

Data analytics certificate MIT - Universidad de Chile  
**Trabajo Final Data Analytics**

---

Modelo de predicción de fuga de clientes para una institución banca personas.



Alumnos:  
Felipe González  
Juan Ignacio Paredes  
Osvaldo Ceballos

## Índice

1. Descripción General del proyecto.
  - 1.1. Descripción del contexto del proyecto.
  - 1.2. Descripción del informe.
  - 1.3. Objetivos generales.
  - 1.4. Objetivos específicos.
2. Análisis descriptivo de los datos (EDA)
  - 2.1. Análisis de variables categóricas .
  - 2.2. Análisis de variables numéricas.
  - 2.3. Feature engineering: Transformación de datos y generación de indicadores
  - 2.4. Feature selection: Selección de ranking y atributos
  - 2.5. Dataframe a utilizar en data mining processes
3. Modelos de Minería de Datos.
  - 3.1. Testeo y resultados.
  - 3.2. Comparación de métricas de performance.
  - 3.3. K-fold cross validation: validación cruzada de modelos
4. Elección de modelo.
5. Conclusiones y discusión.
6. Anexos y código pyp.

## **1. Descripción General del proyecto**

### **1.1. Descripción del contexto del proyecto**

Considere el problema que enfrenta una entidad financiera que posee altas tasas de fuga voluntaria.

Esta institución no tiene claro cuál es el perfil característico que tienen los clientes fugitivos ni cuáles son las razones por las cuales estos se fugan.

Dentro de este contexto, le han encomendado a usted dos tareas fundamentales. La primera tarea consiste en desarrollar un modelo predictivo que permita identificar tempranamente cuáles clientes son más propensos a fugarse. La segunda tarea consiste en definir una serie de políticas comerciales que permitan retener a estos potenciales clientes fugitivos.

### **1.2. Descripción del Informe**

Este informe tiene por fin, mostrar la metodología utilizada, resultados obtenidos y propuestas de negocios basadas en el análisis y resultados obtenidos, con el propósito de ayudar a la entidad financiera que solicitó la ayuda con el proyecto.

El proyecto involucra utilizar modelos de analítica avanzada para generar un entendimiento del fenómeno de fuga de clientes junto con modelos predictivos los cuales permiten definir con una alta precisión y exactitud el grupo de clientes propensos a perder relación contractual con la compañía bancaria. Particularmente, en la sección 2 se presenta un análisis estadístico de la muestra junto con cada variable relacionado el proceso de feature engineering, en la sección 3 se plantean las transformaciones necesarias para codificar el dataframe a utilizar en los modelos posteriores, para finalmente en la sección 4 y 5 se presentan los diferentes modelos de machine learning su evaluación, testing y optimización de la opción escogida para representar el fenómeno.

Para finalizar el informe concluye con 5 acciones comerciales las cuales están relacionadas directamente con planes de retención, ventas, operaciones y marketing para amortizar el cálculo de clientes fugados que resultan del modelo predictivo.

### **1.3. Objetivos Generales**

En primer lugar, lograr generar un modelo predictivo lo suficientemente robusto para clasificar de forma acertada y precisión a clientes con potencialidad de fuga elevada.

En segundo lugar, identificar las variables más importantes en la clasificación de clientes con alta propensión a la fuga, para poder perfilarlos de forma correcta y tomar acciones preventivas.

### **1.4. Objetivos Específicos**

Desarrollar un modelo de decisión que contemple el costo/beneficio de realizar una acción comercial sobre un candidato a fuga, o bien dejar ir un cliente de la cartera. De manera de maximizar la utilidad general de la compañía e impactar al negocio de manera positiva a través del campo de la data science y decisión making.

## 2. Análisis estadístico de los datos EDA

A continuación realizaremos estadístico descriptivo para cada una de las variables dentro de la data proporcionada, con el propósito de ver distribuciones y su relación con nuestra variable objetivo (FUGA).

Antes de iniciar con el análisis individual de cada una de las variables, podemos revisar algunas características generales de la base de entrenamiento. Contamos con 2,294 registros y 19 columnas.

