

Aprendizaje Automático con Tensorflow y R

Edgar Ruiz

 edgararui

 theotheredgar

 edgararui



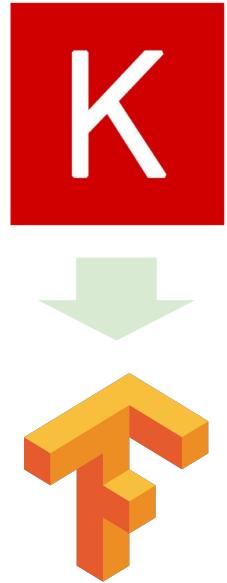
Tensorflow

- ❑ Procesa aprendizaje automático y profundo.
- ❑ Computación de alto desempeño
- ❑ Librería de código abierto
- ❑ Corre en una variedad de plataformas (CPUs, GPUs)
- ❑ Corre en una variedad de dispositivos (Desktops, servidores y teléfonos móvil)



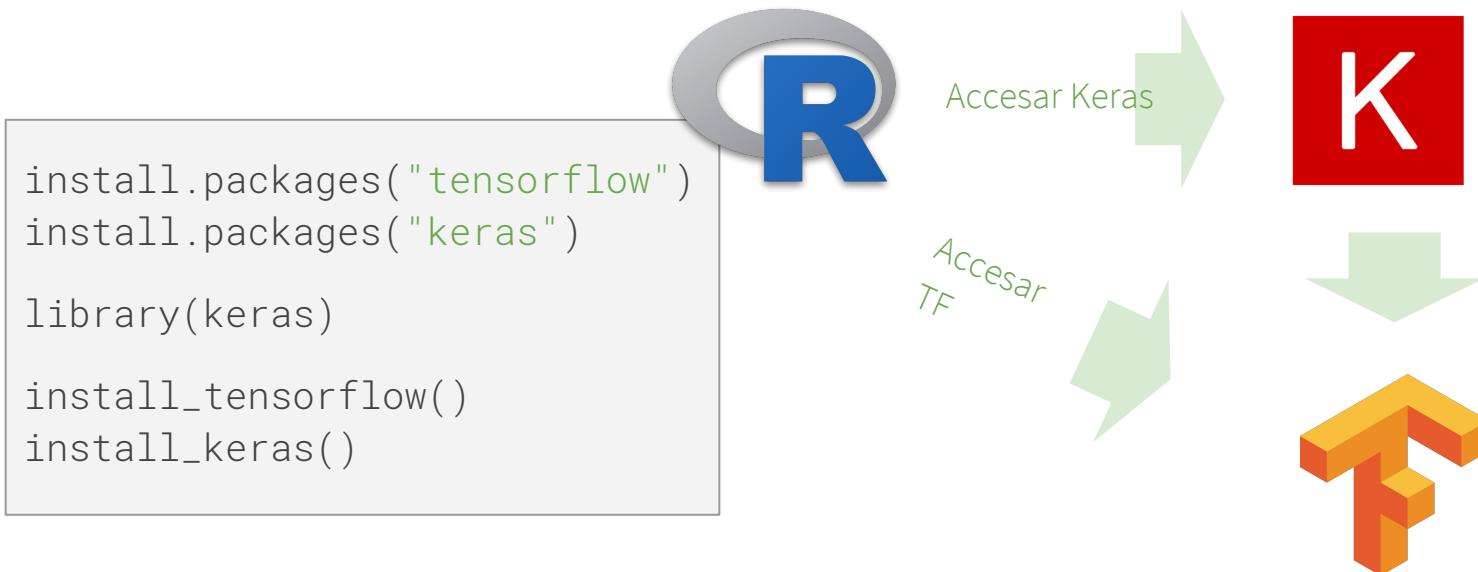
Keras

- ❑ API para redes neuronales
- ❑ El mismo código funciona en CPU or GPU
- ❑ Modelos de aprendizaje profundo fácilmente
- ❑ Redes convolucionales y recurrentes
- ❑ Procesa arquitecturas de redes arbitrarias
- ❑ Varios ambientes: Tensorflow, CNTK o Theano



