

GESTIÓN DE MODELOS PYTHON CON RETICULATE

Seguimiento Continuo y Aplicaciones
Empresariales

Francisco J. Rodríguez Aragón

Ph. D in Statistics, Associate Professional Risk Manager, Operational Risk Manager

Head of Advanced Analytics

Carrefour Financial Services, Spain

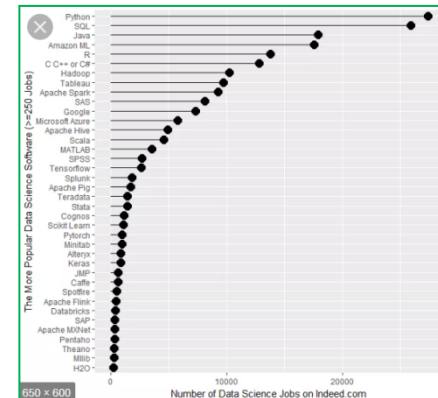
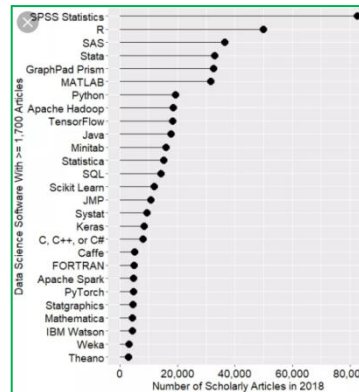
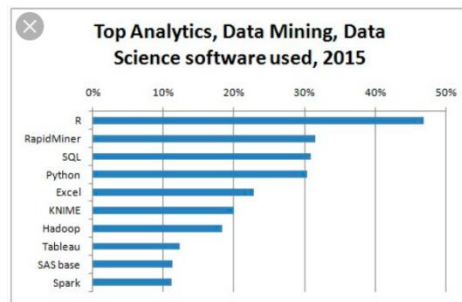
ÍNDICE

- INTRODUCCIÓN
- GESTIÓN INTEGRAL DE MODELOS
- DIVERSIDAD DE MODELOS ESTADÍSTICOS
- GESTIÓN INTEGRAL DE MODELOS CON R
- CONCLUSIONES FINALES

- **INTRODUCCIÓN**
- GESTIÓN INTEGRAL DE MODELOS
- DIVERSIDAD DE MODELOS ESTADÍSTICOS
- GESTIÓN INTEGRAL DE MODELOS CON R
- CONCLUSIONES FINALES

INTRODUCCIÓN: Múltiples Perfiles DS

- Los dos lenguajes que más suelen utilizar los DSs resultan ser R y Python (en cuanto a software libre se refiere)



- Aunque es habitual que un DS conozca ambos, no es menos cierto que en general el conocimiento más experto recaiga en uno de ellos

¿Debe sacrificarse productividad en pos de obligar a que se programe en un determinado lenguaje? Más de un “ingenazi” está sacrificando talento en este sentido

GESTIÓN INTEGRAL DE MODELOS CON R

- Funcionamiento de la librería reticulate
 - Ejemplo sencillo de lectura de un código .py

R permite importar los elementos del código mediante llamadas py, usando su propia sintaxis

```
#Se importan la librerías a utilizar
from sklearn import datasets
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt

#Importamos los datos de la misma librería de scikit-learn
boston = datasets.load_boston()

#Forma del dataset
print('Características del dataset:')
print(boston.DESCR)
print('Cantidad de datos:')
print(boston.data.shape)

#Información de las columnas
print('Nombres columnas:')
print(boston.feature_names)

#Se selecciona solamente la columna 5 del dataset
X1 = boston.data[:, np.newaxis, 1]
X2 = boston.data[:, np.newaxis, 2]
X3 = boston.data[:, np.newaxis, 3]
X4 = boston.data[:, np.newaxis, 4]
X5 = boston.data[:, np.newaxis, 5]
X6 = boston.data[:, np.newaxis, 6]
X7 = boston.data[:, np.newaxis, 7]
X8 = boston.data[:, np.newaxis, 8]
X9 = boston.data[:, np.newaxis, 9]
X10 = boston.data[:, np.newaxis, 10]
```



```
#https://rstudio.github.io/reticulate/articles/calling_python.html

library(reticulate)

py_run_file("C:/Users/Q8868116/Documents/2019/FORMACION/GESTION_MODELOS/Sintaxis_01.py")

py$res$aic
py$res$bic
py$res$rsquared
py$res$params
py$res$pvalues
py$res$conf_int()
```