Respecto a los valores nulos o perdidos de la base, vemos que para la mayoría de las variables esto es muy bajo, por lo que no debería ser un factor relevante a la hora de la selección de variables. Sin embargo, para las variables de Deudas Mensuales, si tenemos bastantes valores 0, que podrían ser nulos o más bien por el tipo de negocio asumir que ese mes el cliente no registro deuda.

Para el análisis estadístico de los datos se realiza un diagnóstico de la base en bruto (raw dataset) con el packages de pandas\_profiling junto con paquetes de visualización seaborn y matplotlib, los cuales se presentan a continuación.

### Diagnóstico general del data frame entregado

Diagnóstico por tipo de variables definidas en el dataframe\_training. En base a esto se separan en variables numéricas las cuales son type: integer/float y por otro lado las variables categóricas las cuales son type: object. Estas últimas tienen que ser transformadas mediante encoders para su manipulación y normalización hacia un dataframe absolutamente numérico. Si bien esta técnica de parametrización es bastante utilizada al momento de tener variables categóricas, es posible hacer análisis de texto de las variables, lo cual no es el scope del proyecto.

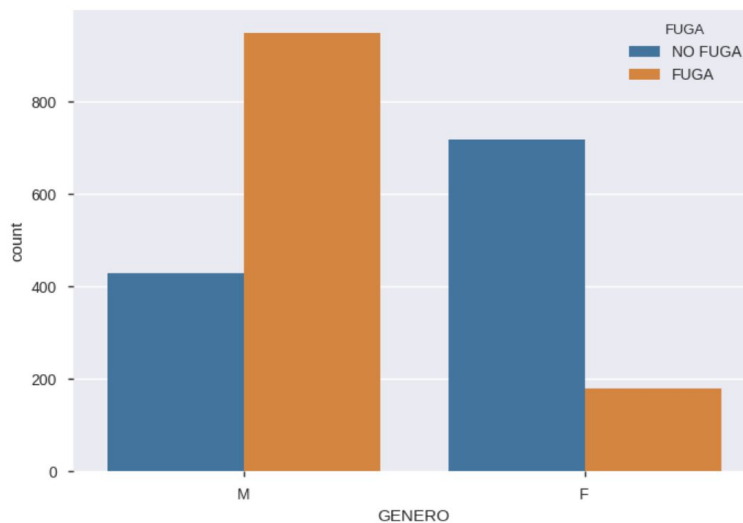
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2287 entries, 1 to 2294
Data columns (total 19 columns):
 GENERO      2276 non-null object
 RENTA       2287 non-null int64
 EDAD        2287 non-null float64
 NIV_EDUC    2276 non-null object
 E_CIVIL      2276 non-null object
 COD_OFI     2287 non-null int64
 COD_COM     2284 non-null float64
 CIUDAD      2284 non-null object
 D_Marzo     2287 non-null int64
 D_Abril     2287 non-null int64
 D_Mayo      2287 non-null int64
 D_Junio     2287 non-null int64
 D_Julio     2287 non-null int64
 D_Agosto    2287 non-null int64
 D_Septiembre 2287 non-null int64
 M_MOROSO    2287 non-null int64
 MONTO       2287 non-null int64
 SEGURO      2287 non-null object
 FUGA        2287 non-null object
dtypes: float64(2), int64(11), object(6)
memory usage: 357.3+ KB
```

A continuación se procede a explorar las variables numéricas y categóricas con foco en la interpretación e importancia en el modelo de machine learning que se verá en la fase de data mining.

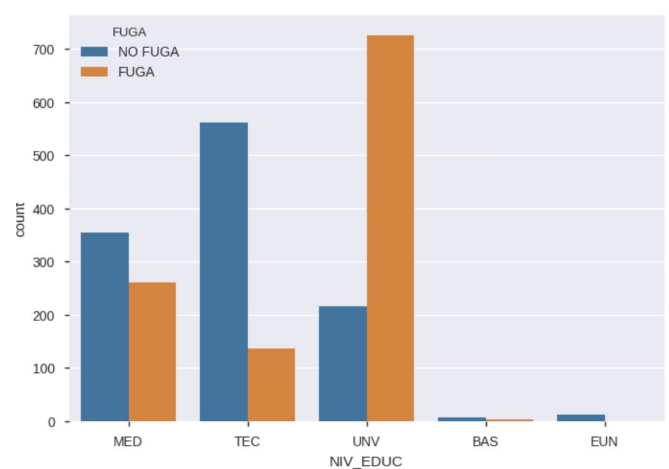
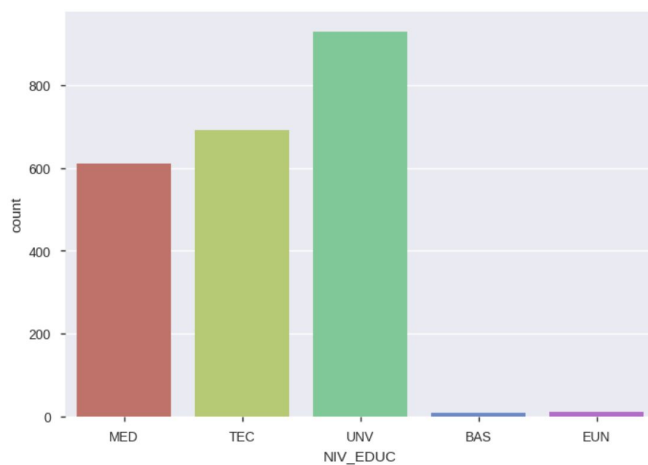
Data analytics certificate MIT - Universidad de Chile