Integrando con R

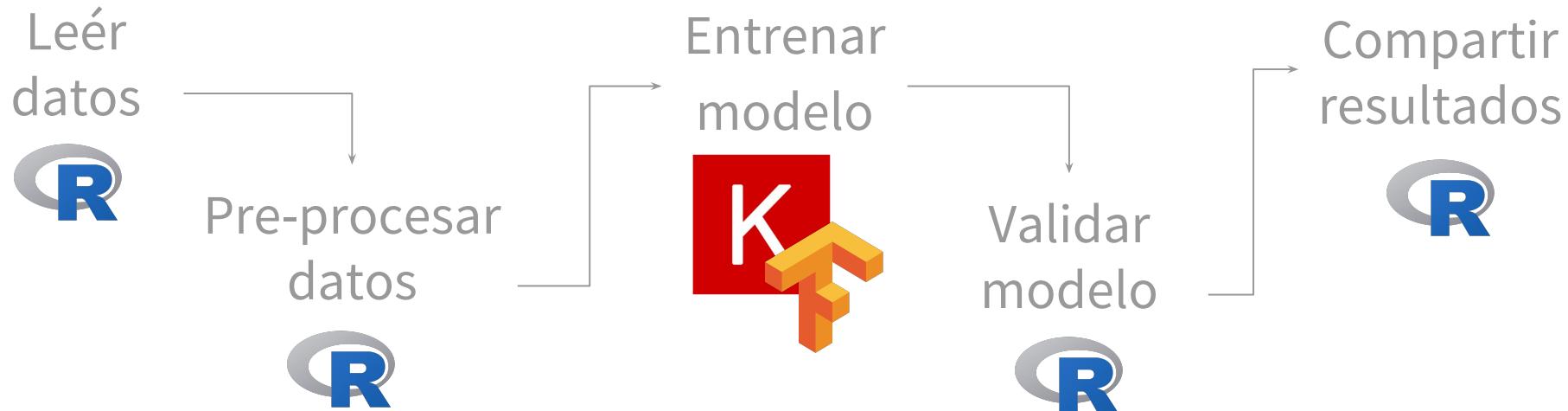
Las tres tecnologías se pueden integrar mediante paquetes de R



¿Por que Tensorflow, y también R?

Varios pasos conforman el Aprendizaje Automático (ML).

Todos los pasos pueden ser corridos con R



Ademas...

R tiene excelentes paquetes que se especializan en el desarrollo de proyectos de Aprendizaje Automático.

Leer
datos



Pre-procesar
datos



Entrenar
modelo



Validar
modelo



Compartir
resultados



Prediciendo pérdida de clientes usando Aprendizaje Profundo en Keras

Demostración

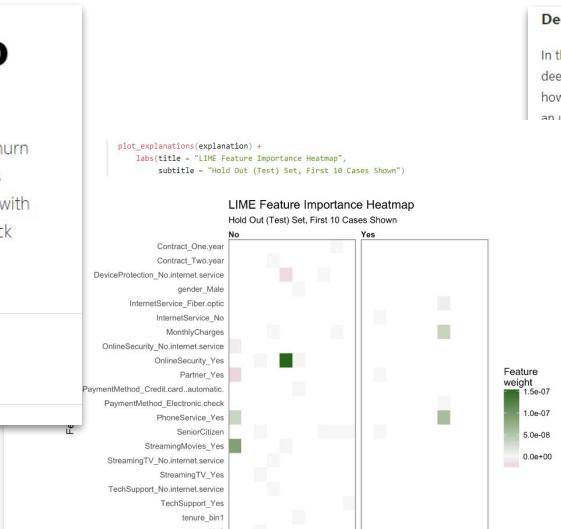
Análisis original

Ejemplo basado en el artículo: “Deep Learning With Keras To Predict Customer Churn”, publicado por Matt Dancho.

Deep Learning With Keras To Predict Customer Churn

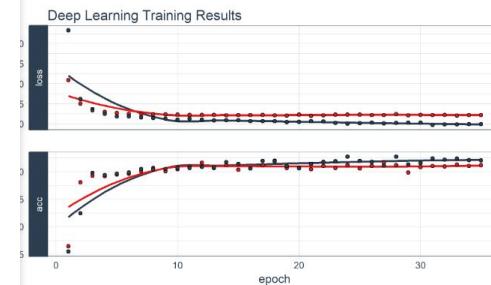
Using Keras to predict customer churn based on the IBM Watson Telco Customer Churn dataset. We also demonstrate using the lime package to help explain which features drive individual model predictions. In addition, we use three new packages to assist with Machine Learning: recipes for preprocessing, rsample for sampling data and yardstick for model metrics.

