

# Aprendizaje Automático con Tensorflow y R

Edgar Ruiz

 edgararui

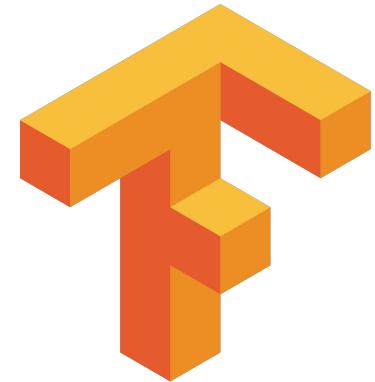
 theotheredgar

 edgararui



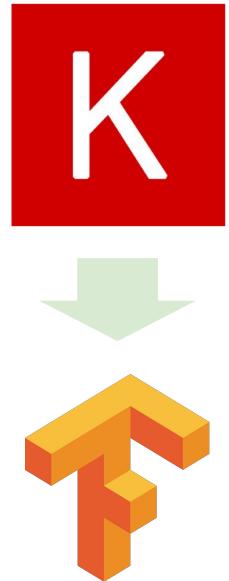
# Tensorflow

- ❑ Para el aprendizaje automático y profundo.
- ❑ Computación de alto desempeño
- ❑ Librería de código abierto
- ❑ Variedad de plataformas y dispositivos

