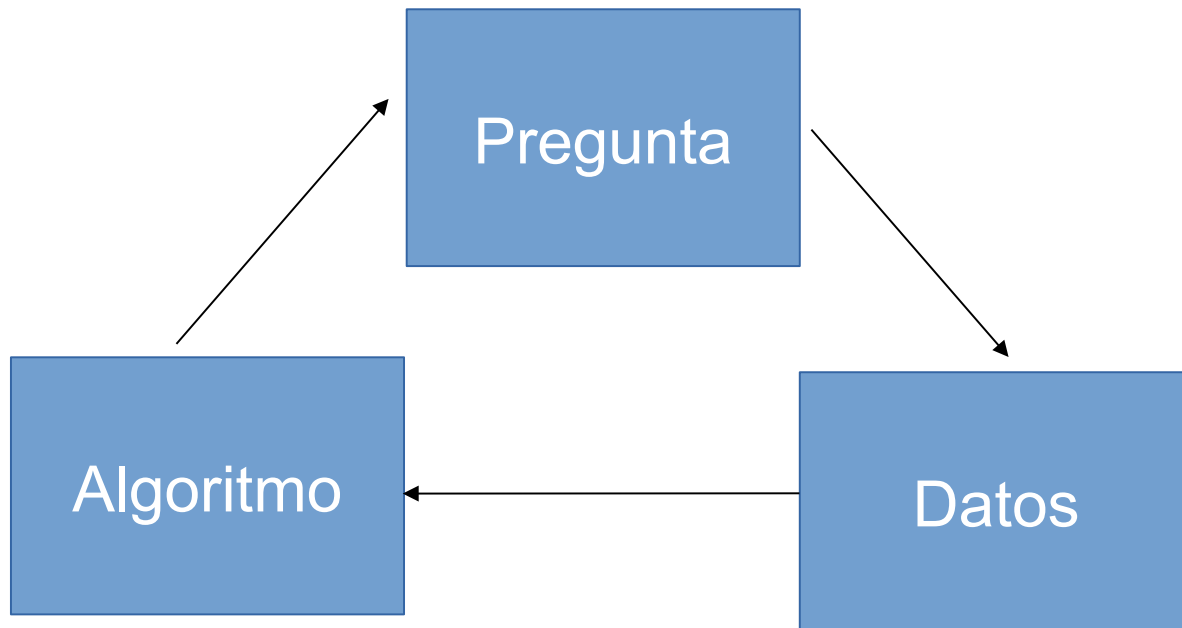
Proyecto de datos

id	superficie_sq_ft	tipo	parcela_acres	habitaciones	banos	precio_venta		
1	719	Casa	1,64	1	1	88.000		
2	2.017	Apartamento		3	2	164.000		
3	697	Apartamento		1	1	72.000		
4	948	Casa	1,02	2	3	85.000		
5	3.375	Apartamento		3	4	271.000		
6	3.968	Apartamento		4	4	482.000		
7	790	Apartamento		1	2	88.000		
8	1.341	Casa	0,66	3	3	128.000		
9	2.379	Apartamento		3	3	235.000		
10	2.495	Casa	0,21	3	4	309.000		
11	1.356	Apartamento		1	1	163.000		
12	3.361	Casa	1,64	3	4	375.000		
13	1.060	Casa	0,05	1	1	98.000	Prediccion	Error
14	582	Casa	0,61	1	1	50.000	41.000	-9.000
15	1.640	Apartamento		2	3	145.000	165.000	20.000
16	3.546	Casa	0,40	4	4	394.000	380.000	-14.000
17	903	Apartamento		2	2	82.000	76.000	-6.000
18	1.096	Casa	0,40	3	4	105.000	128.000	23.000
19	1.280	Casa	0,15	2	2	129.000	115.000	-14.000
20	1.139	Apartamento		1	1	106.000	94.000	-12.000

Proyecto de datos

- Hay casas con mas baños que habitaciones
- División Casa / Apartamento
- Elementos vacíos, outliers (ELT)
- Nuevas columnas (feature engineering)
- Cross Validation
- Nuevos algoritmos
- De donde vienen los datos y, sobre todo: **Cuenta de resultados**

Proyecto de datos



Machine Learning automatizado

- Trifacta Wrangler ([link](#))
- IBM Watson ([link](#))
- Datarobot ([link](#))
- Daitaku ([link](#))
- Domino ([link](#))
- Seldon ([link](#))
- Alterix ([link](#))
- H2O

H2O

- Creada en 2011 (inicialmente Oxddata)
- Noviembre 2015: +\$20M (B) ya tenían \$14M
- Noviembre 2017: +\$40M (C) Total **\$75M**. (Nvidia, Wells Fargo)
- Personas
- Kaggle