AUTHOR	AFFILIATION	PUBLISHED	CITATION
Matt Dancho	Business Science	Jan. 10, 2018	Dancho, 2018



Deep Learning With Keras (What We Did With The Data)

In this example we show you how to use keras to develop a sophisticated and highly accurate deep learning model in R. We walk you through the preprocessing steps, investing time into how to format the data for Keras. We inspect the various classification metrics, and show that an un-tuned ANN model can easily get 82% accuracy on the unseen data. Here's the deep learning history visualization.



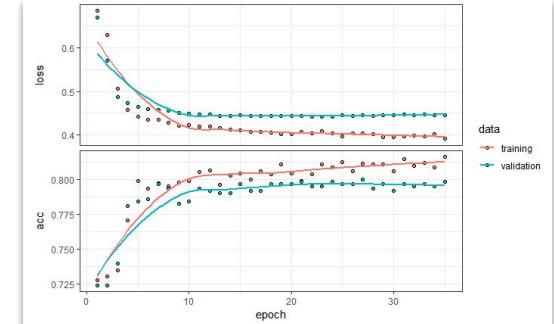
<https://blogs.rstudio.com/tensorflow/posts/2018-01-11-keras-customer-churn/>

Objetivo

Analizar datos de pérdida de clientes de una compañía de telecomunicaciones con R, Keras y Tensorflow.

