

# PREDICCIÓN DE CHURN EN MOBILE GAMING

## DATA MINING - MAESTRIA MIM

Pedro Ferrari

10 de Abril de 2021



- ① **Sobre Mobile Gaming y el Problema de Churn**
- ② Competencia Tipo Kaggle
- ③ Análisis Exploratorio de Datos e Ingeniería de Atributos
- ④ Nuestro Futuro

# EL MERCADO DE MOBILE GAMING

Market

Mobile is the largest and fastest-growing platform within video games

Global video game market:

**\$149B** in 2019

Out of which **\$68B** in mobile

(that's more than music and movies markets combined)

Mobile grows **10%** YoY.

(vs. to **3%** for PC and **7%** for console)

Market

The gaming market is bigger than other main entertainment industries combined.



**\$137.9B**



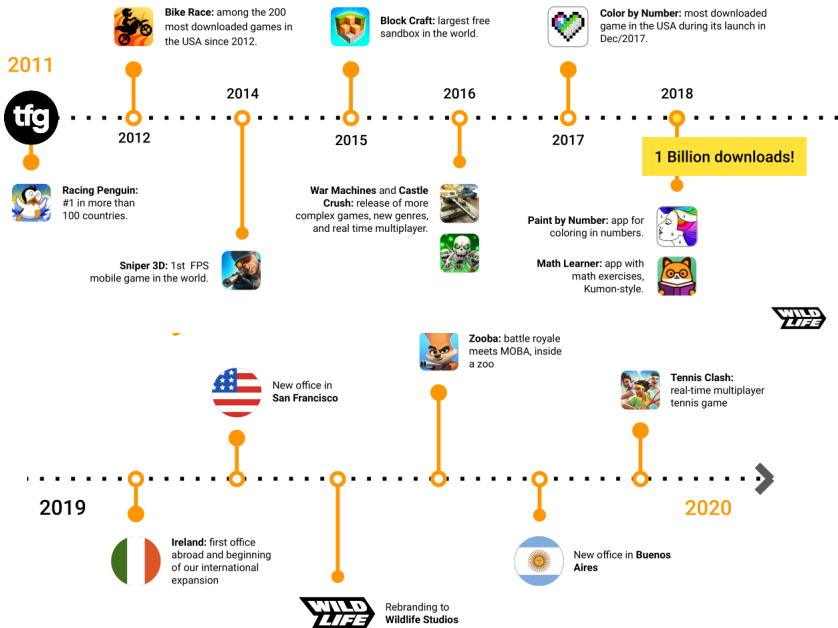
**\$41B**



**\$17B**

in 2018

# SOBRE WILDLIFE STUDIOS



# DATA (SCIENCE) EN WILDLIFE Y MOBILE GAMING

- *Big Data*

- Reciben más de 18PB de nuevos datos todo los días y procesan 1PB
- Data relacionada a todo tipo de comportamiento de los usuarios: sesiones, ingresos, publicidad, eventos dentro de los juegos, etc

- Diversas aplicaciones de Data Science para toma de decisiones:

- Predicción de Lifetime Value
- Optimización de estrategias de bidding y selección de creativos en publicidad
- Inferencia causal: tests A/B
- Sistema de recomendación y definición de promociones para compras dentro del juego (explican el 95 % de los ingresos de la compañía)



Zooba



War Machines

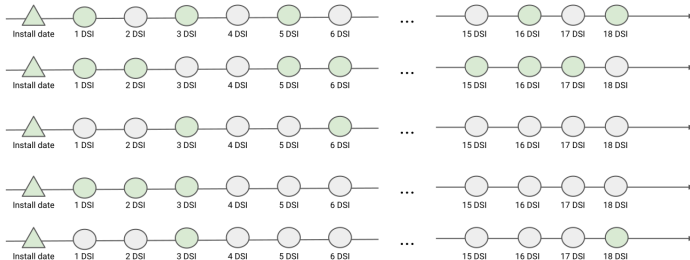


- **Predicción de Churn** ¿Por qué?