Oferta de H2O

Getting Started & User Guides

Open Source | Commercial

H2O

What is H2O?
[H2O User Guide](#) (Main docs)
[H2O Book](#) (O'Reilly)
[Recent Changes](#)
[Open Source License](#) (Apache V2)

Quick Start Video - Flow Web UI
Quick Start Video - R
Quick Start Video - Python

Download H2O

Sparkling Water

What is Sparkling Water?
Sparkling Water User Guide 2.0 2.1 2.2
[Sparkling Water Booklet](#)
[RSparkling Readme](#)

PySparkling Readme	2.0	2.1	2.2
Recent Changes	2.0	2.1	2.2

[Open Source License](#) (Apache V2)

Quick Start Video - Scala

Download Sparkling Water

Driverless AI

What is Driverless AI?
[Driverless AI User Guide](#) HTML PDF
[Recent Changes](#)
[Driverless AI Booklet](#)
[MLI with Driverless AI Booklet](#)

Driverless AI Webinars

Download Driverless AI

H2O4GPU (alpha)

[H2O4GPU Readme](#)
[Open Source License](#) (Apache V2)

Download H2O4GPU

Enterprise Steam

Enterprise Steam Installation Guide	HTML	PDF
Enterprise Steam User Guide	HTML	PDF

Get Enterprise Steam
(sales@h2o.ai)

Steam

What is Steam?
[Steam User Guide](#)
[Recent Changes](#)
[Open Source License](#) (AGPL)

Download Steam

Deep Water (preview)

[Deep Water Readme](#)
[Deep Water Booklet](#)
[Deep Water AMI Guide](#)
[Deep Water Docker Image](#)
[Open Source License](#) (Apache V2)

Launch Deep Water AMI
(choose p2.xlarge)

http://docs.h2o.ai/?_ga=2.107667714.1485748875.1520325919-538902739.1512117166

ML Automatizado / <https://github.com/santiagomota>



H2O

- Basada en java
- Facilidades para escalar
- Paralización. Para R, substituto data.table
- Maquina local, cluster o en cloud
- Funciona como API, pero tiene navegador
- Acceso desde R o Python
- Pagina ([link](#)), blog ([link](#)) y para iniciarse ([link](#) y [link](#))

Localhost H2O

H₂O FLOW

Flow Cell Data Model Score Admin Help

Untitled Flow

getFrame

splitFrame

mergeFrames

getModels

getGrids

getPredictions

getJobs

buildModel

runAutoML

importModel

predict

Split a frame into two or more frames

Merge two frames into one

Get a list of models in H₂O

Get a list of grid search results in H₂O

Get a list of predictions in H₂O

Get a list of jobs running in H₂O

Build a model

Automatically train and tune many models

Import a saved model

Make a prediction

CS

getJobs

125ms

Jobs

Type	Destination	Description	Start Time	End Time	Run Time	Status
Frame	training	Parse	2018-03-12 10:11:00	2018-03-12 10:11:06	00:00:05.999	DONE
Frame	validating	Parse	2018-03-12 10:11:11	2018-03-12 10:11:12	00:00:00.756	DONE
Frame	testing	Parse	2018-03-12 10:11:13	2018-03-12 10:11:14	00:00:00.189	DONE
Auto Model	AutoML_20180312_101114	AutoML build	2018-03-12 10:11:14	2018-03-12 11:45:53	01:34:39.7	RUNNING
Model	Quantiles_model_1520845829907_1	Quantiles	2018-03-12 10:11:15	2018-03-12 10:11:15	00:00:00.31	DONE
Model	Quantiles_model_1520845829907_2	Quantiles	2018-03-12 10:11:15	2018-03-12 10:11:15	00:00:00.0	DONE
Model	Quantiles_model_1520845829907_3	Quantiles	2018-03-12 10:11:15	2018-03-12 10:11:15	00:00:00.0	DONE
Model	DRF_0_AutoML_20180312_101114	DRF	2018-03-12 10:11:15	2018-03-12 10:35:30	00:24:15.435	DONE
Model	XRT_0_AutoML_20180312_101114	DRF	2018-03-12 10:35:31	2018-03-12 11:04:25	00:28:54.356	DONE
Grid	GLM_grid_0_AutoML_20180312_101114	GLM Grid Search	2018-03-12 11:04:26	2018-03-12 11:04:26	00:00:00.244	DONE
Grid	GBM_grid_0_AutoML_20180312_101114	GBM Grid Search	2018-03-12 11:04:27	2018-03-12 11:18:57	00:14:30.348	DONE
Grid	GBM_grid_0_AutoML_20180312_101114	GBM Grid Search	2018-03-12 11:18:58	2018-03-12 11:33:09	00:14:10.541	DONE
Grid	GBM_grid_0_AutoML_20180312_101114	GBM Grid Search	2018-03-12 11:33:09	2018-03-12 11:45:53	00:12:43.496	RUNNING

Ready

OUTLINE FLOWS CLIPS HELP

Help

Using Flow for the first time?