## 2.1. EDA Variables categóricas.

- **GÉNERO:** Para la variable Género, tenemos una proporción 60/40 para Hombres/Mujeres respectivamente, con solo 11 valores perdidos. Si analizamos su distribución para FUGA podemos ver claras diferencias entre ambas clases.

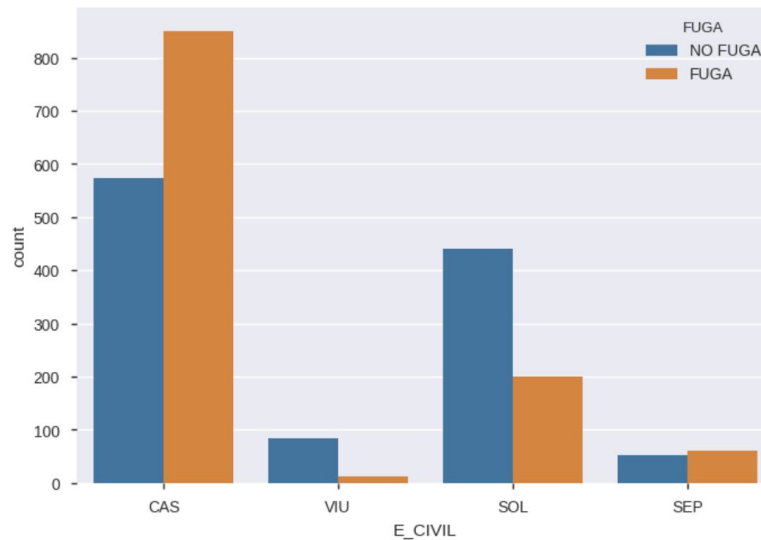


- **NIV\_EDUC:** Para el nivel educacional tenemos 3 clases predominantes que representan el 98% de los datos. Además solo tenemos un 0.5% de valores nulos. Revisando proporciones para FUGA vemos diferencias entre segmentos. Por todo lo anterior dejamos la variable, ya que podría ser buena a la hora de predecir.

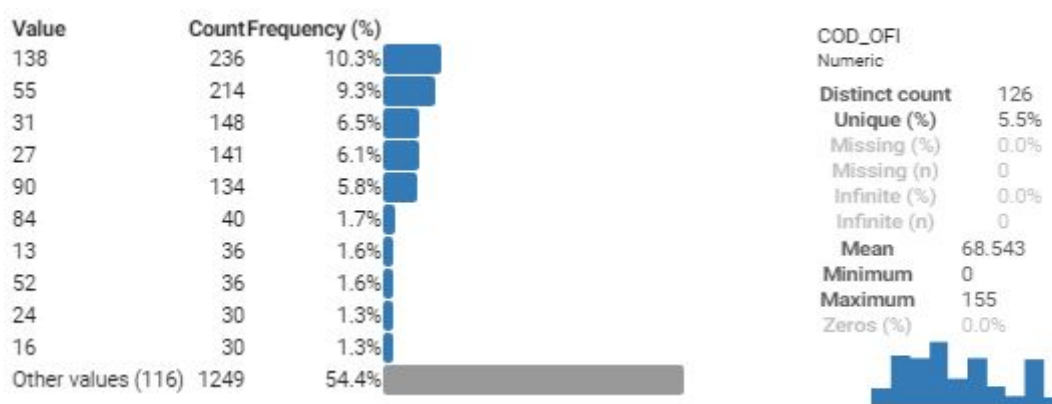


Data analytics certificate MIT - Universidad de Chile

- **E\_CIVIL:** Para Estado Civil podemos ver que hay diferencias entre la proporcionalidad de fuga para cada estado civil. Además, podríamos suponer que si la persona está actualmente sola (viuda/soltera) la tendencia es la no fuga, en cambio para los casados, esta proporción se invierte. Mantenemos esta variable, ya que podríamos transformarla con un “encoding”.

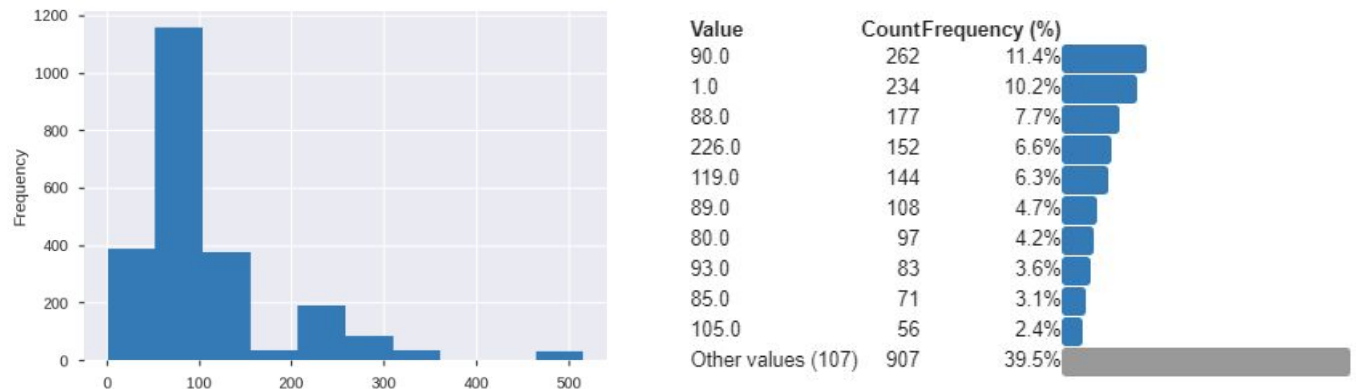