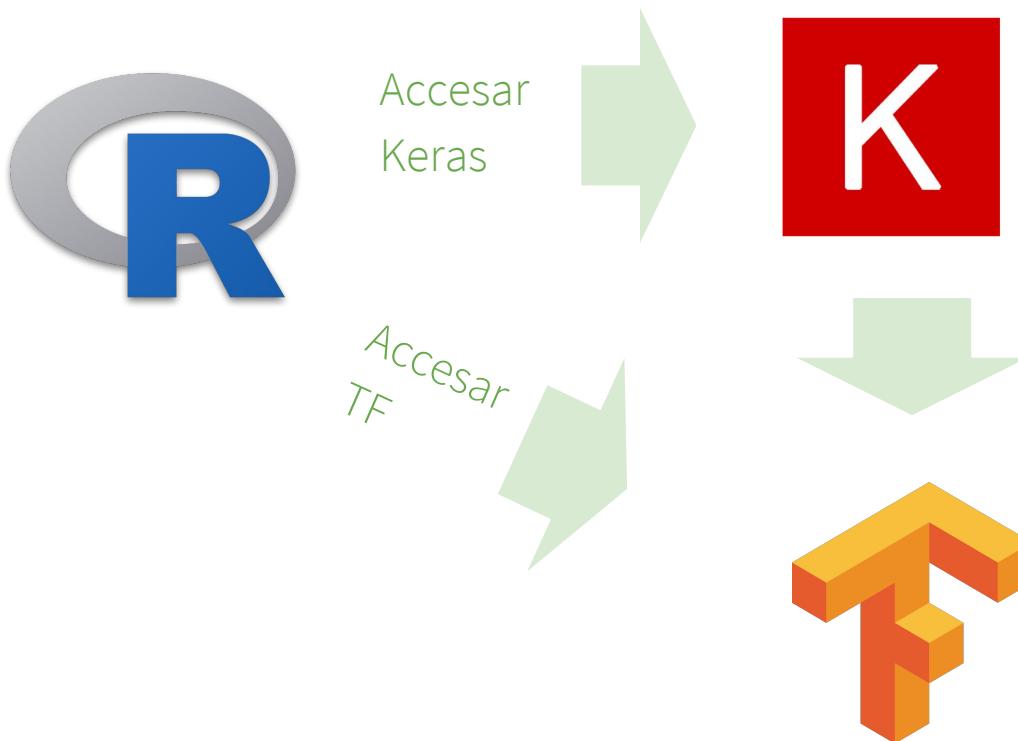


# Keras

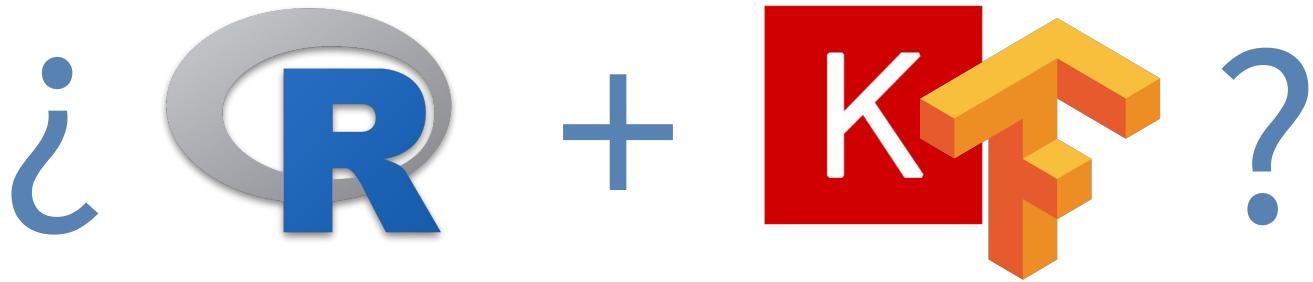
- ❑ Se integra con Tensorflow y otros
- ❑ El mismo código funciona en CPU o GPU
- ❑ Facilita el desarrollo de modelos
- ❑ Redes convolucionales y recurrentes



# Fácil integración de R, Tensorflow y Keras



# Pero, por que...



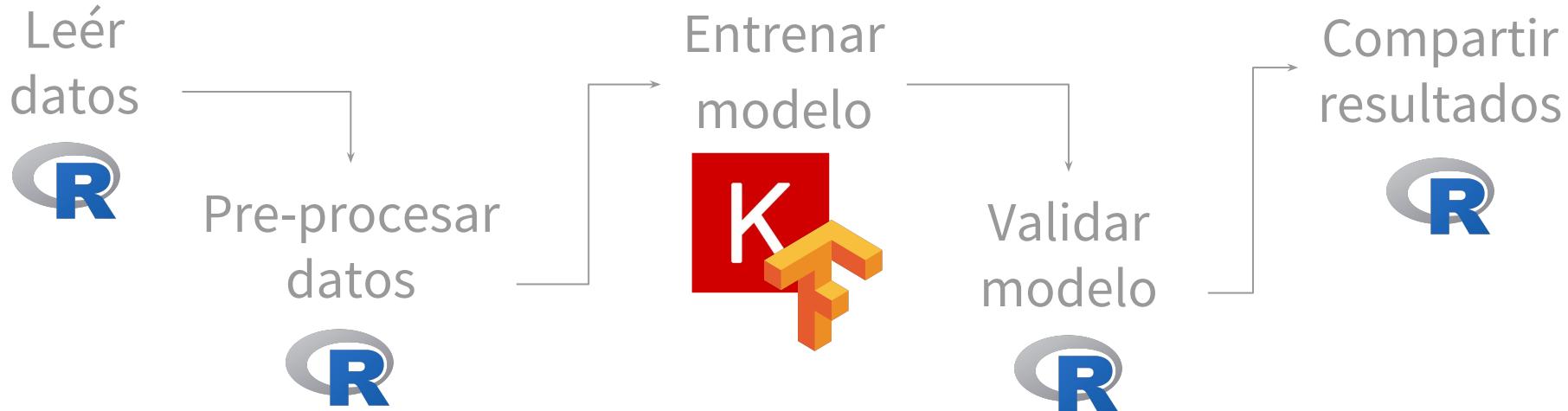
# Aprendizaje Automático es de varios pasos



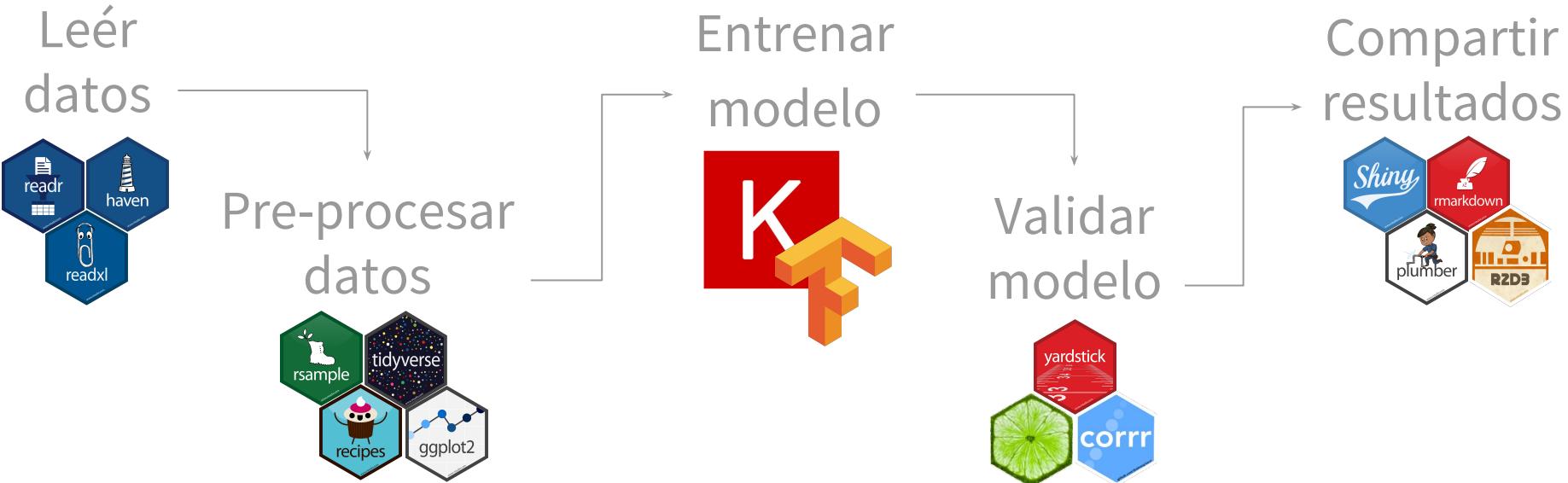
# Entrenar el modelo es solo un paso...



