RETOS EN FINTECH
con R + h2o.ai

Nov 25 18.30

via zoom - meetup.com/R-en-Rosario

Por Camila Burne

Mundo Fintech

Finanzas + Tecnología = Fintech

Criptomonedas, BlockChain, Crowdfunding, Pagos y transferencias, Finanzas personales, Mercados financieros, Originación de Crédito, Servicios de gestión (de fraude, perfil de riesgo), Soluciones de software (ej contable), Insur-tech (aseguradoras), Bancos, Tarjetas de Débito y Crédito.

+INFO

Stori - Tarjeta de Crédito 4+

Sin Anualidad, ni historial

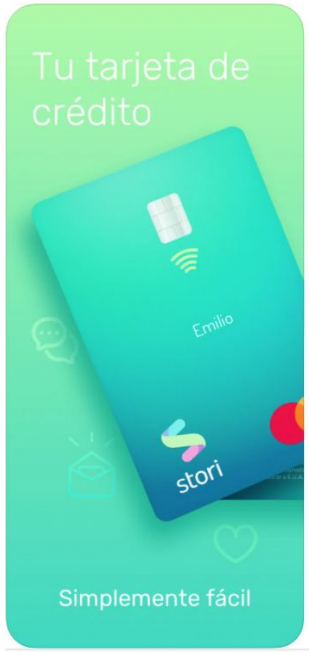
Powerup Technology Inc.