- **COD\_OFI:** Si bien esta variable tiene un formato numérico, se entiende que esta es una variable categoría, ya que hace referencia al código de oficina/sucursal, por lo que hay que tratarla como una variable categórica. Dada la alta cantidad de valores únicos registrados (alta dispersión), preferimos no considerar esta variable para el análisis ya que un “dummy encoding” entregaría muchas variables nuevas, además suponemos que no tendrían un buen desempeño predictivo.

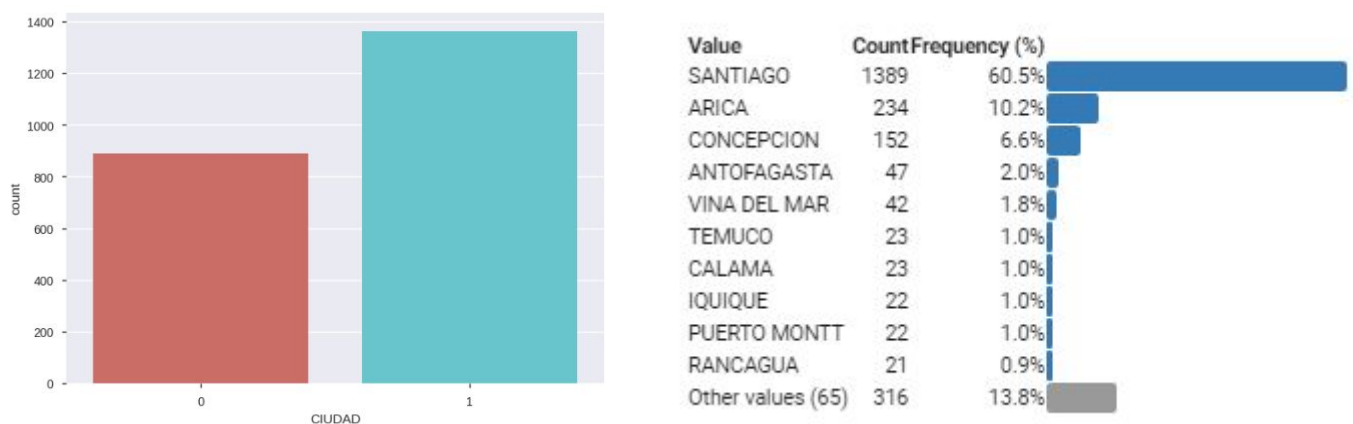


### Data analytics certificate MIT - Universidad de Chile

- **COD\_COM:** Al igual que con la variable de COD\_OFI para código de comuna vemos el mismo patrón los cuales se centran su mayoría en la región metropolitana, y basta con utilizar la variable CIUDAD para considerar la segmentación geográfica dentro del modelo. Es por esto que también es un candidato a no ser utilizada por su condición categórica y subclase de variable CIUDAD.

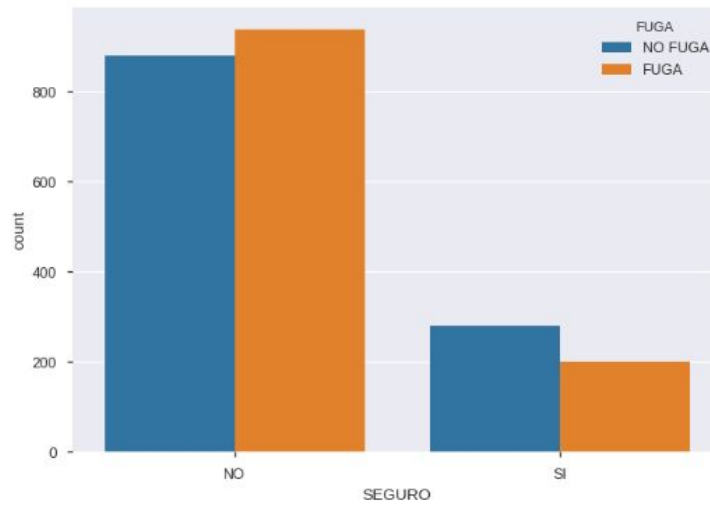


- **CIUDAD:** Para la variable categórica de ciudad, la siguiente concentración relacionada con el 60% en la región metropolitana versus el 40% ubicado en otras regiones. Las distribuciones son relativamente simétricas por lo que se conserva el encoding de ciudad\_stgo: 1 y ciudades\_no\_stgo: 0