# R se especializa en todos los pasos



# 13 mil paquetes especializados de R



# Prediciendo pérdida de clientes

Demostración

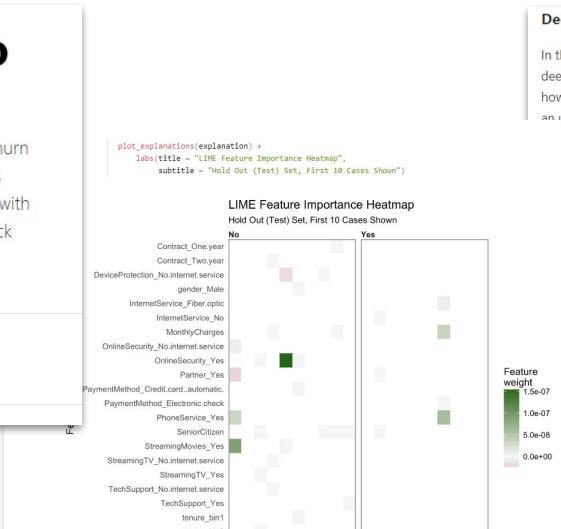
# Análisis original

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

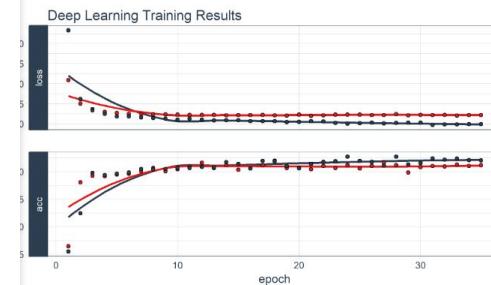
Using Keras to predict customer churn based on the IBM Watson Telco Customer Churn dataset. We also demonstrate using the lime package to help explain which features drive individual model predictions. In addition, we use three new packages to assist with Machine Learning: recipes for preprocessing,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



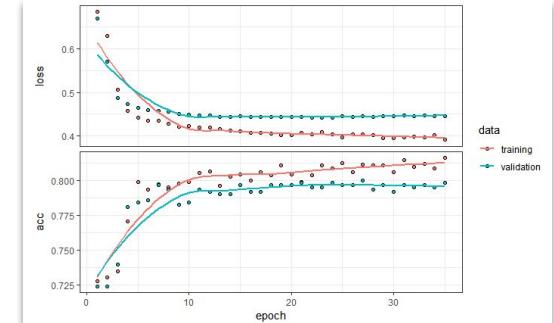
<https://blogs.rstudio.com/tensorflow/posts/2018-01-11-keras-customer-churn/>