Quickstart Videos

Or, view example Flows to explore and learn H₂O.

STAR H₂O ON GITHUB!

Star 2,907

GENERAL

- Flow Web UI ...
- ... Importing Data
- ... Building Models
- ... Making Predictions
- ... Using Flows
- ... Troubleshooting Flow

EXAMPLES

Flow packs are a great way to explore and learn H₂O. Try out these Flows and run them in your browser.

Browse installed packs...

H₂O REST API

- Routes
- Schemas

Connections: 0 H₂O

Instalación desde R

[DOWNLOAD AND RUN](#)[INSTALL IN R](#)[INSTALL IN PYTHON](#)[INSTALL ON HADOOP](#)[USE FROM MAVEN](#)

Use H₂O directly from R

Copy and paste these commands into R one line at a time:

```
# The following two commands remove any previously installed H2O packages for R.
if ("package:h2o" %in% search()) { detach("package:h2o", unload=TRUE) }
if ("h2o" %in% rownames(installed.packages())) { remove.packages("h2o") }

# Next, we download packages that H2O depends on.
pkgs <- c("RCurl","jsonlite")
for (pkg in pkgs) {
  if (!(pkg %in% rownames(installed.packages()))) { install.packages(pkg) }
}

# Now we download, install and initialize the H2O package for R.
install.packages("h2o", type="source", repos="http://h2o-release.s3.amazonaws.com/h2o/rel-wolpert
/4/R")

# Finally, let's load H2O and start up an H2O cluster
library(h2o)
h2o.init()
```



<http://h2o-release.s3.amazonaws.com/h2o/rel-wolpert/4/index.html>

ML Automatizado / <https://github.com/santiagomota>



Conectar dos servidores

CLOUD STATUS

✓ HEALTHY ✓ CONSENSUS 🔒 LOCKED

Version	Started	Nodes (Used / All)
3.18.0.2	3 minutes ago	2 / 2

NODES

Name	Ping	Cores	Load	My CPU %	Sys CPU %	GFLOPS	Memory	Bandwidth	Data (Used/Total)	Data (% Cached)	GC (Free / Total / Max)
✓ 192.168.1.68:55555	a few seconds ago	16	0.032	-1	-1	13.799	11.54 GB	/ s	- / NaN undefined	NaN%	12.84 GB / NaN undefined / 13.33 GB
✓ 192.168.1.148:55555	a few seconds ago	4	0.510	-1	-1	12.229	17.03 GB	/ s	- / NaN undefined	NaN%	6.95 GB / NaN undefined / 6.97 GB
✓ TOTAL	-	20	0.542	-	-	26.028	28.57 GB	/ s	- / NaN undefined	NaN%	19.78 GB / NaN undefined / 20.30 GB

🔄 Refresh

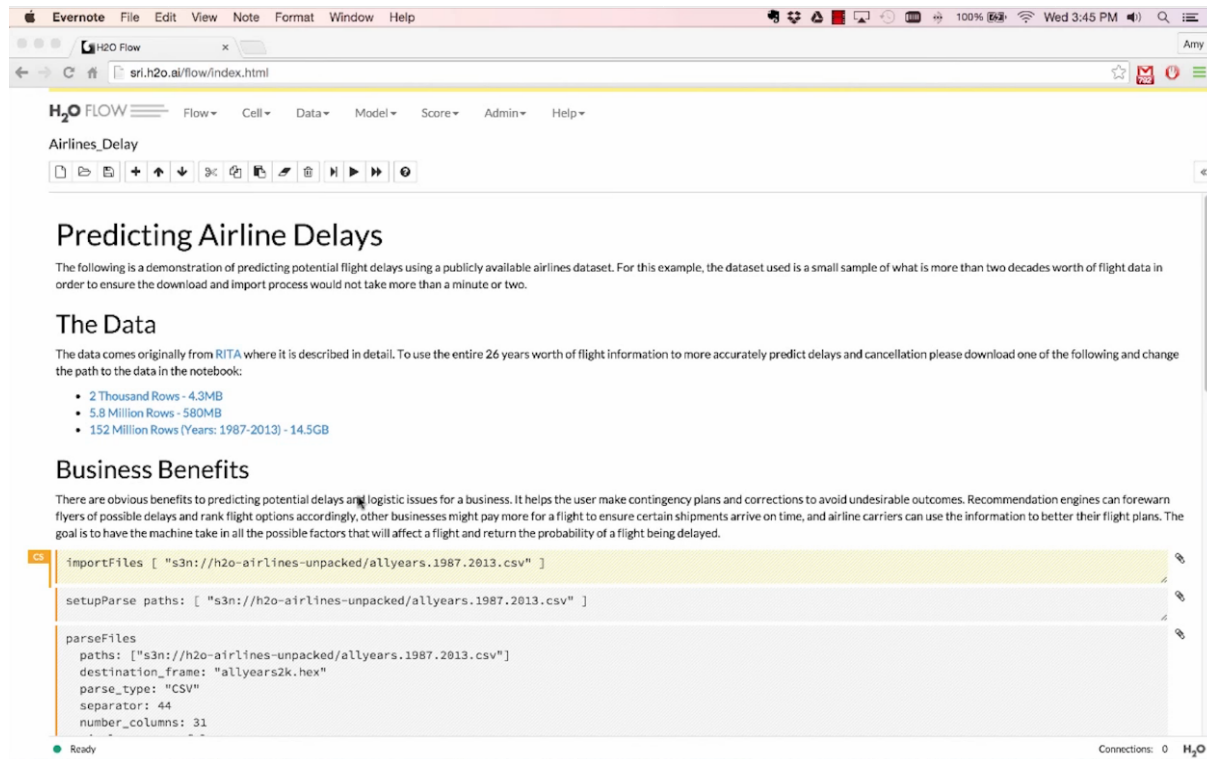
<http://docs.h2o.ai/h2o/latest-stable/h2o-docs/faq/tunneling.html>