WA_Fn-UseC_Telco-Customer-Churn - Excel

	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U
1	customerID	gender	SeniorCitizen	tenure	PhoneService	MultipleLines	InternetService	DraftSms	OnLineBank	Dependent	PhoneService	MultipleLines	InternetService	DraftSms	StreamingContent	StreamingMovies	PaperlessBilling	PaymentMethod	MonthlyChurn	TotalChurn
2	7500-VINICG Female	0	Yes	No	2	No	No	DSL	Yes	No	No	No	No	No	No	No	Electronic cl	29.85	29.85	
3	5575-ONV01M Male	0	No	No	34	Yes	No	DSL	Yes	Yes	No	No	No	No	One year	No	Mobile chec	56.95	1889.5 No	
4	3466-MLKJL Male	0	No	No	3	No	No	No	No	No	No	No	No	No	No	No	Mobile chec	56.95	1889.5 No	
5	7595-CFCVCMale	0	No	No	45	No	No	No	DSL	Yes	Yes	No	No	No	No	No	Bank trans	42.3	1840.75 No	
6	9237-HQXKZ Female	0	No	No	7	Yes	No	No	Fiber optic	No	No	No	No	No	No	No	Electronic cl	70.15	1840.75 Yes	
7	7300-ZKXC Female	0	No	No	2	Yes	No	No	Fiber optic	No	No	No	No	No	No	No	Bank trans	42.3	1840.75 Yes	
8	1452-KDVK Male	0	No	Yes	22	Yes	No	Fiber optic	No	Yes	No	No	No	No	No	No	Credit card	89.1	1849.4 No	
9	4731-MLKJL Female	0	No	No	19	Yes	No	No	No	No	No	No	No	No	No	No	Mobile chec	56.95	1889.5 No	
10	7893-POOHGU Female	0	Yes	No	29	Yes	No	No	Fiber optic	No	Yes	Yes	Yes	No	No	No	Electronic cl	104.8	3066.05 Yes	
11	6388-TABGU Male	0	No	Yes	62	Yes	No	DSL	Yes	Yes	No	No	No	One year	No	Bank trans	56.15	3487.95 No		
12	1393-MLKJL Male	0	No	No	11	Yes	No	No	No	No	No	No	No	No	No	No	Mobile chec	56.95	1889.5 No	
13	7469-LXKCI Male	0	No	No	15	Yes	No	No	No	No	No	No	No	No	No	No	Credit card	18.95	326.8 No	
14	1801-TIVAK Male	0	Yes	No	58	Yes	No	Fiber optic	No	Yes	Yes	Yes	Yes	One year	No	Credit card	100.35	5481.1 No		
15	2300-MLKJL Male	0	No	No	10	Yes	No	No	No	No	No	No	No	No	No	No	Bank trans	42.3	1840.75 No	
16	5129-JPL Male	0	No	No	25	Yes	No	No	Fiber optic	Yes	No	Yes	Yes	Yes	Month-to-me Yes	Electronic cl	105.3	2686.05 No		
17	3655-SMCYF Female	0	Yes	Yes	69	Yes	No	Fiber optic	Yes	Yes	No	Yes	Yes	Two year	No	Credit card	113.75	7995.15 No		
18	1533-MLKJL Female	0	No	No	53	Yes	No	No	No	No	No	No	No	No	No	Mobile chec	56.95	1889.5 No		
19	9959-WDTKTMale	0	No	Yes	21	Yes	No	Fiber optic	Yes	No	No	Yes	Yes	Two year	No	Bank trans	106.7	7782.25 No		
20	4450-MLKJL Female	0	No	No	19	Yes	No	No	No	No	No	No	No	No	No	Credit card	106.7	7782.25 Yes		
21	4183-MHJRH Female	0	No	No	21	Yes	No	Fiber optic	No	Yes	Yes	No	No	Month-to-me Yes	Electronic cl	90.05	1862.9 No			
22	8779-QHGM Male	1	No	No	13	Yes	No	No	No	DSL	No	No	No	No	No	Electronic cl	39.65	39.65 Yes		
23	1588-MLKJL Male	0	No	No	13	Yes	No	No	No	No	No	No	No	No	No	Bank trans	42.3	1840.75 No		
24	1066-JSKCK Male	0	No	No	1	Yes	No	No	No	No	No	No	No	No	No	Mobile chec	20.15	20.15 Yes		
25	3638-WTAKB Female	0	Yes	No	58	Yes	No	DSL	No	Yes	No	No	No	Two year	Yes	Credit card	59.9	3505.1 No		
26	4332-MLKJL Male	0	No	No	49	Yes	No	No	No	No	No	No	No	No	No	Credit card	106.7	7782.25 No		
27	6865-JZKHO Female	0	No	No	30	Yes	No	DSL	Yes	Yes	No	No	No	No	No	Bank trans	55.3	5330.6 No		
28	6467-MLKJL Male	0	Yes	No	47	Yes	No	No	Fiber optic	No	Yes	No	Yes	No	No	Electronic cl	99.35	4749.7 Yes		
29	4650-UTTAK Male	0	No	No	1	Yes	No	No	No	No	No	No	No	No	No	Bank trans	42.3	1840.75 Yes		
30	5248-YGUN Male	0	Yes	No	72	Yes	Yes	DSL	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Credit card	90.25	6369.45 No		
31	3773-MLKJL Female	0	No	No	17	Yes	No	No	No	No	No	No	No	No	No	Mobile chec	56.95	1889.5 No		
32	3841-NFJCK Female	1	Yes	No	73	Yes	No	Fiber optic	No	Yes	Yes	No	No	Two year	Yes	Credit card	96.25	8766.95 No		
33	4929-XHJWV Male	1	Yes	No	2	No	No	Fiber optic	No	No	Yes	Yes	Yes	Month-to-me Yes	Mobile chec	95.5	1815.65 No			
34	5887-IEAKC Male	0	No	No	27	Yes	No	DSL	Yes	Yes	Yes	No	No	One year	No	Mobile chec	66.5	1874.45 No		



Paquetes de R

Demostrar cuándo y cómo estos paquetes estos paquetes se pueden usar en un proyecto the auto-aprendizaje.