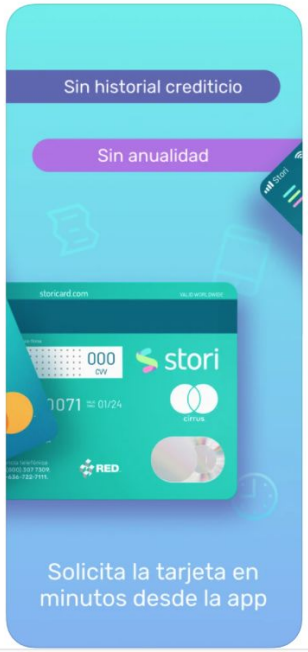
#31 en Finanzas

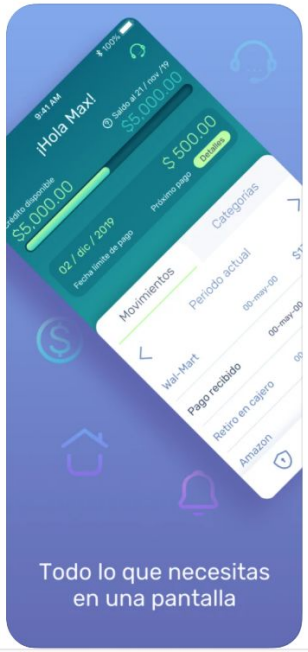
★★★★★ 4.8 • 4.4 k valoraciones

Gratis

Capturas de pantalla del iPhone







KueskiUp 4+

Kueski

#158 en Finanzas

★★★★★ 5.0 • 6 valoraciones

Gratis

Ver en Mac App Store ↗

Broxel 17+

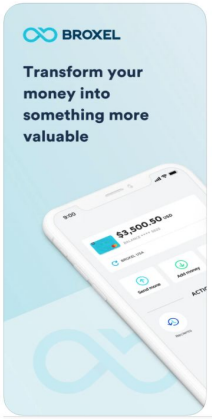
Broxel

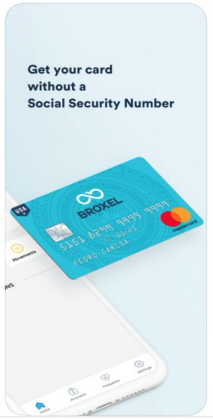
★★★★★ 3.9 • 11 Ratings

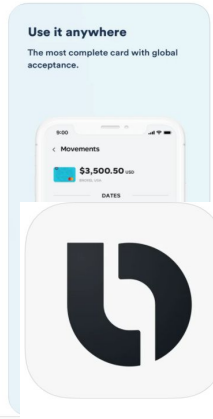
Free

View in Mac App Store ↗

iPhone Screenshots







s de pantalla del iPhone

Bitso - Buy and sell bitcoin 4+

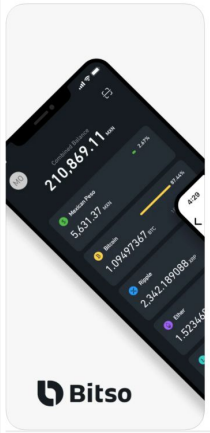
Evolve your money

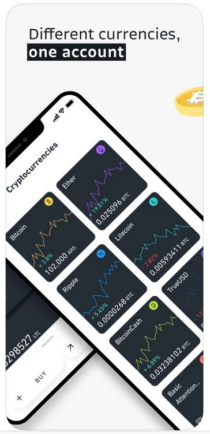
Bitso SAPI de CV

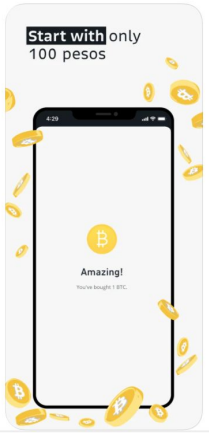
★★★★★ 4.3 • 25 Ratings

Free

iPhone Screenshots





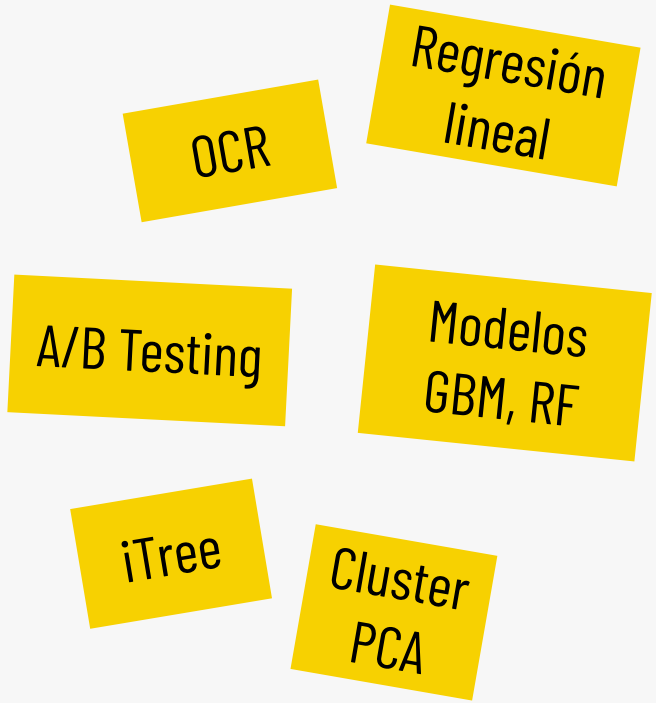


Retos

- Validación de identidad (1st & 3rd party fraud)
- Perfil de riesgo
- Asignación de producto
- Identificación de tranx sospechas
- Perfil de compra
- Gestión de cobranza
- Growth

Retos

- Validación de identidad (1st & 3rd party fraud)
- Perfil de riesgo
- Asignación de producto
- **Identificación de tranx sospechas**
- **Perfil de compra**
- Gestión de cobranza
- Growth



h2o.ai

h2o es re versátil

x1: para python y para R funciona igual

x2: para research y para implementación (modelos en prod en aws 😬)

x3: métodos clásicos (PCA, 1901,Karl Pearson) y métodos nuevos (iTree, 2008/2012, Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou)

Tranx sospechas

Cada día recibimos como mínimo 5000 transacciones de nuestros clientes, de las cuales no sabemos si son reales o fraude.

Como no tenemos identificadas las fraudulentas, tenemos que aislar las "raras" e investigarlas. El objetivo final es definir
Y = 1 si la transacción es fraudulenta.