ML Automatizado / <https://github.com/santiagomota>



Analizar retrasos en vuelos con H2O

- Video ([link](#)), en flow ([link](#)), en R ([link](#)) y datos ([link](#))



The screenshot shows the H2O Flow web interface in a browser. The top navigation bar includes 'Flow', 'Cell', 'Data', 'Model', 'Score', 'Admin', and 'Help'. The main content area is titled 'Airlines_Delay' and contains a notebook with the following sections:

Predicting Airline Delays

The following is a demonstration of predicting potential flight delays using a publicly available airlines dataset. For this example, the dataset used is a small sample of what is more than two decades worth of flight data in order to ensure the download and import process would not take more than a minute or two.

The Data

The data comes originally from [BITA](#) where it is described in detail. To use the entire 26 years worth of flight information to more accurately predict delays and cancellation please download one of the following and change the path to the data in the notebook:

- 2 Thousand Rows - 4.3MB
- 5.8 Million Rows - 580MB
- 152 Million Rows (Years: 1987-2013) - 14.5GB

Business Benefits

There are obvious benefits to predicting potential delays and logistic issues for a business. It helps the user make contingency plans and corrections to avoid undesirable outcomes. Recommendation engines can forewarn flyers of possible delays and rank flight options accordingly, other businesses might pay more for a flight to ensure certain shipments arrive on time, and airline carriers can use the information to better their flight plans. The goal is to have the machine take in all the possible factors that will affect a flight and return the probability of a flight being delayed.

```
importFiles [ "s3n://h2o-airlines-unpacked/allyears.1987.2013.csv" ]

setupParse paths: [ "s3n://h2o-airlines-unpacked/allyears.1987.2013.csv" ]

parseFiles
paths: ["s3n://h2o-airlines-unpacked/allyears.1987.2013.csv"]
destination_frame: "allyears2k.hex"
parse_type: "CSV"
separator: 44
number_columns: 31
```

At the bottom, it shows 'Ready' and 'Connections: 0 H2O'.

<http://university.h2o.ai/data-science-101/lesson2.html>

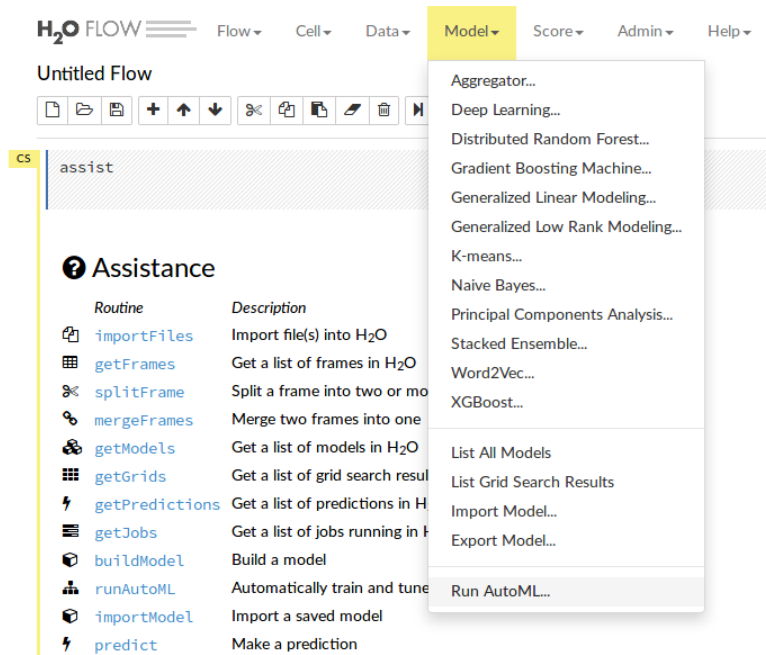
ML Automatizado / <https://github.com/santiagomota>



AutoML

- Sólo hay que darle el dataset, target y tiempo
- Básicamente hace un stacking de modelos

```
tiempo_inicio <- Sys.time()
automl_models_h2o <- h2o.automl(
  x          = x,
  y          = y,
  training_frame = training_h2o,
  validation_frame = validating_h2o,
  # leaderboard_frame = test_h2o,
  max_runtime_secs = 6000, # 180
  stopping_metric = "AUTO")
print(Sys.time()-tiempo_inicio)
```



Demo. Crímenes en L.A.