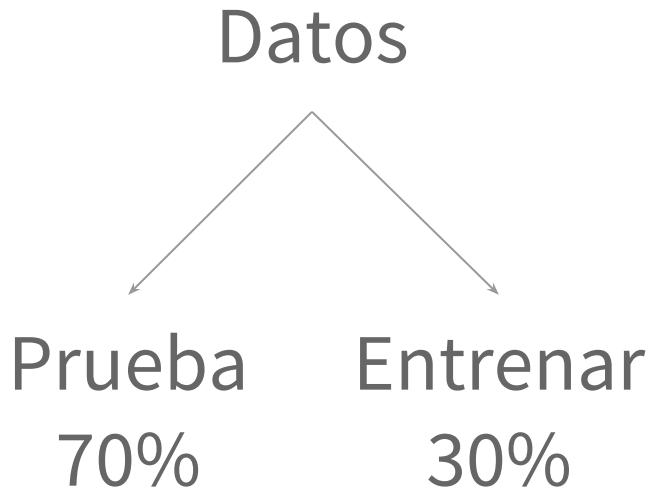
Leér datos

```
datos_perdimiento <- read_csv("customer_churn.csv")
```

```
# A tibble: 7,043 x 21
  customerID gender SeniorCitizen Partner Dependents tenure PhoneService
  <chr>        <chr>      <dbl>   <chr>    <chr>     <dbl>   <chr>
1 7590-VHVEG Female          0 Yes     No           1 No
2 5575-GNVDE Male            0 No      No           34 Yes
3 3668-QPYBK Male            0 No      No           2 Yes
4 7795-CFOCW Male            0 No      No           45 No
5 9237-HQITU Female          0 No      No           2 Yes
6 9305-CDSKC Female          0 No      No           8 Yes
7 1452-KIOVK Male            0 No      Yes          22 Yes
8 6713-OKOMC Female          0 No      No           10 No
9 7892-P0OKP Female          0 Yes     No           28 Yes
10 6388-TABGU Male           0 No      Yes          62 Yes
# ... with 7,033 more rows, and 14 more variables: MultipleLines <chr>
```



Pre-procesar - Muestra de datos



```
separa_datos <- initial_split(  
  datos_perdimiento,  
  prop = 0.3  
)  
tbl_entrenar <- training(separa_datos)  
tbl_prueba <- testing(separa_datos)
```



Pre-procesar - *La receta!*

Remueve columnas

Omitir NAs

Partir en categorías

Conversión logarítmica

Conversión lógica

Variables ficticias

Centrar valores

Escalar valores

```
receta <- tbl_entrenar %>%  
  
  recipe(Churn ~ .) %>%  
  step_rm(customerID) %>%  
  step_naomit(all_outcomes(), all_predictors()) %>%  
  step_discretize(tenure, options = list(cuts = 6)) %>%  
  step_log(TotalCharges) %>%  
  step_mutate(Churn = ifelse(Churn=="Yes", 1, 0)) %>%  
  step_dummy(all_nominal(), -all_outcomes()) %>%  
  step_center(all_predictors(), -all_outcomes()) %>%  
  step_scale(all_predictors(), -all_outcomes()) %>%  
  prep()
```



Pre-procesar - “Cocinar” receta

```
x_tbl_entrenar <- receta %>%
  juice(all_predictors(),
        composition = "matrix")

y_vec_entrenar <- receta %>%
  juice(all_outcomes()) %>%
  pull()
```

