

# Creación de un Servicio RESTful para el Modelo de Crédito con Aprendizaje Automático, en R

Maria Alejandra Quevedo Vera

Maestra en Ciencias de la Información y las Comunicaciones  
Inteligencia Computacional

7 Abril de 2018

- 1 Artículo 1
  - Resumen
  - Qué es REST?
- 2 Base de Datos
  - Descripción
- 3 Machine Learning
  - Modelo de Aprendizaje
- 4 Servicio REST
  - Comunicación JSON y R
  - API con R
  - Otros Experimentos
- 5 Peticiones desde otros Ambientes

- 1 Artículo 1
  - Resumen
  - Qué es REST?
- 2 Base de Datos
  - Descripción
- 3 Machine Learning
  - Modelo de Aprendizaje
- 4 Servicio REST
  - Comunicación JSON y R
  - API con R
  - Otros Experimentos
- 5 Peticiones desde otros Ambientes

En este documento se ilustrará paso a paso el proceso que se siguió para obtener un servicio RESTful utilizando RStudio. Para esto, se utilizó una base de datos con información crediticia de algunos ciudadanos alemanes, relevante para las entidades bancarias en el momento de realizar un estudio de crédito, tras lo cual se crea un árbol de predicción y se accede a esta información a través de la WEB utilizando POSTMAN.

## 1 Artículo 1

- Resumen
- Qué es REST?

## 2 Base de Datos

- Descripción

## 3 Machine Learning

- Modelo de Aprendizaje

## 4 Servicio REST

- Comunicación JSON y R
- API con R
- Otros Experimentos

## 5 Peticiones desde otros Ambientes

# Representational State Transfer

Es una interfaz que utiliza HTTP para obtener datos o generar operaciones sobre los mismos.

El concepto más importante es que es **SIN ESTADO** lo cuál significa que no almacena datos. Un ejemplo es el sistema World Wide Web.

Solicitudes de HTTP: CREAR, ACTUALIZAR, LEER, ELIMINAR.

- 1 Artículo 1
  - Resumen
  - Qué es REST?
- 2 Base de Datos
  - Descripción
- 3 Machine Learning
  - Modelo de Aprendizaje
- 4 Servicio REST
  - Comunicación JSON y R
  - API con R
  - Otros Experimentos
- 5 Peticiones desde otros Ambientes

# Descripción

La base de datos se tomó del Repositorio de la Universidad Irvine de California (UCI): Statlog (German Credit Data) Data Set. Características [5]:

- 1000 instancias
- 20 atributos: 7 numéricos, 13 cualitativos.

	Status.of.existing.checking.account	Duration.in.month	Credit.history	Purpose	Credit.amount	Savings.account.bonds
1	A11	6	A34	A43	1169	A65
2	A12	48	A32	A43	5951	A61
3	A14	12	A34	A46	2096	A61
4	A11	42	A32	A42	7882	A61
5	A11	24	A33	A40	4870	A61
6	A14	36	A32	A46	9055	A65
7	A14	24	A32	A42	2835	A63
8	A12	36	A32	A41	6948	A61
9	A14	12	A32	A43	3059	A64
10	A12	30	A34	A40	5234	A61
11	A12	12	A32	A40	1295	A61
12	A11	48	A32	A49	4308	A61
13	A12	12	A32	A43	1567	A61
14	A11	24	A34	A40	1199	A61
15	A11	12	A32	A40	1403	A61

Figura: Base de Datos tomada de la URL y cargada en R



- 1 Artículo 1
  - Resumen
  - Qué es REST?
- 2 Base de Datos
  - Descripción
- 3 Machine Learning
  - Modelo de Aprendizaje
- 4 Servicio REST
  - Comunicación JSON y R
  - API con R
  - Otros Experimentos
- 5 Peticiones desde otros Ambientes

# Modelo de Aprendizaje

Consiste en tomar la base de datos y crear un árbol de decisión como el de la Figura 2 para saber el resultado del estudio crediticio [3].