# Objetivo

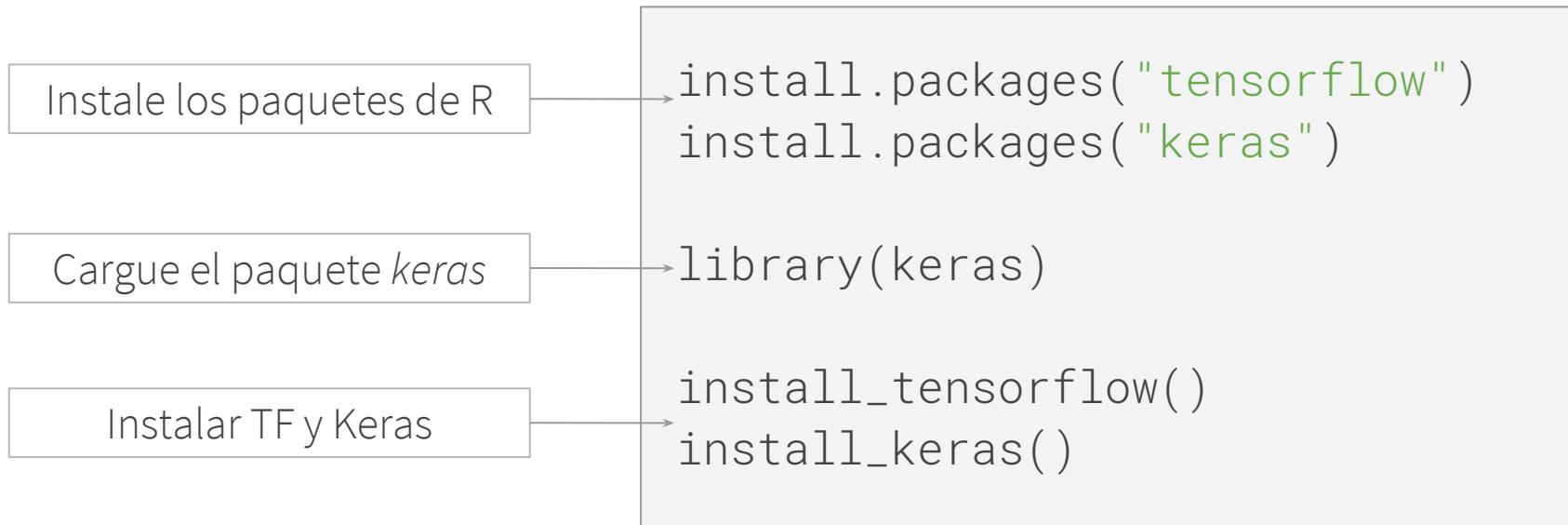
Analizar datos de pérdida de clientes de una **compañía de telecomunicaciones** con R, Keras y Tensorflow.

