

## **Tarea para el Hogar 2021-09-22**

La idea de esta tarea para el hogar es

- incorporar canaritos y ver las posiciones que ocupan en la importancia de variables, quizás esto sirva para acelerar las corridas.
- entrenar en la union de varios meses de la historia
- ampliar el dataset con variables del tipo lag y delta lag, Feature Engineering
- unir lo anterior al feature engineering.

El algoritmo que estamos utilizando es el LightGBM. ya no utilizamos otros.

La clase que estamos utilizando es la binaria2 pos={BAJA+2, BAJA+1} , neg={CONTINUA}

Todas las corridas se harán en Google Cloud

Los scripts de esta tarea se encuentran en la carpeta src/lightgbm

Usted debe completar la planilla andamios.ods

Atento a las sugerencias recogidas en la última clase, se han realizado los siguientes cambios:

- En la carpeta work ahora los archivos de un mismo experimento quedan debajo de una carpeta exclusiva para ese experimento llamada Exxxx
- El nombre del log de la optimización bayesiana, donde están los hiperparámetros utilizados en cada iteración, ahora termina con \_BOlog.txt
- En el archivo log de la optimización bayesiana ahora aparece un nuevo campo llamado iteracion\_bayesiana

To Protect and to Serve

Dimensiones del problema		
Dimension	Easy	Hard
Data Drifting	No elimino, ni transformo, ninguna variable con drifting	Elijo que variables eliminar y/o transformar. Esta es la tarea más "mística", al menos por la enorme cantidad de combinaciones que aparentemente hay.
Meses donde Entrenar	Entreno solo en [202011]	Entreno en n meses, que incluyen a 202011
Correccion del Dataset	Dejo las variables como estan, a pesar que estén dañadas un mes	Asigno nulo a las variables que están dañadas un mes, o algun tratamiento más sofisticado
Junio-2020 esta muy dañado	Dejo junio-2020 en el dataset	Elimino junio-2020 del dataset
Variables lags y deltas	No agrego	Agrego lags y deltas, al comienzo solo de orden 1, luego quizas orden 2
Variables nuevas de Feature Engineering	No agrego	Hago Feature Engineering

### 1. Prerrequisito Storage Bucket de Google Cloud

Conéctese en su navegador al Google Cloud Console <https://console.cloud.google.com/>  
Vaya al Cloud Storage Browser <https://console.cloud.google.com/storage/browser>

Allí deberá ver su bucket, haga click sobre su bucket y navegando deberá ver el siguiente contenido

- datasets
- datasetsOri
  - `paquete_premium.csv.gz`
- exp
- kaggle
- log
- modelitos
- work

### 2. Prerrequisito Imagen de la máquina Virtual

Desde el browser que tiene conectado a Google Cloud vaya al link  
<https://console.cloud.google.com/compute/images>

y verifique que tiene la imagen `image-dm`

### 3. Actualización en su PC local de su repositorio

Primero actualice SU repositorio github con el oficial de la materia, si no recuerda como hacerlo siga este instructivo <https://docs.github.com/es/github/collaborating-with-pull-requests/working-with-forks/syncing-a-fork>

Luego vaya a su PC local y actualice la copia que tiene en su pc local de su repositorio GitHub, con el comando `git pull`

#### 4. Script 711\_lgb\_bin2\_lagdelta.r

Este script debe correr en Google Cloud, al igual que todos los de esta tarea.

En esta corrida se agregarán al dataset los campos de tipo lag y delta lag.

Recuerde que siempre debe hacer el git pull de su repositorio en la máquina virtual recién creada.

Se entrenará como hasta ahora en la asignatura, solo en el mes de noviembre-2020 y por supuesto se aplicará el modelo a enero-2021, pero ahora se crearán las variables de tipo lag y delta lag

- Cambie la semilla del script por la suya
- En la línea `campos_malos <- c( "mpasivos_margen" )` `#aquí se deben cargar todos los campos culpables del Data Drifting` agregue TODAS las variables que usted encontró que cometen data drifting, recuerde esa es su "salsa mágica" que lo diferenciará de sus compañeros.
- Lea el código entre las líneas 240 y 280, ahí se agregan las variables lag y luego los canaritos

Recuerde que al inicio del script están los requerimientos de vCPU, memoria RAM y espacio en el disco rígido. Como para todas las corridas usted deberá crear una máquina preemptible.

Luego de la corrida, en work, dentro de la exclusiva carpeta de ese experimento, abra alguno de los archivos de importancia de variables y busque canaritos.

¿Qué quieren decir las variables que aparecen por debajo de los canaritos en importancia?

#### 5. Script 712\_lgb\_bin2\_lagdelta.r

El objetivo de este script es ver que sucede si NO se eliminan campos que hacen drifting, y compararlo con el script anterior.

En este script únicamente debe cambiar la semilla por la suya.

¿Que script da mayor ganancia en el Public Leaderboard, el 711 o el 712?

#### 6. Script 715\_fe\_simple.r

Este script hace el feature engineering sobre todos los 36 meses de historia.  
Agregue a este dataset SUS campos que con tanta sagacidad ha creado.

El script 715 deja la salida en /datasets/paquete\_premium\_ext.csv.gz

## 7. Script 717\_lgb\_bin2\_lagdelta\_ext.r

Este script correrá utilizando el dataset creado en el script 715  
En esta corrida se agregarán al dataset los campos de tipo lag y delta lag.  
Recuerde que siempre debe hacer el git pull en la maquina virtual recién creada.