Hay dos enfoques para detectar anomalías:

1. Definir un comportamiento normal
2. Buscar outliers → *isolation tree*

Isolation Forest

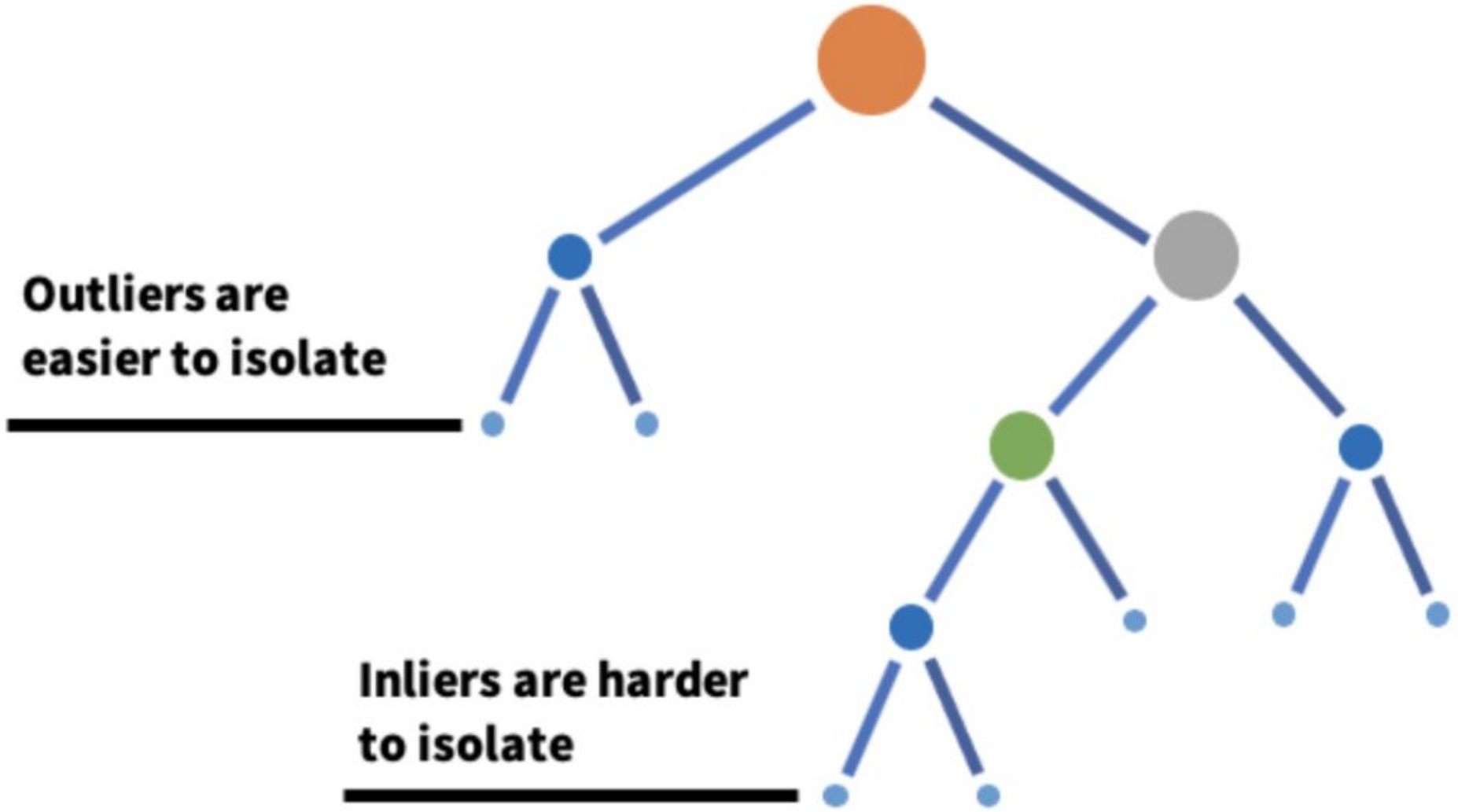
Introduction

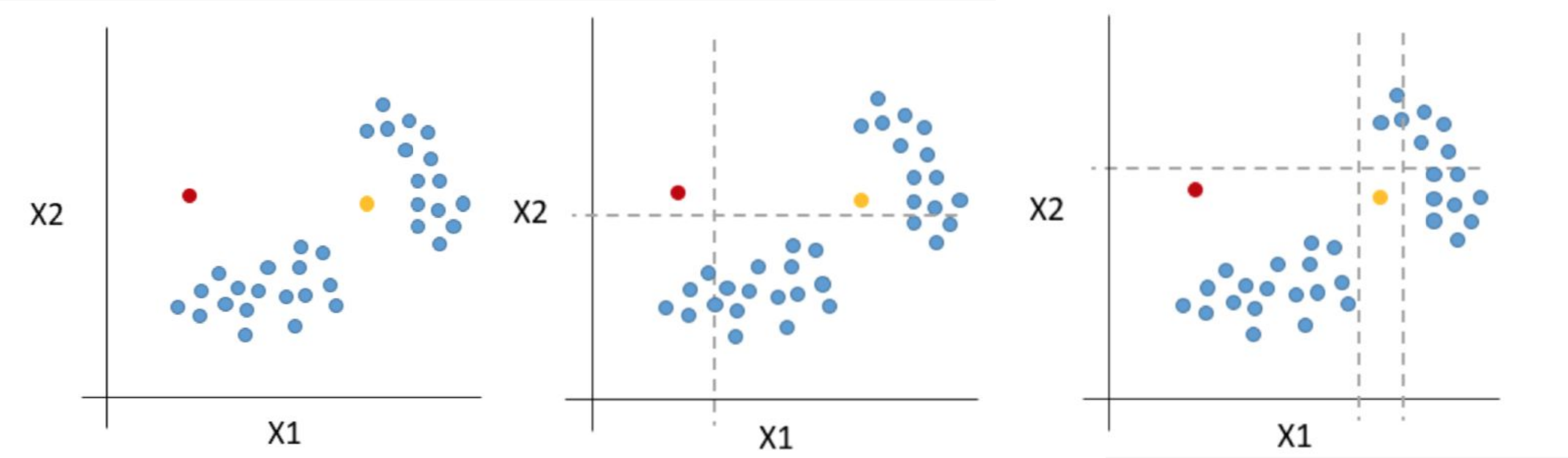
Isolation Forest is similar in principle to Random Forest and is built on the basis of decision trees. Isolation Forest, however, identifies anomalies or outliers rather than profiling normal data points. Isolation Forest isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of that selected feature. This split depends on how long it takes to separate the points.

Random partitioning produces noticeably shorter paths for anomalies. When a forest of random trees collectively produces shorter path lengths for particular samples, they are highly likely to be anomalies.

Tutorials and Blogs

The following tutorials are available that describe how to use Isolation Forest to find anomalies in a dataset and how to interpret the results.