- Para lograr *re-engagement* de los usuarios a través de notificaciones, premios especiales, descuentos, etc

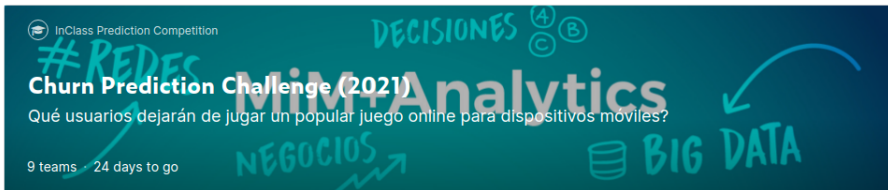


- Wildlife nos proveyó un dataset del juego de estrategia “Castle Crush”
  - Contiene fechas de instalación, variables categóricas anonimizadas, variables monetarias, suma de eventos dentro del juego, etc
- Objetivo: predecir *churn*, usuarios que jugaron al tercer día de haberlo instalado y no jugaron en las dos semanas siguientes
  - Existe cierta censura en los datos



- ① Sobre Mobile Gaming y el Problema de Churn
- ② **Competencia Tipo Kaggle**
- ③ Análisis Exploratorio de Datos e Ingeniería de Atributos
- ④ Nuestro Futuro





InClass Prediction Competition

#REDES

DECISIONES (4) (B)

Churn Prediction Challenge (2021)

Qué usuarios dejarán de jugar un popular juego online para dispositivos móviles?

9 teams · 24 days to go

NEGOCIOS

BIG DATA

MIM-Analytics

- Formalidades:

- *Hosteado* en [Kaggle](#) con fecha de finalización 02/05.
- Equipos de 3 (o 4).
- Lenguajes: R (preferido) o Python.
- Entregables
  - **Ayer**: hacer un primer *submit* con un modelo (simple)
  - Subir/enviar dos gráficos relevantes analizando los datos antes del Domingo 18/04.
  - Informe final (en espíritu similar a [este](#)) explicando trabajo realizado antes de la primera fecha de examen.

- Podemos *framear* la competencia como un problema de clasificación binaria...

THIS LOOKS  
LIKE A JOB



- ① Sobre Mobile Gaming y el Problema de Churn
- ② Competencia Tipo Kaggle
- ③ **Análisis Exploratorio de Datos e Ingeniería de Atributos**
- ④ Nuestro Futuro

## Data Scientist



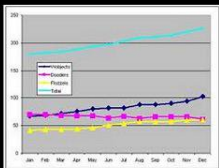
What my friends think I do



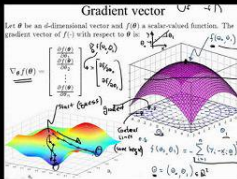
What my mom thinks I do



What society thinks I do



What my boss thinks I do



What I think I do



What I actually do

- EDA: Análisis exploratorio de datos (Tukey [1977])
  - Data mining en su acepción pura: técnicas para analizar los datos, encontrar patrones y generar insights
    - Visualizaciones: univariadas (para un mismo feature), bi-variadas (entre features y target), multivariadas (entre distintos features)
    - Técnicas de estadística clásica: correlaciones, test de hipótesis, ANOVA
    - Reducción de dimensionalidad: PCA, SVD
    - Clustering
  - Objetivos:
    - Comprender el dataset
    - Definir y refinar la selección e ingeniería de atributos que alimentan los modelos
- ¿Que es realmente hacer data science en la industria?
  - **E**xtract **T**ransform **F**it **L**oad
  - Distribución de tiempos: E 30 % T 50 % F 10 %, L 10 %

Empecemos a explorar el dataset de la competencia ...

# EL PRINCIPIO DEL PRINCIPIO

- Extracción (carga) de los datos
  - Cargamos un archivo de datos de entrenamiento usando la librería `data.table` :

```
library("data.table")

load_csv_data <- function(csv_file, sample_ratio = 1, drop_cols = NULL,
                          sel_cols = NULL) {
  dt <- fread(csv_file, header = TRUE, sep = ",", stringsAsFactors = TRUE,
              na.strings = "", drop = drop_cols, select = sel_cols,
              showProgress = TRUE)

  if (sample_ratio < 1) {
    sample_size <- as.integer(sample_ratio * nrow(dt))
    dt <- dt[sample(.N, sample_size)]
  }
  return(dt)
}

csv_file <- "../.../R/code/competition-data/train_1.csv"
df <- load_csv_data(csv_file, sample_ratio = 1)
```

- ¿Cómo reordenamos las columnas alfabéticamente?