OLS Regression Results						
=====						
Dep. Variable:	y	R-squared:	0.653			
Model:	OLS	Adj. R-squared:	0.647			
Method:	Least Squares	F-statistic:	103.6			
Date:	Mon, 05 Aug 2019	Prob (F-statistic):	5.39e-108			
Time:	19:38:32	Log-Likelihood:	-1572.6			
No. Observations:	506	AIC:	3165.			
Df Residuals:	496	BIC:	3207.			
Df Model:	9					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]

x1	0.0323	0.016	2.067	0.039	0.002	0.063
x2	3.7812	0.982	3.851	0.000	1.852	5.711
x3	-21.9691	4.200	-5.230	0.000	-30.222	-13.717
x4	6.3639	0.399	15.968	0.000	5.581	7.147
x5	-0.0496	0.014	-3.471	0.001	-0.078	-0.021
x6	-1.4352	0.223	-6.429	0.000	-1.874	-0.997
x7	0.1552	0.070	2.228	0.026	0.018	0.292
x8	-0.0129	0.004	-3.326	0.001	-0.021	-0.005
x9	-0.9926	0.149	-6.654	0.000	-1.286	-0.700
const	25.0582	5.392	4.647	0.000	14.463	35.653
=====						

- INTRODUCCIÓN
- GESTIÓN INTEGRAL DE MODELOS
- DIVERSIDAD DE MODELOS ESTADÍSTICOS
- **GESTIÓN INTEGRAL DE MODELOS CON R**
- CONCLUSIONES FINALES

GESTIÓN INTEGRAL DE MODELOS CON R

- ¿Qué elementos resultan claves cuando se mide la actuación de un modelo en términos estadísticos?
 - Comportamiento genérico en una muestra retardadas en el tiempo
 - Análisis univariante de las variables de un modelo retardadas en el tiempo
 - Análisis actual de las características y medida de la desviación de éstas respecto a las hipótesis iniciales
- Así pues un ejemplo de gestión estadística de modelos sencilla que actualmente se está aplicando surge cuando se crean sistemas para evaluar la “performance” de los modelos de Credit Scoring

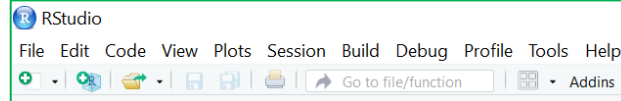
GESTIÓN INTEGRAL DE MODELOS CON R

- Gestión Estadística de un Credit Scoring con R
 - Elementos básicos de las componentes de la aplicación Shiny

reticulate

Base de
datos
Oracle

Código .py de
extracción de
datos

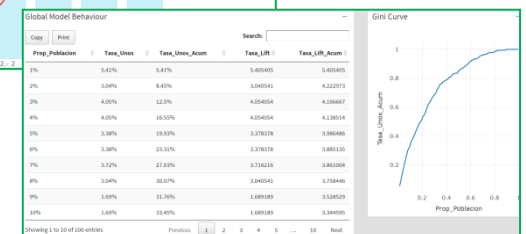
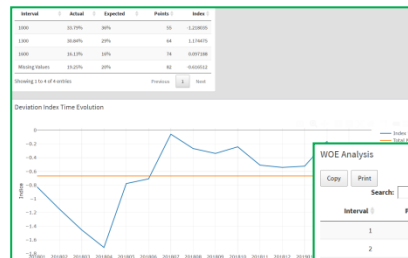


- Demo de Aplicación

CS Management

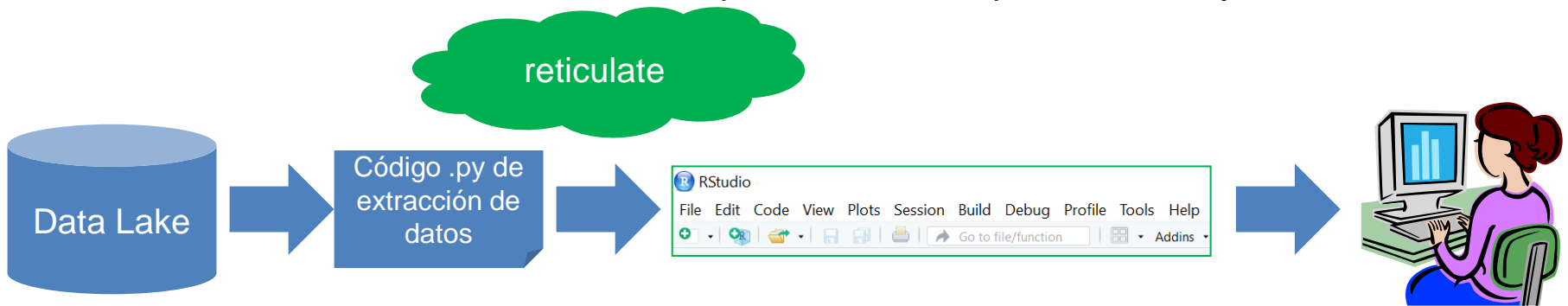
Equifax Known-Customer Model

- » Characteristic Analysis
- » WOE by Variable
- » Gini Behaviour



GESTIÓN INTEGRAL DE MODELOS CON R

- Gestión Económica de un Credit Scoring con R
 - Elementos básicos de las componentes de la aplicación Shiny