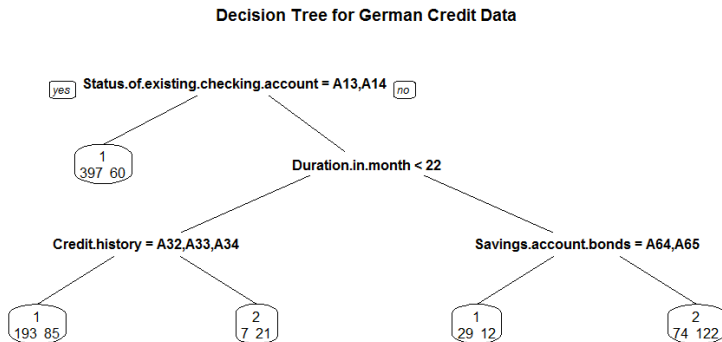


Figura: Árbol de decisión del Sistema

- 1 Artículo 1
  - Resumen
  - Qué es REST?
- 2 Base de Datos
  - Descripción
- 3 Machine Learning
  - Modelo de Aprendizaje
- 4 Servicio REST
  - Comunicación JSON y R
  - API con R
  - Otros Experimentos
- 5 Peticiones desde otros Ambientes

Utilizando la librería *jsonlite* se habilita el paso de datos entre JSON y R.

```
install.packages("rpart.plot")
library(rpart.plot)
# Visualización del árbol
# 1 es bueno, 2 es malo
prp(
  decision.tree
  , extra=1
  , varlen=0
  , faclen=0
  , main="Árbol de Decisión para Datos Alemanes de Crédito"
)
new.data <- list(
  Status.of.existing.checking.account='A11'
  , Duration.in.month=20
  , Credit.history='A32'
  , Savings.account.bonds='A65'
)
predict(decision.tree, new.data) #save(decision.tree, file='arbol.decision.RData')
```

Figura: Código de comunicación JSON y R

- 1 Artículo 1
  - Resumen
  - Qué es REST?
- 2 Base de Datos
  - Descripción
- 3 Machine Learning
  - Modelo de Aprendizaje
- 4 Servicio REST
  - Comunicación JSON y R
  - API con R
  - Otros Experimentos
- 5 Peticiones desde otros Ambientes

En este caso se utilizó la librería *plumber*

```
library(plumber)
r <- plumb("modelo_de_credito.R")
r$run(port=8000)
```

Figura: Implementación con plumber

En este caso, se enviará y recibirá información por el puerto 8000.

Se utiliza la Terminal para probar que el sistema responde.

```
curl -X POST -d '{"Status.of.existing.checking.account": "A11", "Duration.in.month": 24, "Credit.history": "A32",  
"Savings.account.bonds": "A63"}' -H 'Content-Type: application/json' localhost:8000/predict
```

**Figura:** Envío de requerimiento al sistema a través del puerto 8000.

```
MacBook-Air-de-Alejandra:~ alejandra$ curl -X POST -d '{"Status.of.existing.checking.account": "A12", "Duration.in.month": 20, "Credit.history": "A30", "Savings.account.bonds": "A62"}' -H 'Content-Type: application/json' localhost:8000/predict  
{"default.probability":0.75}MacBook-Air-de-Alejandra:~ alejandra$
```

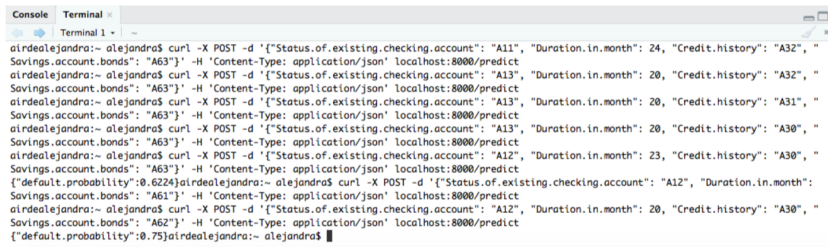
**Figura:** Respuesta del sistema.