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	EverMarrying	EverWorked	StreamingMovies	StreamingTV	Contract	PaperlessBilling	PaymentMethod	MonthlyChurn	TotalChurn	Churn
2	7500-VINICG Female	O	Yes	No	2	No	No	DSL	Yes	No	No	No	No	No	Month-to-mo	Yes	Electronic cl	29.85	29.85	Yes
3	5575-GN001 Male	O	No	No	34	Yes	No	DSL	Yes	Yes	No	No	No	No	One year	No	Mobile chec	56.95	1889.5 No	No
4	3662-ML001 Male	O	No	No	3	No	No	No	No	No	No	No	No	No	Month-to-mo	No	Mobile chec	56.95	1889.5 No	No
5	7595-CFCVCMale	O	No	No	45	No	No	No	DSL	Yes	Yes	No	No	No	One year	No	Bank trans	42.3	1840.75 No	No
6	9237-HQ001 Male	O	No	No	7	Yes	No	No	Fiber optic	No	No	No	No	No	Month-to-mo	Yes	Electronic cl	70.15	1840.75 Yes	Yes
7	3200-ZK001 Male	O	No	No	2	Yes	No	No	Fiber optic	No	Yes	No	No	No	Month-to-mo	Yes	Mobile chec	60.0	825.5 Yes	Yes
8	1452-KD001 Male	O	No	Yes	22	Yes	No	No	Fiber optic	No	Yes	No	No	No	Month-to-mo	Yes	Credit card	89.1	1949.4 No	No
9	4731-CH001 Male	O	No	No	19	Yes	No	No	No	No	No	No	No	No	Month-to-mo	No	Mobile chec	56.95	1889.5 No	No
10	7893-PO001 Female	O	Yes	No	29	Yes	Yes	No	Fiber optic	No	Yes	Yes	Yes	Yes	Month-to-mo	Yes	Electronic cl	104.8	3066.05 Yes	Yes
11	6388-TA001 Male	O	No	Yes	62	Yes	No	DSL	Yes	Yes	No	No	No	No	One year	No	Bank trans	56.15	3487.95 No	No
12	1393-TR001 Male	O	No	No	11	Yes	No	No	No	No	No	No	No	No	Month-to-mo	No	Mobile chec	56.95	1889.5 No	No
13	7469-LX001 Male	O	No	No	15	Yes	No	No	No	No	No	No	No	No	Month-to-mo	No	Credit card	18.95	326.8 No	No
14	8091-TV001 Male	O	Yes	No	58	Yes	No	No	Fiber optic	No	Yes	Yes	Yes	Yes	One year	No	Credit card	100.25	1581.1 No	No
15	2300-CH001 Male	O	No	No	49	Yes	No	No	No	No	No	No	No	No	Month-to-mo	No	Bank trans	56.95	1889.5 No	No
16	5129-JP001 Male	O	No	No	25	Yes	No	No	Fiber optic	Yes	No	Yes	Yes	Yes	Month-to-mo	Yes	Electronic cl	105.2	2686.05 No	No
17	3655-SM001 Female	O	Yes	Yes	69	Yes	Yes	No	Fiber optic	Yes	Yes	No	No	No	Two year	No	Credit card	113.75	7995.15 No	No
18	4133-CH001 Male	O	No	No	53	Yes	No	No	No	No	No	No	No	No	Month-to-mo	No	Mobile chec	56.95	1889.5 No	No
19	9959-WDTKTMale	O	No	Yes	21	Yes	No	No	Fiber optic	Yes	No	No	Yes	Yes	Two year	No	Bank trans	106.7	7782.25 No	No
20	4450-CH001 Male	O	No	No	19	Yes	No	No	No	No	No	No	No	No	Month-to-mo	No	Credit card	100.25	1581.1 No	No
21	4183-MH001 Female	O	No	No	21	Yes	No	No	Fiber optic	Yes	Yes	No	No	No	Month-to-mo	Yes	Electronic cl	90.05	1862.9 No	No
22	8779-QH001 Male	I	No	No	13	No	No	No	No	DSL	No	No	No	No	Month-to-mo	Yes	Electronic cl	39.65	39.65 Yes	Yes
23	1588-CH001 Male	O	No	No	13	Yes	No	No	No	No	No	No	No	No	Month-to-mo	No	Bank trans	56.95	1889.5 No	No
24	1066-JSKCK Male	O	No	No	1	Yes	No	No	No	No	No	No	No	No	Month-to-mo	No	Mobile chec	20.15	20.15 Yes	Yes
25	3638-WT001 Female	O	Yes	No	58	Yes	No	DSL	No	Yes	No	No	No	Two year	Yes	Credit card	59.9	1505.1 No	No	
26	4332-CH001 Male	O	No	No	49	Yes	No	No	No	No	No	No	No	Month-to-mo	No	Credit card	100.25	1581.1 No	No	
27	6865-JZK001 Female	O	No	No	30	Yes	No	DSL	Yes	Yes	No	No	No	Month-to-mo	No	Bank trans	55.3	5330.6 No	No	
28	6467-LT001 Male	O	Yes	No	47	Yes	No	No	Fiber optic	No	Yes	No	Yes	Yes	Month-to-mo	Yes	Electronic cl	99.35	4749.7 Yes	Yes
29	4650-LT001 Male	O	No	No	1	Yes	No	No	No	No	No	No	No	Month-to-mo	No	Credit card	100.25	1581.1 No	No	
30	5248-YG001 Male	O	Yes	No	72	Yes	Yes	DSL	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Credit card	90.25	6369.45 No	No	
31	3773-NF001 Male	O	No	No	19	Yes	No	No	No	No	No	No	No	Month-to-mo	No	Mobile chec	96.25	8766.95 No	No	
32	3841-NF001 Female	I	Yes	No	73	Yes	No	No	Fiber optic	No	Yes	Yes	Yes	Yes	Month-to-mo	Yes	Credit card	96.25	8766.95 No	No
33	4292-XH001 Male	I	Yes	No	2	Yes	No	No	Fiber optic	No	No	No	No	No	Month-to-mo	Yes	Mobile chec	95.5	1815.6 No	No
34	6887-IC001 Male	O	No	No	27	Yes	No	DSL	Yes	Yes	Yes	Yes	Yes	One year	No	Mobile chec	66.5	1874.45 No	No	