La muestra de **entrenamiento** es extraída

```
baked_test <- bake(receta, tbl_prueba)

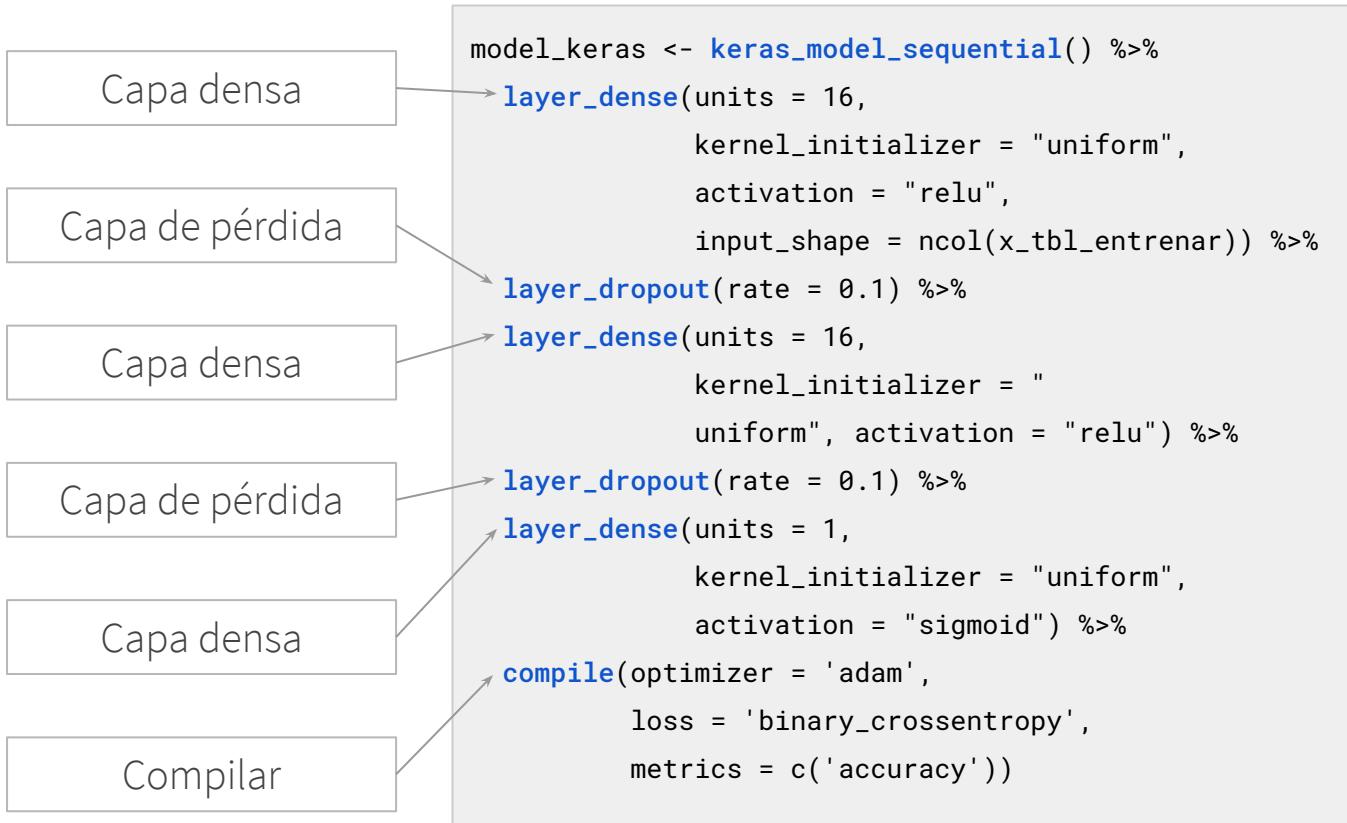
x_tbl_prueba <- baked_test %>%
  select(-Churn) %>%
  as.matrix()

y_vec_prueba <- baked_test %>%
  select(Churn) %>%
  pull()
```

La porción para **prueba** es procesada

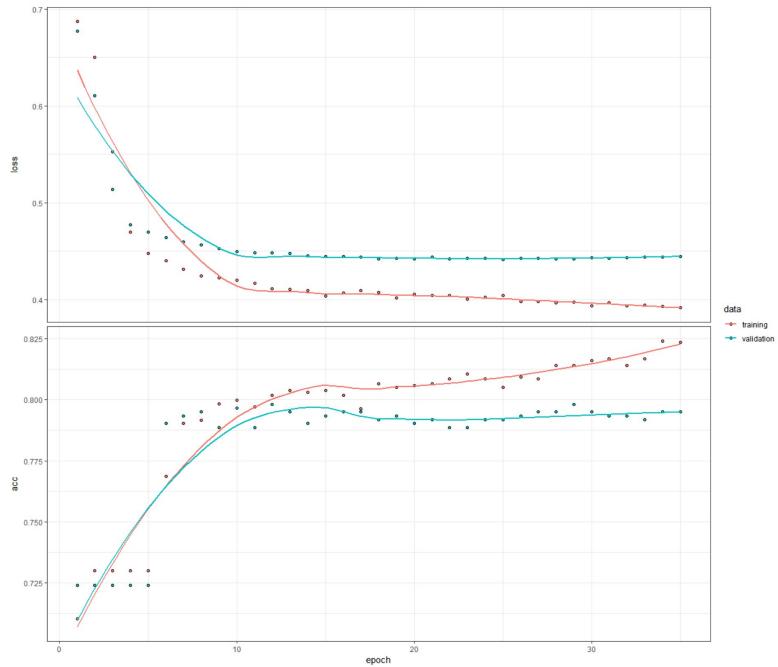
K

Entrenar el modelo - *Preparación*





Entrenar el modelo - *Entrenar el modelo*



```
history <- fit(  
  object = model_keras,  
  x = x_tbl_entrenar,  
  y = y_vec_entrenar,  
  batch_size = 50,  
  epochs = 35,  
  validation_split = 0.30,  
  verbose = 0  
)
```