- 1 Artículo 1
  - Resumen
  - Qué es REST?
- 2 Base de Datos
  - Descripción
- 3 Machine Learning
  - Modelo de Aprendizaje
- 4 Servicio REST
  - Comunicación JSON y R
  - API con R
  - Otros Experimentos
- 5 Peticiones desde otros Ambientes



# Otros Experimentos

Se realizan experimentos con otros valores para comprobar que el sistema está respondiendo de forma correcta.



```
airdealejandra:~ alejandra$ curl -X POST -d '{"Status.of.existing.checking.account": "A11", "Duration.in.month": 24, "Credit.history": "A32", "Savings.account.bonds": "A63"}' -H 'Content-Type: application/json' localhost:8000/predict
airdealejandra:~ alejandra$ curl -X POST -d '{"Status.of.existing.checking.account": "A13", "Duration.in.month": 20, "Credit.history": "A32", "Savings.account.bonds": "A63"}' -H 'Content-Type: application/json' localhost:8000/predict
airdealejandra:~ alejandra$ curl -X POST -d '{"Status.of.existing.checking.account": "A13", "Duration.in.month": 20, "Credit.history": "A31", "Savings.account.bonds": "A63"}' -H 'Content-Type: application/json' localhost:8000/predict
airdealejandra:~ alejandra$ curl -X POST -d '{"Status.of.existing.checking.account": "A13", "Duration.in.month": 20, "Credit.history": "A30", "Savings.account.bonds": "A63"}' -H 'Content-Type: application/json' localhost:8000/predict
airdealejandra:~ alejandra$ curl -X POST -d '{"Status.of.existing.checking.account": "A12", "Duration.in.month": 23, "Credit.history": "A30", "Savings.account.bonds": "A63"}' -H 'Content-Type: application/json' localhost:8000/predict
airdealejandra:~ alejandra$ curl -X POST -d '{"Status.of.existing.checking.account": "A12", "Duration.in.month": 20, "Credit.history": "A30", "Savings.account.bonds": "A61"}' -H 'Content-Type: application/json' localhost:8000/predict
airdealejandra:~ alejandra$ curl -X POST -d '{"Status.of.existing.checking.account": "A12", "Duration.in.month": 20, "Credit.history": "A30", "Savings.account.bonds": "A62"}' -H 'Content-Type: application/json' localhost:8000/predict
airdealejandra:~ alejandra$
```

Figura: Experimentos con otros valores.

# Otros Experimentos

Petición desde POSTMAN [4] para comprobar que es posible obtener el servicio de respuesta al modelo, desde un programa diferente a R y desde otro lenguaje de programación.

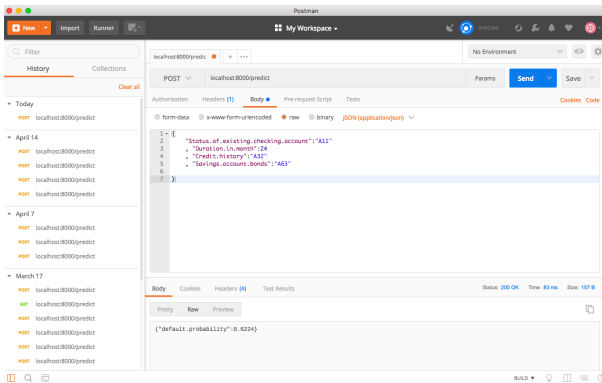


Figura: Petición desde POSTMAN.

# REsultados y Conclusiones

- Este trabajo permitió comprobar el concepto e servicio RESTful utilizando un programa sencillo desde R y con el cual fue posible no sólo encontrar una aplicación útil, sino que también fue posible evidenciar la interoperabilidad entre ambientes de trabajo y lenguajes de programación.
- Para este caso no se tomaron en cuenta algunos aspectos como mensajes de alerta y manejo de excepciones, lo que podría significar trastornos y confusiones a la hora de utilizarlo.
- POSTMAN es una herramienta útil para la verificación de APIs, enviando peticiones y en donde también es posible visualizar las respuestas.



Terry M. Therneau

An Introduction to Recursive Partitioning Using the RPART Routines.  
2018



A. Ohri

R for Cloud Computing.  
2018



Knowru

How to create a RESTful API for a machine learning credit model in R.  
2018.

<https://www.knowru.com/blog/how-create-restful-api-for-machine-learning-credit-model-in-r/>



## POSTMAN

Download POSTMAN.

*2018.*

<https://www.getpostman.com>



## UCI Machine Learning

Download POSTMAN.

*2018*

<https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/german.data>



## DR. M. ELKSTEIN.

Learn REST: A Tutorial

*2018*

<https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/german.data>