# Instale TF y Keras desde R



# Paquetes de R para cada paso





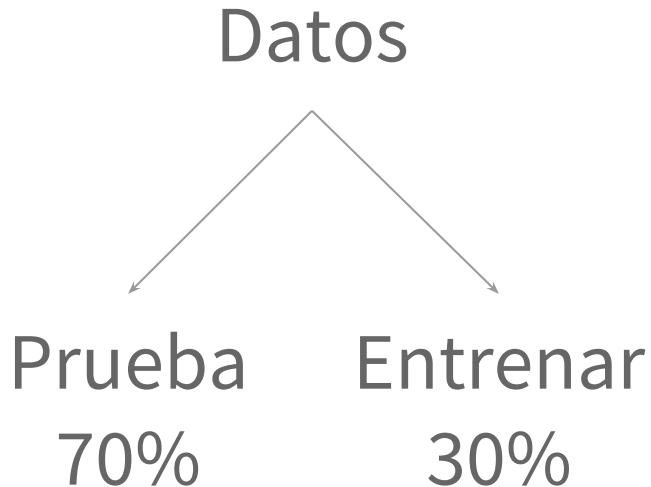
# Leér datos

```
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)

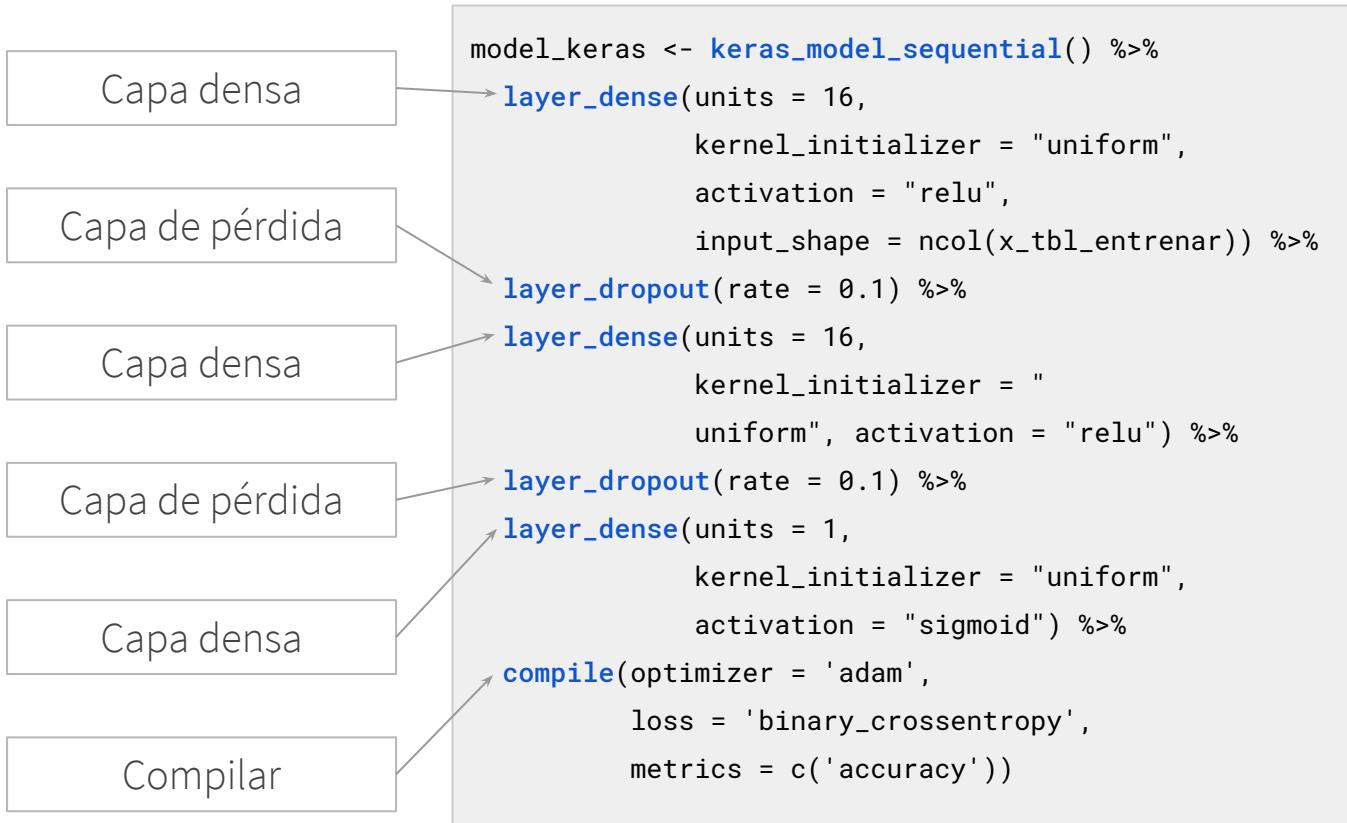
x_tbl_prueba <- baked_test %>%
  select(-Churn) %>%
  as.matrix()

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

La porción para **prueba** es procesada

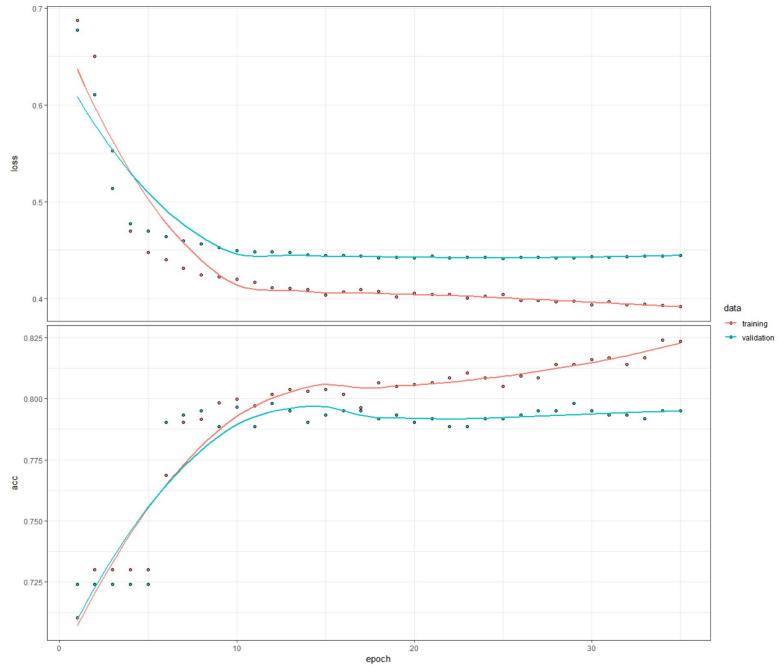
# K

# 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)
```