K

Entrenar el modelo - *Predicciones*

```
yhat_keras_class_vec <- model_keras %>%
  predict_classes(x_tbl_prueba) %>%
  as.factor() %>%
  fct_recode(yes = "1", no = "0")
yhat_keras_prob_vec <- model_keras %>%
  predict_proba(x_tbl_prueba) %>%
  as.vector()
test_truth <- y_vec_prueba %>%
  as.factor() %>%
  fct_recode(yes = "1", no = "0")
estimates_keras_tbl <- tibble(
  truth      = test_truth,
  estimate   = yhat_keras_class_vec,
  class_prob = yhat_keras_prob_vec)
estimates_keras_tbl
```

```
# A tibble: 4,920 x 3
  truth estimate class_prob
  <fct> <fct>     <dbl>
1 no     yes       0.765
2 yes    no        0.368
3 yes    yes       0.749
4 yes    yes       0.782
5 no     yes       0.565
6 no     no        0.106
7 yes    yes       0.579
8 no     no        0.00366
9 no     no        0.221
10 no    no        0.0118
# ... with 4,910 more rows
```



Validar modelo - *Mediciones*

		Truth
Prediction	no	yes
no	3205	547
yes	420	748

```
conf_mat(estimate_keras_tbl,  
         truth, estimate)
```

```
# A tibble: 2 x 3  
.metric .estimator .estimate  
<chr> <chr> <dbl>  
1 accuracy binary 0.803  
2 kap binary 0.477
```

```
metrics(estimate_keras_tbl,  
        truth, estimate)
```

```
# A tibble: 1 x 3  
.metric .estimator .estimate  
<chr> <chr> <dbl>  
1 roc_auc binary 0.843
```

```
roc_auc(estimate_keras_tbl,  
        truth, class_prob)
```

```
estimate_keras_tbl %>%
```

```
# A tibble: 2 x 3  
.metric .estimator .estimate  
<chr> <chr> <dbl>  
1 precision binary 0.640  
2 recall binary 0.578
```

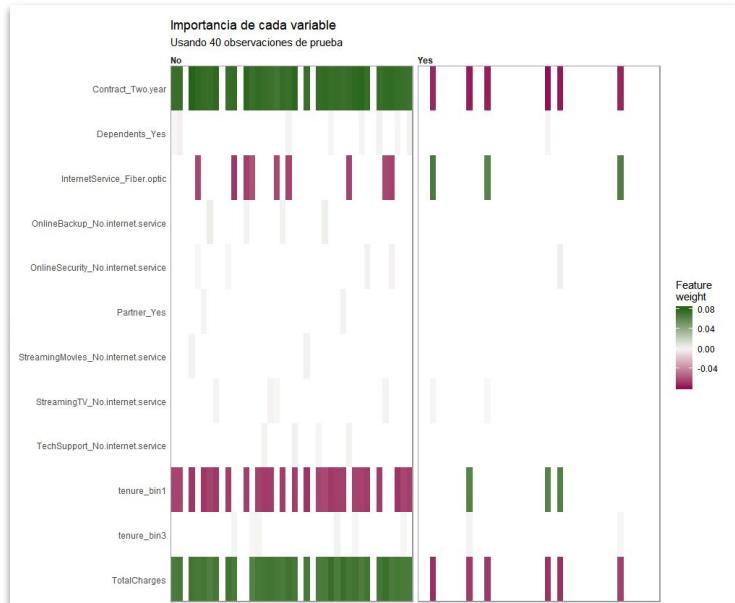
```
precision(truth, estimate) %>%  
bind_rows(estimate_keras_tbl %>%  
          recall(truth, estimate))
```

```
# A tibble: 1 x 3  
.metric .estimator .estimate  
<chr> <chr> <dbl>  
1 f_meas binary 0.607
```

```
f_meas(estimate_keras_tbl,  
       truth, estimate, beta = 1)
```