- Basada en estos posts ([link1](#) y [link2](#))
- Con datos de opendata de Los Ángeles ([link](#)). Hay que bajarlos
- Los datos necesitan de tratamiento previo
- Necesitaría mas ETL y mas feature engineering
- Página de github ([link](#))

DriverlessAI

- Licencia
- Coste (precio anual + equipos)
- Docker
- Vídeo

DriverlessAI. Requerimientos

- 64G de RAM
- GPU con CUDA (Pascal o Volta)
- Docker (o Nvidia docker)
- Cloud, Server, Desktop
- Linux, Mac, Windows 10

<http://docs.h2o.ai/driverless-ai/latest-stable/docs/userguide/installing.html>

ML Automatizado / <https://github.com/santiagomota>



DriverlessAI

< H2O.ai Experiment

1.0.20

DATASETS EXPERIMENTS MLI H2O-3 HELP PY_CLIENT LOGOUT H2OAI

TRAINING DATA

DATASET

CreditCard_train.csv

ROWS

17K

COLUMNS

25

DROPPED COLS

1

VALIDATION DATASET

Yes

CreditCard_valid.csv

TEST DATASET

Yes

CreditCard_test.csv

TARGET COLUMN

default payment next

FOLD COLUMN

--

WEIGHT COLUMN

--

TIME COLUMN

[AUTO]

TYPE

bool

COUNT

16784

UNIQUE

2

FREQ

3740

EXPERIMENT SETTINGS HELP

8

ACCURACY

2

TIME

8

INTERPRETABILITY

CLASSIFICATION

REPRODUCIBLE

ENABLE GPUS

SCORER

BINI

MCC

F1

LOGLOSS

AUC

AUCPR

LAUNCH EXPERIMENT

<http://docs.h2o.ai/driverless-ai/latest-stable/docs/userguide/launching.html>

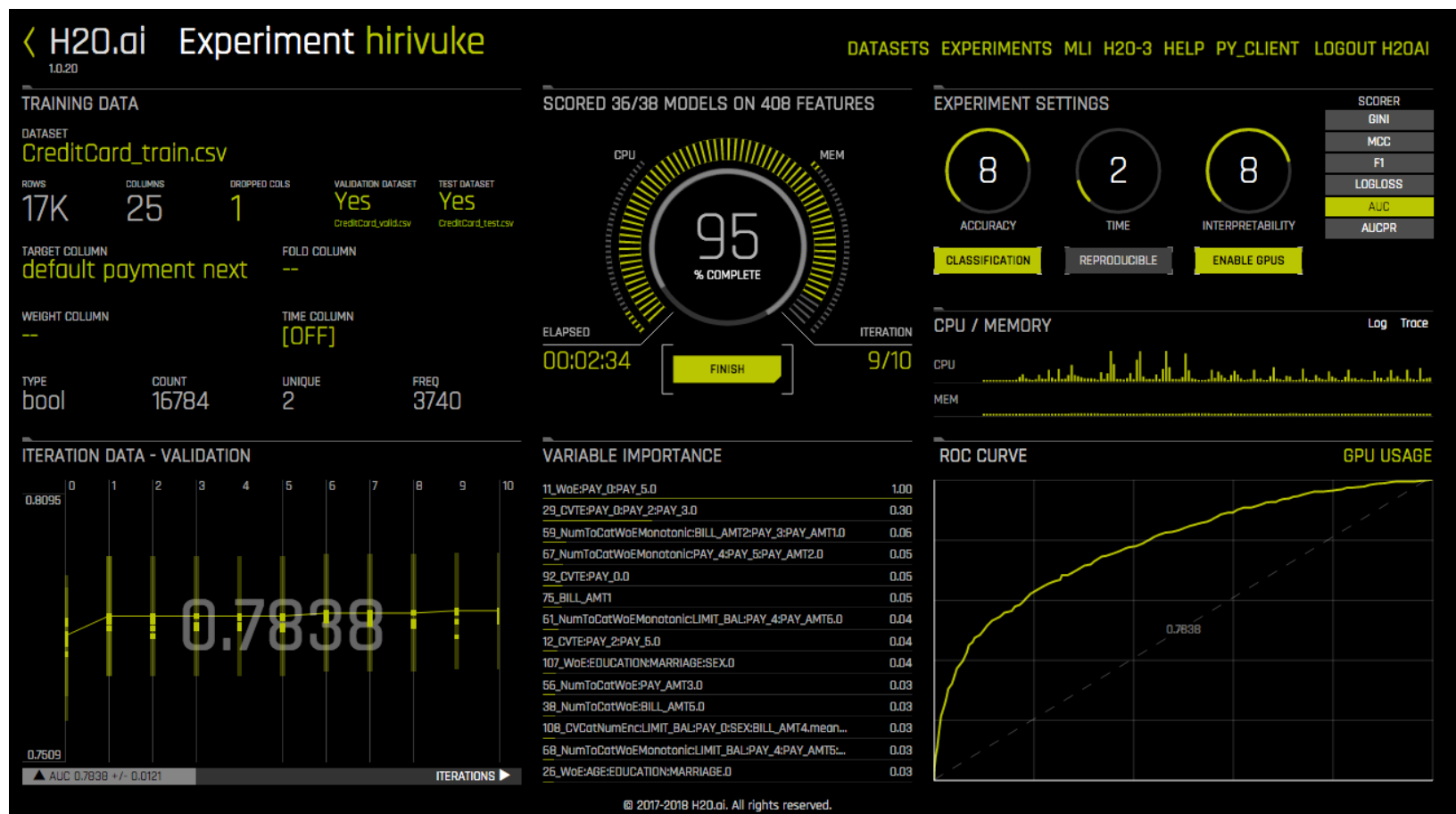
ML Automatizado / <https://github.com/santiagomota>