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

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

```
estimate_keras_tbl %>%
```

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

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

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

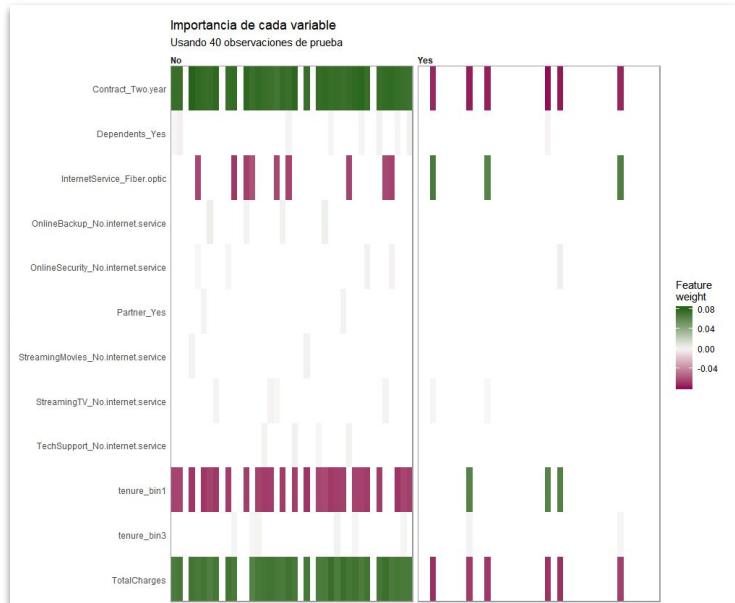
```
# A tibble: 1 x 3  
.metric .estimator .estimate  
<chr> <chr> <dbl>  
1 roc_auc binary 0.843
```

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

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



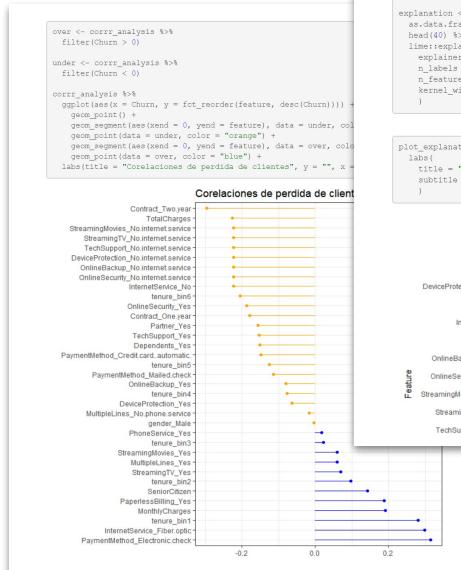
# Validar modelo - Valores significantes



```
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*

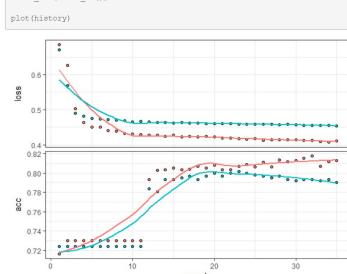