Outliers are
easier to isolate
Inliers are harder
to isolate

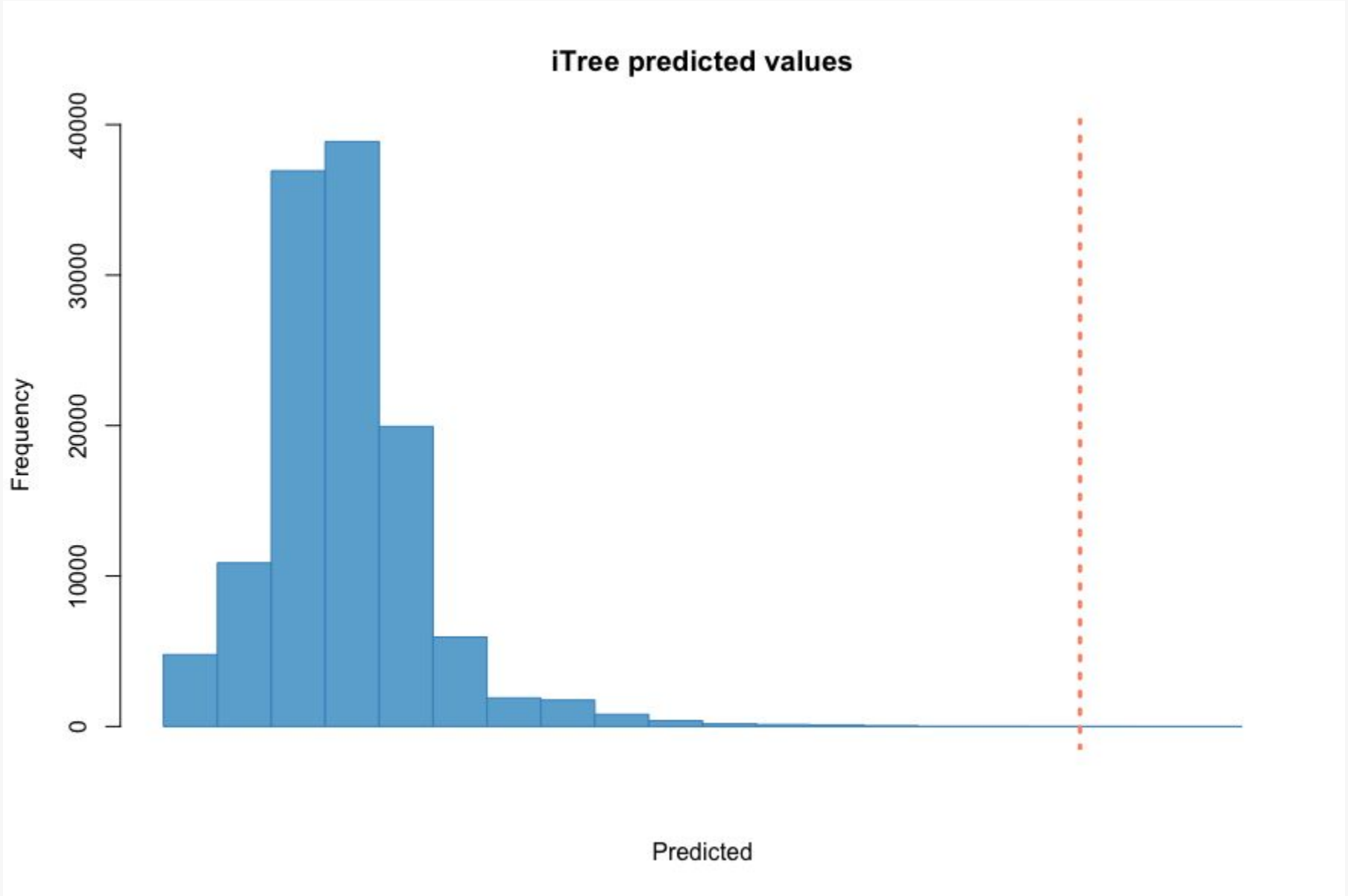

X2
X1
X2
X1
X2
X1

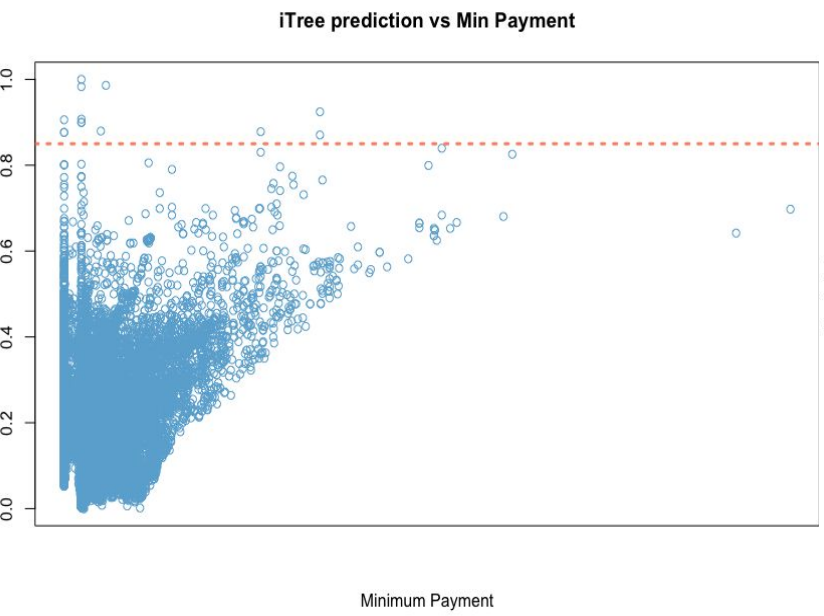
Tranx sospechas

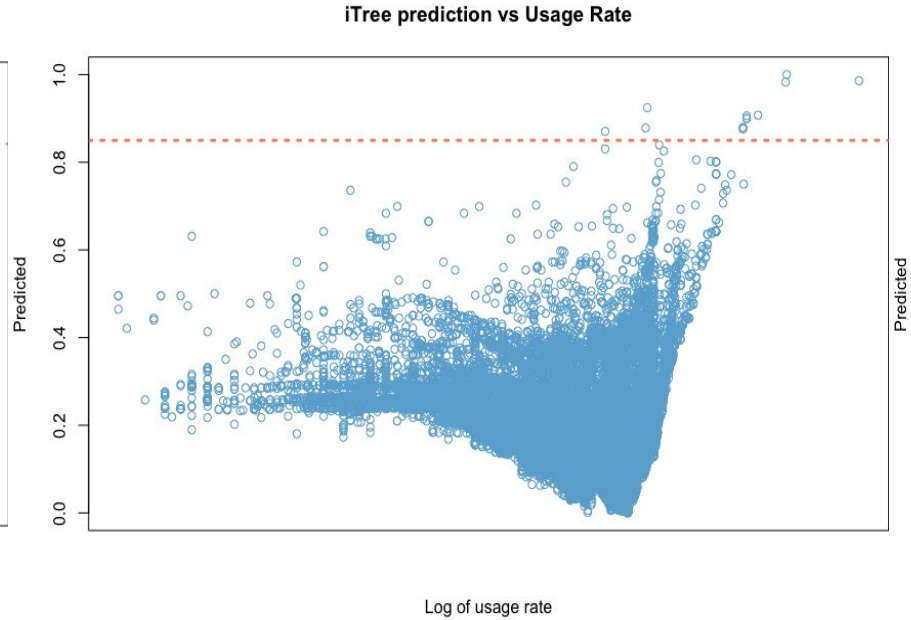
- **Credit limit**
- **Invoice amount**
- **Usage Rate**
- **$100 Min payment**

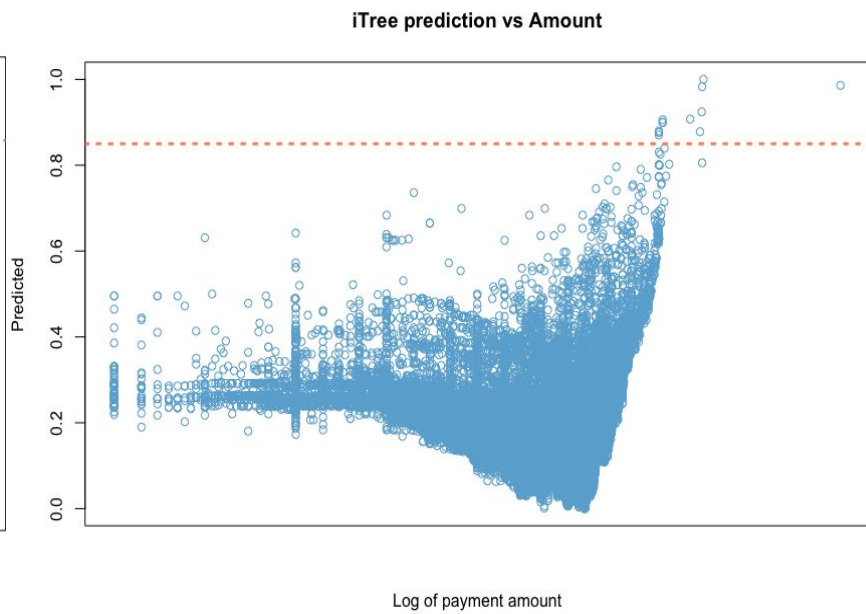
	credit_limit	invoice_amt	usage_rate	min_pmt
236233	1000	100.00	-10.00000000	100.00
236234	3000	100.00	-3.33333333	223.99
236235	1000	800.00	-80.00000000	100.00
236236	1000	100.00	-10.00000000	717.88
236237	1000	350.00	-35.00000000	100.00
236238	2000	624.29	-31.21450000	100.00
236239	1000	634.00	-63.40000000	100.00
236240	1000	470.00	-47.00000000	629.14
236241	3000	389.50	-12.98333333	739.50
236242	1000	400.00	-40.00000000	272.62
236243	8000	500.00	-6.25000000	435.55
236244	1000	265.00	-26.50000000	264.87
236245	1000	253.00	-25.30000000	100.00
236246	2000	115.00	-5.75000000	112.58
236247	2500	95.00	-3.80000000	189.39
236248	3000	1031.13	-34.37100000	100.00
236249	5000	1047.00	-20.94000000	100.00

```r
library(h2o)
df  <- read.csv("payments_itree.csv")

# Use h2o object
h2o.init()
df  <- as.h2o(pay[,c("credit_limit","invoice_amt","usage_rate","min_pmt")])

# Build Isolation Tree
it  <- h2o.isolationForest(training_frame = df,
                           sample_rate = 0.1,
                           max_depth = 20,
                           ntrees = 100)

# Keep prediction and mean length and add to original df
pred   <- h2o.predict(it, df)
pay_df <- cbind(pay,as.data.frame(pred))

# Distribution of mean length and predicted values:
hist(pay_df$mean_length)
hist(pay_df$predict)
```