The image shows the Creative Commons license logo, specifically the CC BY NC SA version, which stands for Attribution, Non-Commercial, and Share-Alike.

DriverlessAI. Parámetros

- Speed
- Accuracy
- Interpretability
- Train / [Test] / [Validation]
- Target
- Scorer

DriverlessAI

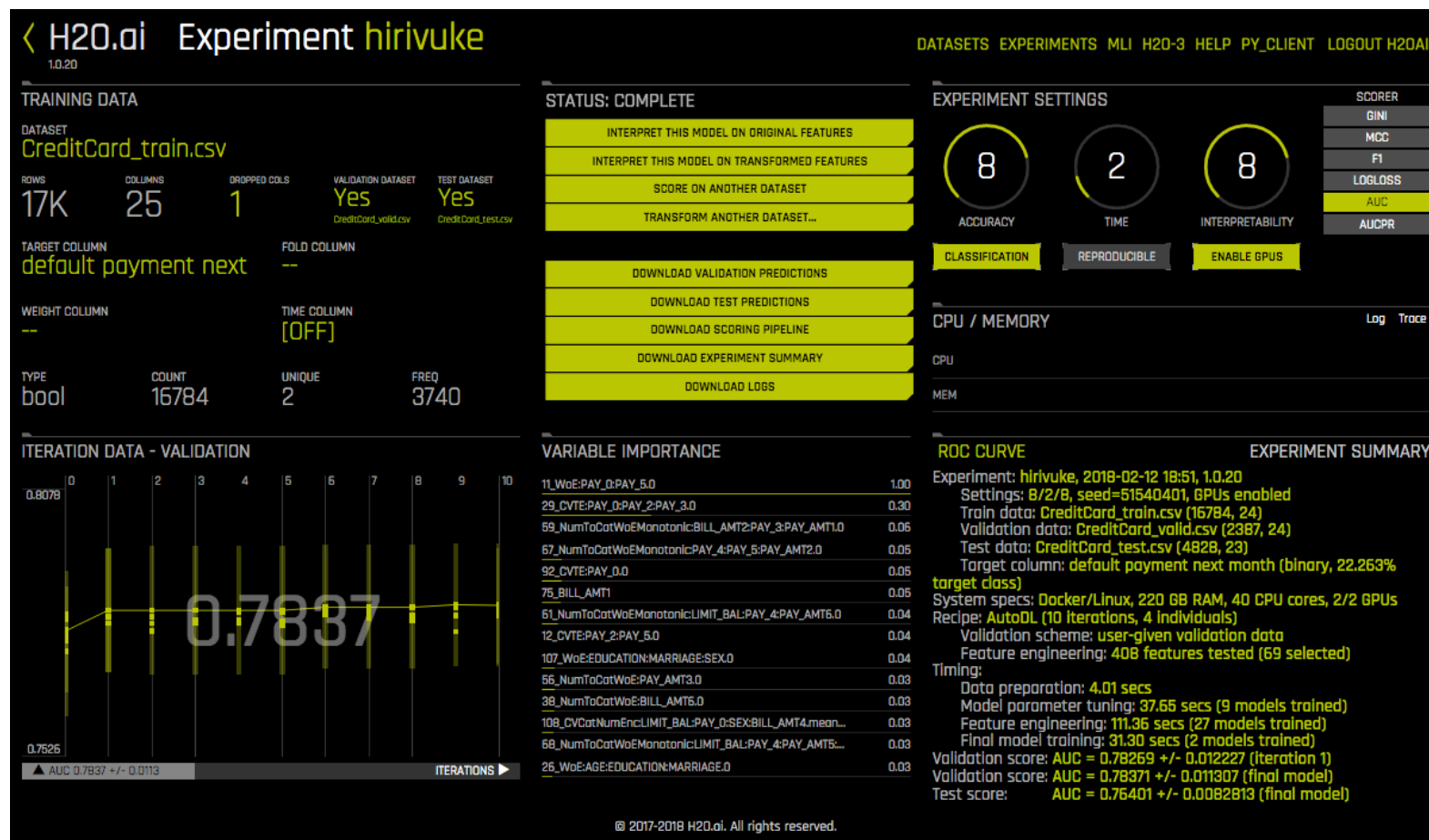


<http://docs.h2o.ai/driverless-ai/latest-stable/docs/userguide/launching.html>

ML Automatizado / <https://github.com/santiagomota>



DriverlessAI



https://www.youtube.com/watch?time_continue=43&v=KkvWX3FD7y