Validar modelo - Valores significantes



```
explainer <- x_tbl_entrenar %>%  
  as_tibble() %>%  
  lime(model_keras, bin_continuous = FALSE)  
  
explanation <- x_tbl_entrenar %>%  
  as.data.frame() %>%  
  head(40) %>%  
  lime::explain(  
    explainer = explainer, n_labels = 1,  
    N_features = 4, kernel_width = 0.5)
```



Compartir resultados - *Reportes*

Aprendizaje Automatico con Tensorflow y R

Code ▾

Install necessary packages

Hide

```
pkgs <- c("keras", "lime", "rsample", "recipes", "yardstick", "corrr")
install.packages(pkgs)
```

tidyverse

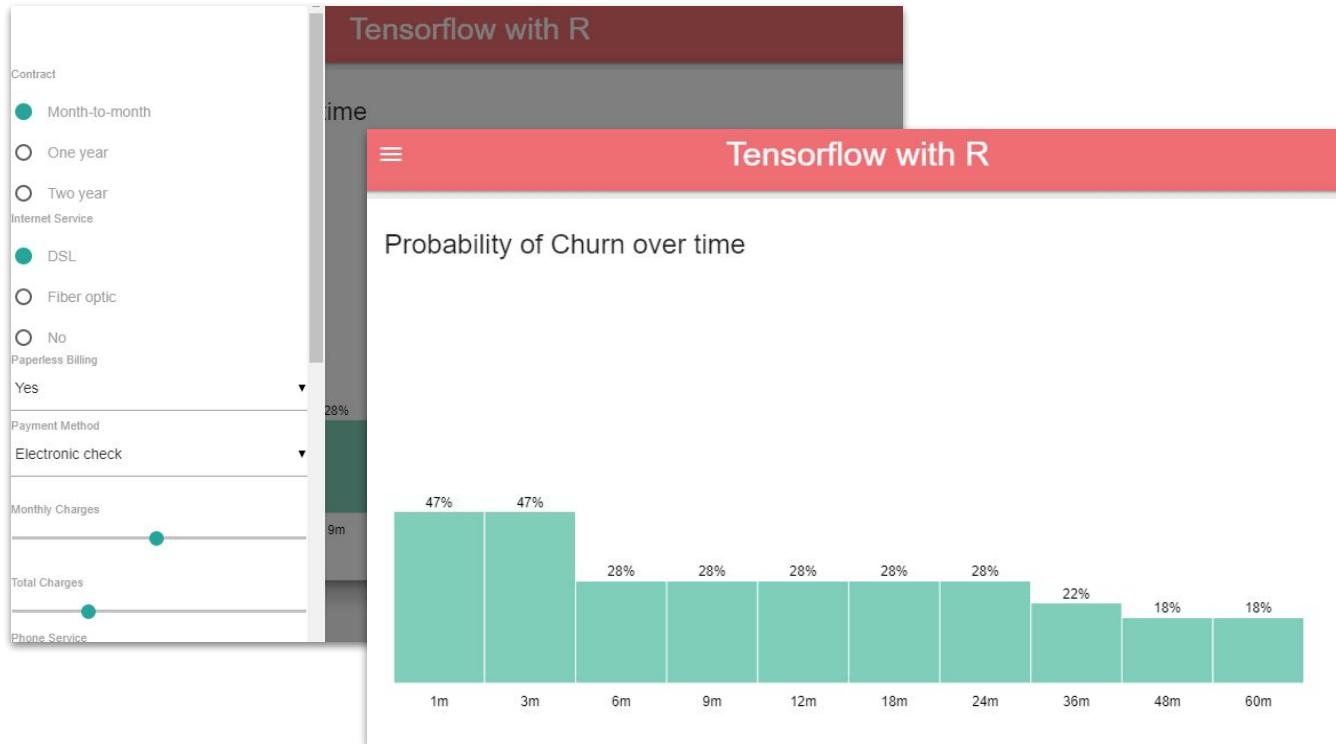
<http://tidyverse.org/>

The `tidyverse` packages provide an easy way to **import**, **tidy**, **transform** and **visualize** the data. Some of its component R packages are:

- `dplyr`
- `tidyr`
- `readr`
- `ggplot2`



Compartir resultados - Aplicación



Demostración

¡Pruébalo en tu teléfono!

rstd.io/churn