```
library(lime)

explains <- x_tbl_entrenar %>%
  as_tibble() %>%
  lime(model_keras,
       bin_continuous = FALSE)

explanation <- x_tbl_entrenar %>%
  as.data.frame() %>%
  head(40) %>%
  lime::explain(
    explanation = explainer,
    n_explanations = 1,
    n_features = 4,
    kernel_width = 0.5
  )

plot_explanations(explanation) +
  labs(
    title = "Importancia de cada variable",
    subtitle = "Usando 40 observaciones de prueba"
  )
```



truth	estimate	class_prob
<dbl>	<dbl>	<dbl>
no	yes	0.5608066916
yes	no	0.2966657579
yes	yes	0.5619708896
yes	yes	0.5619708896
no	yes	0.5364581347

## Aprendizaje Automatico con Tensorflow y R

### Instalar paquetes

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

### tidyverse

<http://tidyverse.org/>

```
Parsed with column specification:
cols(
  .default = col_character(),
  SeniorCitizen = col[32mcol_double()](39m,
  tenure = col[32mcol_double()](39m,
  MonthlyCharges = col[32mcol_double()](39m,
  TotalCharges = col[32mcol_double()](39m
)
See spec(...) for full column specifications.
```

```
glimpse(datos_perdimiento)
```

### rsample

<https://tidymodels.github.io/rsample/>

```
library(rsample)
set.seed(100)

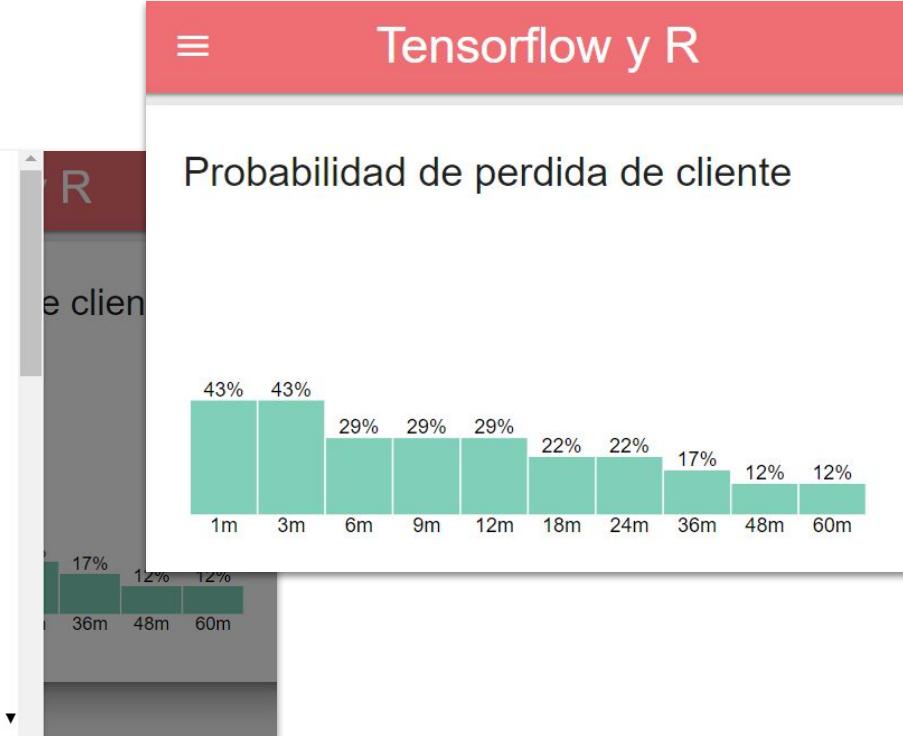
separa_datos <- initial_split(
  datos_pardimiento,
  prop = 0.3)

tbl_entrenar <- training(separa_datos)
tbl_prueba <- testing(separa_datos)
```





# Compartir resultados - Aplicación



# Demostración

¡Pruébalo en tu teléfono!

[rstd.io/Clientes](https://rstd.io/Clientes)