iTree predicted values
Frequency
0
10000
20000
30000
40000
Predicted


iTree prediction vs Min Payment
Predicted
0.0
0.2
0.4
0.6
0.8
1.0
Minimum Payment


iTree prediction vs Usage Rate
Predicted
0.0
0.2
0.4
0.6
0.8
1.0
Log of usage rate


iTree prediction vs Amount
Predicted
0.0
0.2
0.4
0.6
0.8
1.0
Log of payment amount

Tranx sospechas: *acciones*

iTree

En vez de establecer 4 puntos de corte para 4 variables, usé un Isolation Tree con un punto de corte en sus predichos.

El área de fraude tiene un número menor de cuentas a revisar, y yo tengo un método sistemático para encontrar cuentas con transacciones sospechosas.

$100

Perfil de compra

De las 5000 transacciones que recibimos a diario o 650,000 en total, es difícil resumir "en qué gastan nuestros clientes".

Para reducir tanta info h2o tiene métodos clásicos como análisis de componentes principales que son el punto de partida para entender comportamientos.

Las categorías de compras son muchísimas: restaurantes, fast food, tienditas, retail, ecommerce... y se podrían hasta desagregar por marcas.

Principal Component Analysis (PCA)

Introduction

Principal Components Analysis (PCA) is closely related to Principal Components Regression. The algorithm is carried out on a set of possibly collinear features and performs a transformation to produce a new set of uncorrelated features.

PCA is commonly used to model without regularization or perform dimensionality reduction. It can also be useful to carry out as a preprocessing step before distance-based algorithms such as K-Means since PCA guarantees that all dimensions of a manifold are orthogonal.

Defining a PCA Model

- model_id: (Optional) Specify a custom name for the model to use as a reference. By default, H2O automatically generates a destination key.
- training_frame: (Required) Specify the dataset used to build the model. **NOTE**: In Flow, if you click the **Build a model** button from the `Parse` cell, the training frame is entered automatically.
- validation_frame: (Optional) Specify the dataset to calculate validation metrics.
- x: Specify a vector containing the names or indices of the predictor variables to use when building the model. If `x` is missing, then all columns are used.

Perfil de compra

amt.fastfood	cnt.fastfood	amt.foodstores	cnt.foodstores	amt.gas	cnt.gas	amt.grocerystore	cnt.grocerystore	amt.miscellaneou
0.00	0	0.00	0	0.00	0	0.00	0	0.
978.00	6	0.00	0	664.94	1	0.00	0	3594.
617.00	4	0.00	0	0.00	0	0.00	0	660.
356.50	1	4561.34	4	1244.11	1	0.00	0	2670.
1703.50	9	370.00	1	0.00	0	0.00	0	593.
200.00	1	0.00	0	0.00	0	0.00	0	2518.
0.00	0	0.00	0	0.00	0	0.00	0	2767.
837.00	5	1535.15	3	985.70	4	0.00	0	1576.
0.00	0	0.00	0	0.00	0	0.00	0	3106.
0.00	0	0.00	0	0.00	0	0.00	0	40.
217.98	1	156.00	1	0.00	0	0.00	0	3195.
0.00	0	0.00	0	0.00	0	0.00	0	0.
1076.10	7	0.00	0	0.00	0	0.00	0	1556.
1563.86	4	0.00	0	0.00	0	0.00	0	0.
0.00	0	0.00	0	0.00	0	0.00	0	249.
853.60	4	0.00	0	0.00	0	0.00	0	0.
638.00	1	512.25	1	0.00	0	0.00	0	0.
0.00	0	0.00	0	0.00	0	0.00	0	0.
1647.65	9	0.00	0	0.00	0	0.00	0	0.
387.50	3	0.00	0	0.00	0	0.00	0	4648.

```r
# PCA with h20 ####
library(h2o)
h2o.init()
h_df <- as.h2o(x = df[, pca_vars])

# Build and train the model:
pca <- h2o.prcomp(training_frame = h_df, k = 2, use_all_factor_levels = TRUE,
                  pca_method = "GLRM", transform = "Descale", impute_missing = FALSE)
# PC Importance
pca@model$importance

# Eigenvectors
eigen <- as.data.frame(pca@model$eigenvectors)