- Demo de Aplicación

Resumen de Datos Globales

Choose Initial Data: 2019-05-11

Choose Final Data: 2019-05-15

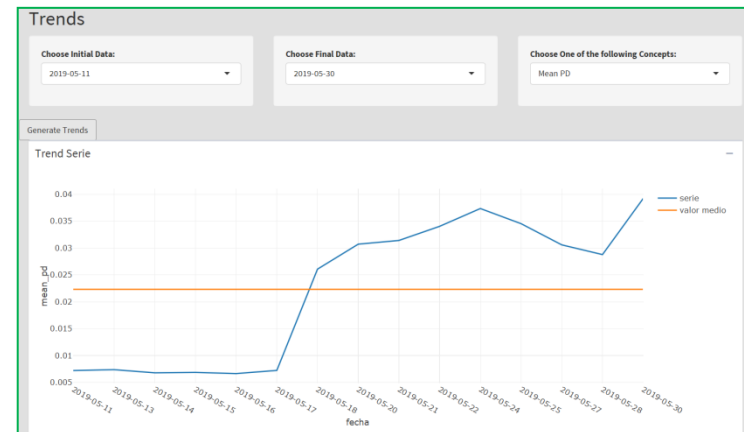
Generate Table

Global Data Resume

Copy Print

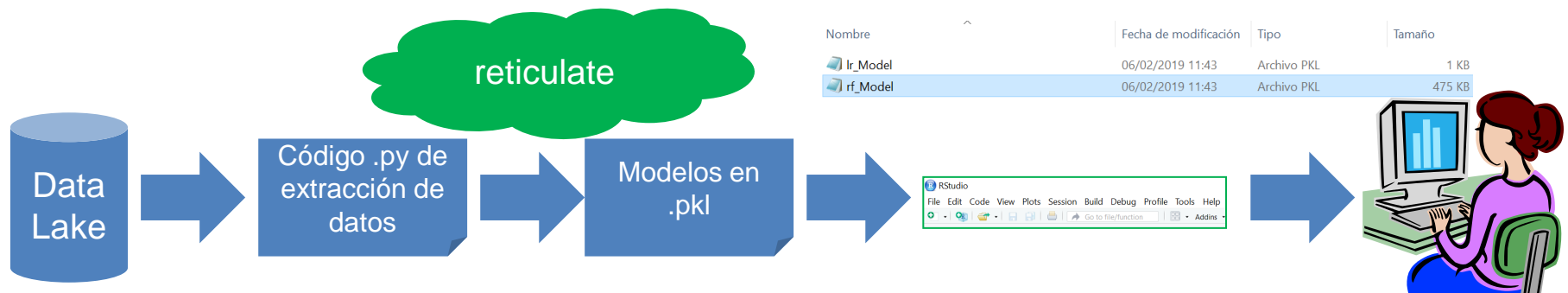
Variables	Value
Contratos	6192
Clientes	6048
Límite Pto Aceptado	14186039
Blengo Aceptado	2489028
Ingreso Esperado	96372
Mensualidad Media	37.28
PD pond a 6m	0.7101%
Pérdida Esperada a 6m	17674
Límite por Cliente	2346
Blengo por Cliente	412

Showing 1 to 10 of 12 entries

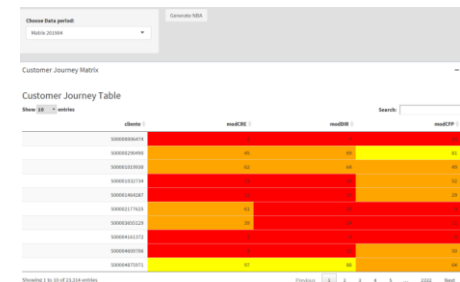
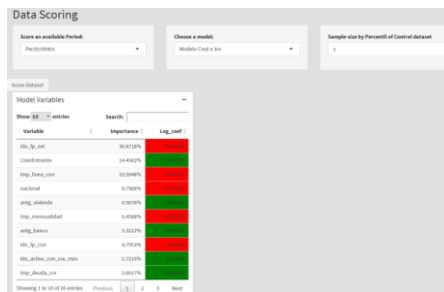