ML Automatizado / <https://github.com/santiagomota>



DriverlessAI. Prueba en Kaggle Favorita

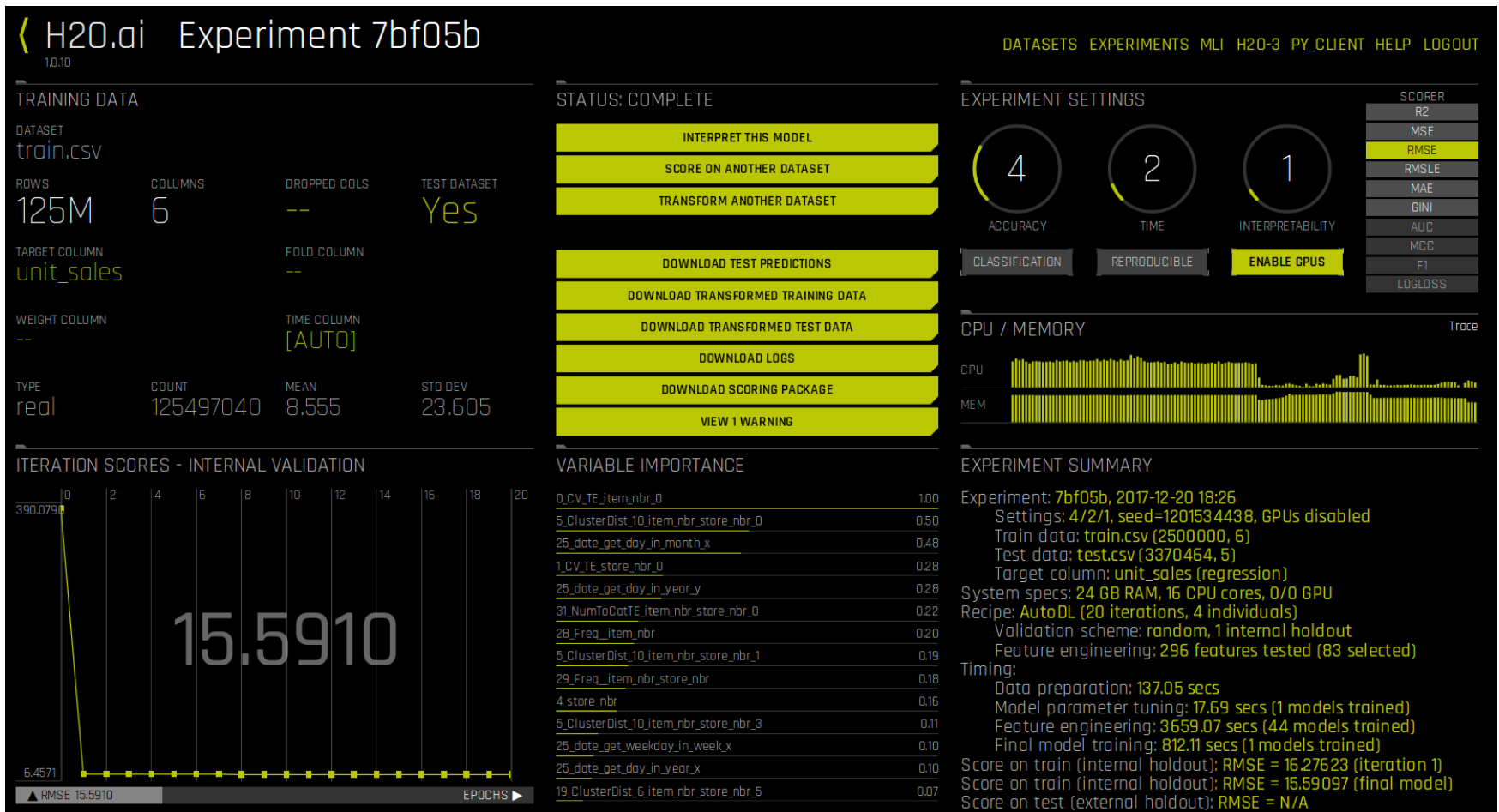
- Concurso Kaggle Favorita ([link](#))
- Estación de trabajo Z800. 16 cores. 24G RAM sin GPU
- Posición final: 126 de 1675 (medalla de bronce)
- Métrica: Normalized Weighted Root Mean Squared Logarithmic Error
- Mis mejores resultados: 0,520 (combinado) y 0,521 con un modelo LGBM.
- Resultado del ganador: 0,509
- Mejor resultado DriverlessAI: 1,240 (posición 1.131)