```

170:46 PCA with h20

Console Terminal

~/tx_r/

```
> eigen <- as.data.frame(pca@model$eigenvectors)
> pca@model$importance
Importance of components:
                          pc1      pc2
Standard deviation     8.221224 2.389000
Proportion of Variance 0.533428 0.045044
Cumulative Proportion  0.533428 0.578472
> pca@model$eigenvectors
Rotation:
                      pc1       pc2
statement_num    -0.332801 -0.207840
original_bal     -0.126139  0.032610
min_pmt          -0.080992 -0.146054
purchase_amt     -0.021970  0.165898
purchase_tnx_cnt -0.086152  0.319513

---
                      pc1       pc2
cnt.transportati -0.028424  0.117232
amt.utilityservi -0.020206  0.081714
cnt.utilityservi -0.023749  0.079208
b_state          -0.153523 -0.036781
j_ind            -0.239223  0.004317
age              -0.487255 -0.160744
```

PC 1	
Nivel/Antiguedad de clientes	
age	higher PC, older
resident_type	higher PC, serius resident (rent, owner)
education	higher PC, higher education
statement_num	higher PC, older stmt
j_ind	higher PC, better earning job by cat
income_month	higher PC, higher income (quant)
b_state	higher PC, more populous / modern state
credit_limit	higher PC, higher credit limit
original_bal	higher PC, higher original_bal
cur_balance	higher PC, higher cur_balance
addr_name_match	higher PC, more POA to their name
num_dependent	higher PC, more dependents
marital_status	higher PC, older in marital status

PC 2	
Gasto en compras corto plazo, necesidades	
purchase_amt_summar	more purchase, higher
purchase_tnx_cnt	more purchase, higher
cnt.fastfood	more purchase, higher
cnt.restaurants	more purchase, higher
cnt.miscellaneou	more purchase, higher
amt.miscellaneou	more purchase, higher
statement_num	early statment, higher
age	younger age, higer
min_pmt	less payment, higher

PC 3	
Gasto en compras largo plazo, lujo	
interest_summary	more interest, higher
avg_balance	more spent, higher
original_bal	more spent on previous, higher
cur_balance	more purchase, higher
credit_limit	higher limit, higher
min_pmt	higher min payment, higher
amt.entertainment	more purchase, higher
amt.other	more purchase, higher
amt.retailoutlet	more purchase, higher
amt.transportati	more purchase, higher
age	younger, higher
purchase_tnx_cnt	less transactions, higher
j_ind	worse job by cat, higher
resident_type	worse resident type, higher
education	worse education, higher

Perfil de compra: *acciones*

Mi población que se define más fácil son los viejos, con mejores sueldos, más educados, dueños de su casa, en estados más urbanos... sacando esta población, el resto se divide mejor por qué compra: comida, restaurantes y tienditas (menor gasto) vs (mayor gasto) otras tiendas, retailers, entretenimiento y transporte.

Esto me sirve para encarar un análisis de comportamiento con una idea más clara de qué categorías de compras se asocian mejor entre sí.

h2o es versátil pero consistente

Para PCA y para iTree, la sintaxis fue similar:

```
model_obj <- h2o.princomp(..) / h2o.isolationforest(..) / h2o.gbm(..)

model_obj@model$importance
model_obj@model$eigenvectors

pred <- h2o.predict(model_obj, newdata = ..)
perf <- h2o.performance(model_obj)

h2o.auc(perf)
```

Espero que h2o.ai les sirva

Gracias!