GESTIÓN INTEGRAL DE MODELOS CON R

- Funcionamiento de la librería reticulate: Ejecución con R de modelos fabricados en Python
 - Con reticulate se puede llamar objetos serializados con python directamente



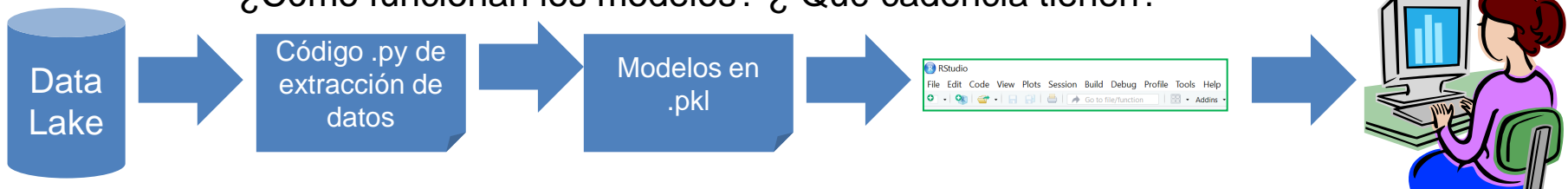
- Demo aplicación:



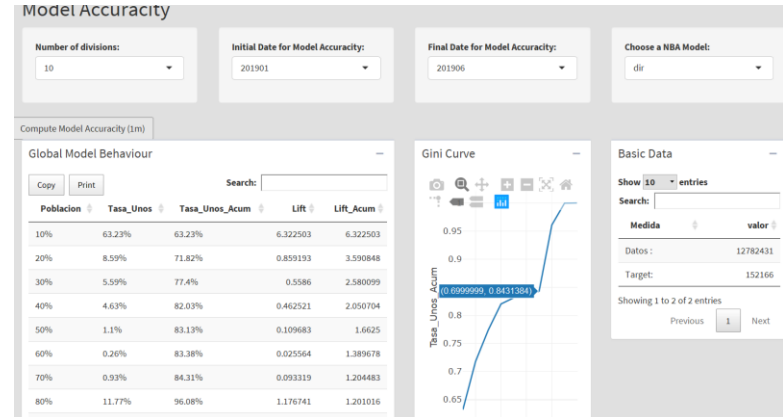
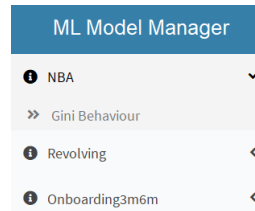
GESTIÓN INTEGRAL DE MODELOS CON R

- Gestión Masiva de Modelos desde un Data-Lake
 - Se tienen 9 modelos NBA, 13 Modelos de Fuga y 3 Modelos de Propensión mensuales puestos en producción y gestionados

- ¿Cómo funcionan los modelos? ¿Qué cadencia tienen?



- Demo aplicación:



- INTRODUCCIÓN
- GESTIÓN INTEGRAL DE MODELOS
- DIVERSIDAD DE MODELOS ESTADÍSTICOS
- GESTIÓN INTEGRAL DE MODELOS CON R
- **CONCLUSIONES FINALES**

CONCLUSIONES FINALES

- Conforme los datos van creciendo, la necesidad de su explotación mediante modelos estadísticos es cada vez mayor, por lo que surge una multiplicidad de nuevos modelos
- En muy corto período de tiempo, áreas de Analytics pueden tener que controlar una gran cantidad de modelos
- Los modelos pueden ser de diversos tipos y se pueden realizar en distintas lenguajes, siendo actualmente los más comunes python y R
- Se necesita de un framework analítico capaz de aglutinar los distintos modelos que se realizan por las áreas DS tanto si están como si no están puestos en producción

CONCLUSIONES FINALES

- R ofrece una alternativa eficaz para unificar distintos lenguajes y permitir que distintos grupos de DS trabajen al unísono, cada uno especializado en su labor al unificar fácilmente sus trabajos
- La librería reticulate ofrece todo un conjunto de posibilidades que permite la explotación de programas .py y de objetos serializados
- La combinación de reticulate con otras librerías como shiny ofrece alta productivización, presentación y ejecución de resultados finales
- Aunque aquí no se ha comentado, R permite conexiones tanto a los modernos Data Lake de cloudera, como a datos en la nube AWS y GCP, permite conexión a prácticamente todas las bases de datos propietarias y libres, es capaz de generar modelos en spark y está perfectamente integrado con h2o