<https://www.kaggle.com/c/favorita-grocery-sales-forecasting/leaderboard>

ML Automatizado / <https://github.com/santiagomota>

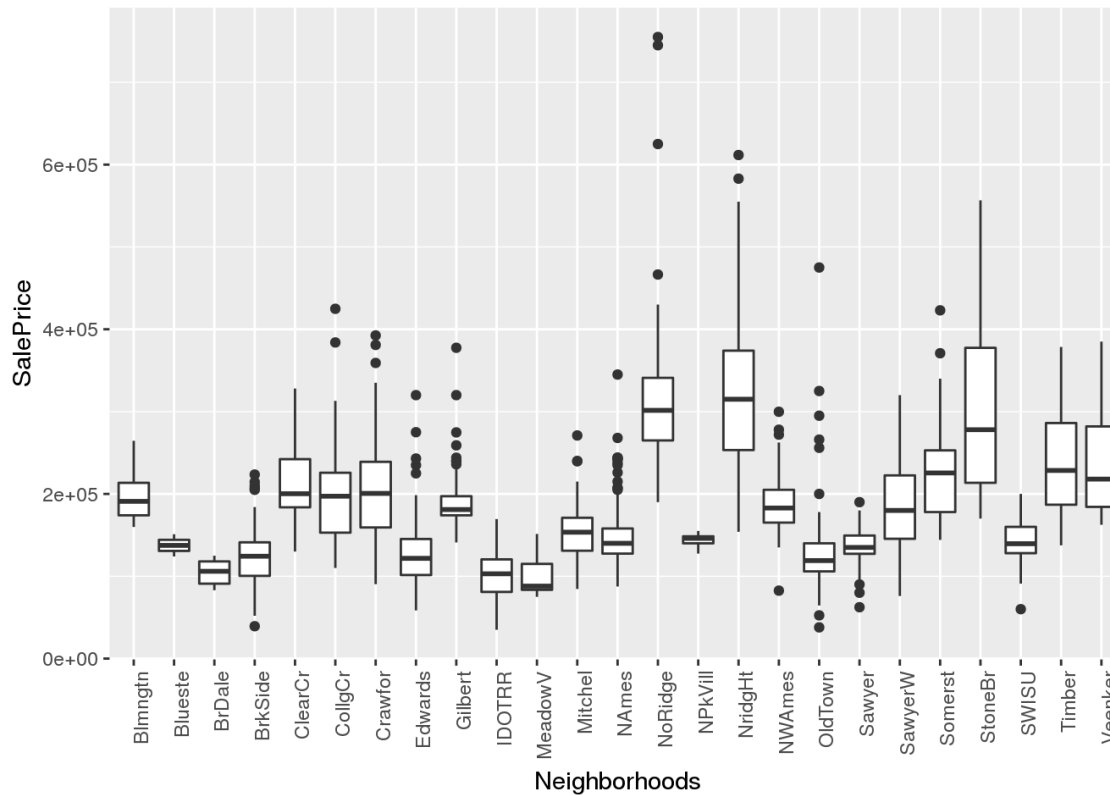


Una solución: Pred028. 1,264. 1:20:00



Concurso precios de alquileres (Kaggle)

- [Página del concurso](#). [Tutoriales](#). [Kernels](#) y [Foro](#)



Conclusiones

- En muy poco tiempo (¿este año?) vamos a tener herramientas comerciales de Machine Learning Automatizado como DriverlessAI
- Inicialmente su uso tendrá sentido en determinados escenarios
- La herramienta H2O (gratuita) tiene mucho sentido, por su capacidad de escalar, sus distintas interfaces y sus posibilidades de paralelización
- AutoML tiene aún mucho camino que recorrer

GRACIAS

Datos de contacto:

Santiago Mota Herce

E-mail: santiago_mota@yahoo.es

Github: <https://github.com/santiagomota>

LinkedIn: <https://es.linkedin.com/in/santiagomota>