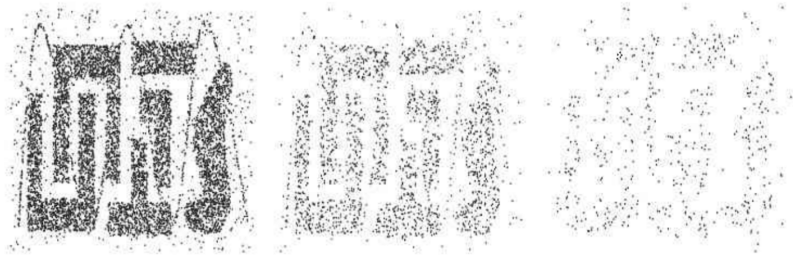
```
setcolorder(df, sort(colnames(df)))
```

- ¿Cómo definimos nuestro target? (sin entrar en cuestiones de censura de información)

```
df[, Label := as.numeric(Label_max_played_dsi == 3)]
```



# DIGRESIÓN: ¿POR QUE PODEMOS USAR UNA MUESTRA?



Fuente: *Introduction to Data Mining*

- Si son muchos datos conviene tener alguna forma de acceder fácilmente a una muestra (aleatoria) de ellos. Pero ....
- Reducir tamaño de muestra genera pérdida de estructura
  - ¿Es un problema cuando uno tiene casi 6M de datos?
- ¿Funciona siempre el muestreo aleatorio simple?
  - En clases muy desbalanceadas se suele hacer muestreo estratificado...

# VOLVIENDO AL DATASET

- Partiendo del código anterior:

- ¿Cuántas observaciones y features tenemos?

```
R> dim(df)
[1] 1109485    102
```

- ¿Cómo es la estructura de los datos?

```
R> str(df)
Classes 'data.table' and 'data.frame': 1109485 obs. of 102 variables:
 $ age                               : num  NA NA NA NA NA NA NA NA NA NA ...
 $ BuyCard_sum_dsi0                 : int  0 0 0 0 0 0 5 0 12 0 ...
```

- ¿Están las clases positivas y negativas desbalanceadas?

```
R> table(df$Label)
 0      1
897923 211562
R> prop.table(table(df$Label))
 0      1
0.8093151 0.1906849
```

- ¿Cómo levantamos solo una muestra de 10 % de una única columna?

```
R> df_sample <- load_csv_data(csv_file, sample_ratio = 0.1,
                             sel_cols = c("Label_max_played_dsi"))
R> prop.table(table(df_sample$Label_))
 3      4
0.19192775 0.08697768
```

## EJERCICIO 1

Tenemos multiples archivos de entrenamiento y además no dijimos nada aun sobre el conjunto de test...

(i) Usando el código anterior:

- Escribir una función que levante todos los archivos del conjunto de entrenamiento, los concatene y los guarde en un nuevo (único archivo) csv (`train.csv`).  
¿Cuántas observaciones tenemos?
  - Ahora en vez de guardar todo guardar 10 %.
- Hard (*caching* (simple)): modificar la función anterior de tal manera que si el archivo ya existe entonces la función no realiza la rutina de cargado al ser llamada sino que directamente devuelve ese archivo.

```
# Hint to get started
load_train_data <- function(data_dir, train_file="train.csv", sample_ratio=1,
                             drop_cols=NULL, sel_cols=NULL) {
  train_days <- seq(1, 5, by=1)
  dfs <- list()
  # Do something...
  df <- (rbindlist(dfs, fill=TRUE))
  return(df)
}
```

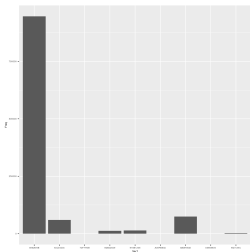
(ii) Modificar las funciones anteriores o escribir una nueva para poder cargar el conjunto de test.