Se entrenará como hasta ahora en la asignatura, solo en el mes de noviembre-2020 y por supuesto se aplicará el modelo a enero-2021, pero ahora se crearán las variables de tipo lag y delta lag

- Cambie la semilla del script por la suya
- En la línea `campos_malos <- c( "mpasivos_margen" )` *#aquí se deben cargar todos los campos culpables del Data Drifting* agregue TODAS las variables que usted encontró que cometen data drifting, recuerde esa es su "salsa mágica" que lo diferenciará de sus compañeros.
- Lea el código entre las líneas 240 y 280, ahí se agregan las variables lag y luego los canaritos

¿Que script da mayor ganancia en el Public Leaderboard, el 711 , 712 o 717 ?

## 8. Script 721\_lgb\_bin2\_hist.r

Esta es una corrida muy pesada.

En esta corrida se entrenará en 11 meses de historia, en la union de los meses de enero-2020 a noviembre-2020

NO se crearán las variables de tipo lag y delta lag

- Cambie la semilla del script por la suya
- En la linea `campos_malos <- c( "mpasivos_margen" )` `#aqui se deben cargar todos los campos culpables del Data Drifting` agregue TODAS las variables que usted encontró que cometen data drifting, recuerde esa es su "salsa mágica" que lo diferenciará de sus compañeros.
- Lea el código entre las líneas 240 y 280, ahí se agregan las variables lag y luego los canaritos

Recuerde que al inicio del script estan los requerimientos de vCPU, memoria RAM y espacio en el disco rígido. Como para todas las corridas usted deberá crear una máquina preemptible.

Luego de la corrida, en work, dentro de la exclusiva carpeta de ese experimento, abra alguno de los archivos de importancia de variables y busque canaritos.

Este script es muy lento en correr ya que entrena en 11 meses.

Luego de las tres ultimas corridas, que conclusión obtiene?

Atención que la ganancia de 5-fold cross validation en este caso, al entrenar en 11 meses, es extremadamente alta y NO es comparable con la ganancia de 5-cross validation de un solo mes.

## 9. Script 740\_graficar\_zero\_rate.r

El data drifting fue un problema de entre noviembre-2020 a enero-2021  
Pero ahora estamos probando entrenar en mayor cantidad de meses  
Pasamos a analizar que campos tienen problemas y en que meses.

Correr el script 740\_graficar\_zero\_rate.r

Analizar los resultados que se generan en la carpeta work:

- zeroes\_ratio.pdf
- nas\_ratio.pdf
- promedios.pdf
- promedios\_nocero.pdf

Analizar en profundidad el script 740\_graficar\_zero\_rate.r y cada una de las salidas generadas.

Esta parte de la tarea le llevará por lo menos 60 minutos de materia gris intensiva.

## 10. Script 745\_corrige.r

A partir del análisis de las salidas del script 740, se decide pasar para cada campo que tiene severos problemas en un mes, pasarlo a nulo.

Leer el script en detalle. Son bienvenidas ideas alternativas, modifique a gusto el script !

El script 745 deja la salida en /datasets/paquete\_premium\_corregido.csv.gz

## 11. Script 746\_corrige\_fe.r

Este script corrige y agrega nuevos campos con feature engineering.  
Agregue a este dataset SUS campos de feature engineering

El script 746 deja la salida en  
/datasets/paquete\_premium\_corregido\_ext.csv.gz

## 12. Script 751\_lgb\_bin2\_histcorregido.r

En esta corrida se entrenará en 11 meses de historia, en la union de los meses de enero-2020 a noviembre-2020 y trabaja sobre el dataset corregido `datasets/paquete_premium_corregido.csv.gz`

NO se crearán las variables de tipo lag y delta lag

- Cambie la semilla del script por la suya
- En la linea `campos_malos <- c( "mpasivos_margen" )` `#aqui se deben cargar todos los campos culpables del Data Drifting` agregue TODAS las variables que usted encontró que cometen data drifting, recuerde esa es su "salsa mágica" que lo diferenciará de sus compañeros.
- Lea el código entre las líneas 240 y 280, ahí se agregan las variables lag y luego los canaritos

Recuerde que al inicio del script estan los requerimientos de vCPU, memoria RAM y espacio en el disco rígido. Como para todas las corridas usted deberá crear una máquina preemptible.

Luego de la corrida, en work, dentro de la exclusiva carpeta de ese experimento, abra alguno de los archivos de importancia de variables y busque canaritos.

Este script es muy lento en correr ya que entrena en 11 meses.

Luego de las corridas del script 721 u 751, que conclusión obtiene?



### 13. Script 761\_lgb\_bin2\_apiacere.r

Es una corrida extremadamente pesada.

En esta corrida se entrenará en 11 meses de historia, en la union de los meses de enero-2020 a noviembre-2020 y trabaja sobre el dataset corregido

Se crearán las variables de tipo lag y delta lag

Esta es una corrida realmente pesada que llevará más de 24 horas.

En función de los resultados obtenidos en las tareas anteriores usted deberá determinar que camino seguir:

En la linea 48 se deberá elegir alguna de estas cuatro opciones:

```
karch_dataset <- "./datasetsOri/paquete_premium.csv.gz"
karch_dataset <- "./datasetsOri/paquete_premium_ext.csv.gz"
karch_dataset <- "./datasets/paquete_premium_corregido.csv.gz"
karch_dataset <- "./datasets/paquete_premium_corregido_ext.csv.gz"
```

En la linea 66

```
campos_malos <- c("mpasivos_margen") #aqui se deben cargar todos los
campos culpables del Data Drifting
```

se deberá decidir se se elige algun campo como malo, o ninguno.