- Buena idea: trabajar con un único dataset y agregar una columna de booleanos indicando si la observación pertenece al conjunto de entrenamiento o no. Por qué?



- ¿Cuáles son los valores más comunes y cuales los más extremos?
  - Mirar distribuciones con histogramas o, de acuerdo al caso, gráfico de barras o nubes de puntos. *text*
  - ¿Tiene sentido reemplazar valores extremos? **Ejercicio 2: Hacer otro gráfico**

```
library("ggplot2")  
ggplot(as.data.frame(table(df$categorical_6)), aes(x=Var1, y = Freq))  
  + geom_bar(stat="identity")
```



- Para valores numéricos computar medidas de resumen clásicas (media, desvío, etc):
  - ¿Tienen contenido informativo aquellas variable con nula o cuasi-nula varianza?

```
R> table(df$traffic_type)
```

2

1109485

# TRANSFORMACIÓN (LIMPIEZA DEL DATASET)

- Valores faltantes y repetidos

- ¿Cuál es la frecuencia de valores nulos o faltantes? ¿Vale la pena descartar por esto motivo alguna variable? ¿Imputamos valores?

```
na_prop <- sapply(df, function(x) sum(is.na(x)) / length(x))
df_na <- (data.frame(na_prop))
df_na <- df_na[df_na$na_prop > 0.7, , drop = FALSE]
R> df_na
      na_prop
age 0.9698563
site 0.7296493
```

- ¿Si hacemos modificaciones sobre el conjunto de entrenamiento estas se deberían replicar en el conjunto de test?
- ¿Hay filas repetidas? ¿Habría que conservarlas?
- Podemos, por ejemplo, usar la función duplicated de data.table para chequear esto sobre data sintética:

```
library(data.table)
df <- data.table(A = rep(1:3, each=4), B = rep(1:4, each=3), C = rep(1:2, 6), key = 1:6)
R> duplicated(df)
[1] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
```

- Podemos usar la función unique para quedarnos con valores únicos (filtrando duplicados).

# (ALGO DE) FEATURE ENGINEERING

- Generación de nuevas variables con algunos hints para empezar:
  - Counting features (ejemplo “country” count)
    - Agregamos una nueva variable llamada `country_count` con la cantidad de apariciones de cada país:

```
R> df[, country_count := .N, by = country]
R> head(df[, c("country", "country_count")])
  country country_count
1:     IN          166123
2:     PL           12663
3:     BY            4068
```

- “Tipos” de usuario
  - Usuario de tipo “acreedor”:

```
R> df[, creditor := ((soft_positive + hard_positive) -
  (soft_negative + hard_negative)) > 0]
R> tail(df[, c("creditor", "soft_positive", "hard_positive", "soft_negative",
  "hard_negative")])
  creditor soft_positive hard_positive soft_negative hard_negative
1:    TRUE           674           8           245           143
2:    TRUE           518          116           207           160
3:    TRUE          1232           18           692           198
4:    TRUE          1121           13           872           153
5:   FALSE           662          14          1193           104
```

- Bin counting y otras técnicas a ser presentadas en clase teórica

## ○ Ejercicio 3: Crear algún otro atributo

- ① Sobre Mobile Gaming y el Problema de Churn
- ② Competencia Tipo Kaggle
- ③ Análisis Exploratorio de Datos e Ingeniería de Atributos
- ④ **Nuestro Futuro**

- Para hacer por su cuenta en el ínterin:
  - Tener el dataset completamente analizado: completar los ejercicios que quedaron y hacer todo tipo de EDA relevante para tener intuiciones sobre los atributos.
  - Intentar fittear el dataset completo con los modelos que saben de la materia (o al menos con una muestra). ¿Varían mucho las métricas de performance entre el caso completo y el parcial?
  - **Hacer un segundo submit a Kaggle.**
- ¿Qué vamos a hacer durante la clase? Mostrar un pipeline completo y general para atacar este tipo de problemas.
  - Retomar la idea de ETFL
  - Ver distintos modelos y cómo performan.
  - Hacer tratamiento de atributos categóricos (¿darle una chance al “hashing trick”?)
  - Enfatizar (inclusive desde el código) la importancia de la transformación / procesamiento de los datos.
  - Otros misterios