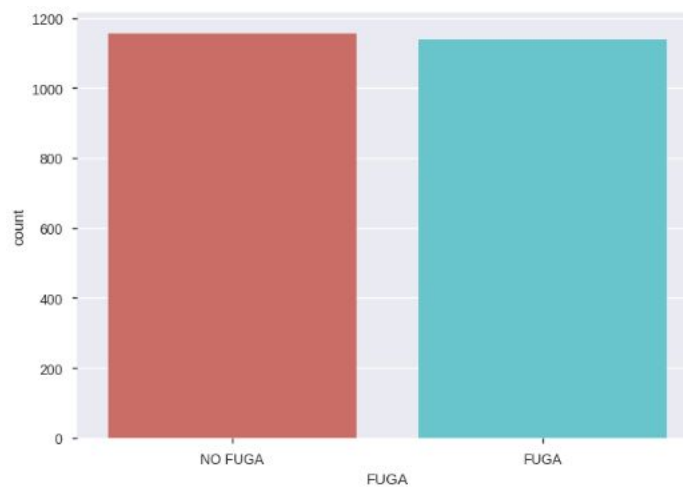


**Data analytics certificate MIT - Universidad de Chile**

- **SEGURO:** Al revisar las proporciones para FUGA sobre SEGURO, vemos que hay proporciones inversas, pero tampoco con una magnitud tan significativa, podríamos suponer que clientes con SEGURO (SI) tienen una propensión menor a la fuga que clientes sin seguro, lo cual podría tener sentido con la realidad. Mantendremos esta variable para el modelo.



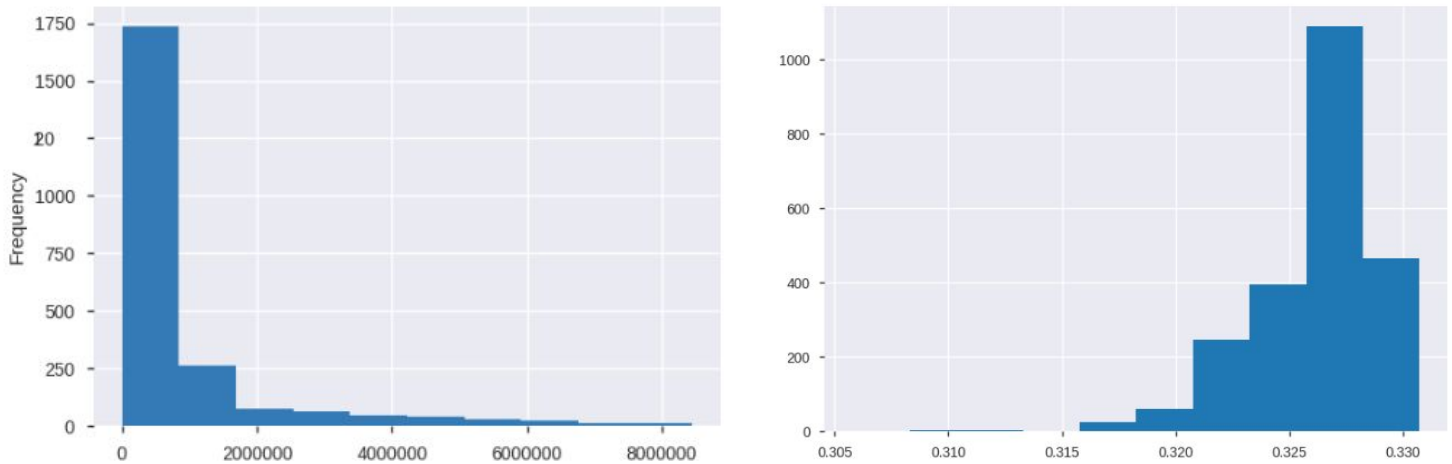
- **FUGA:** Esta es nuestra variable objetivo (Y) que queremos predecir. El primer paso es revisar si esta variable está lo suficientemente balanceado, ya que en caso contrario deberíamos usar algún algoritmo como "smote" para balancear ambas clases y que el algoritmo **no** tienda a predecir la clase predominante.



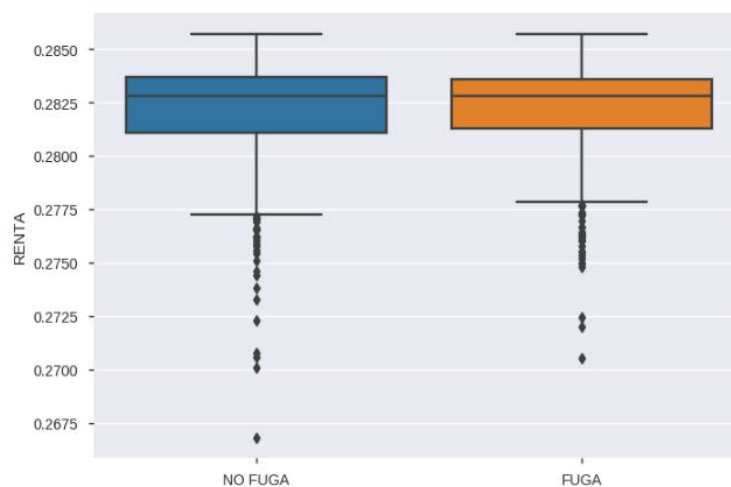


## 2.2. EDA Variables Numéricas

- **RENTA:** Al analizar la variable RENTA esta presenta una alta kurtosis y skewness positiva centrado en ingresos menores a 20.000.000 USD (Costado izquierdo de la imagen). Esto tiende a normalizarse con transformación "log" dentro de la etapa de preprocesamiento del data frame (Costado derecho de la imagen)

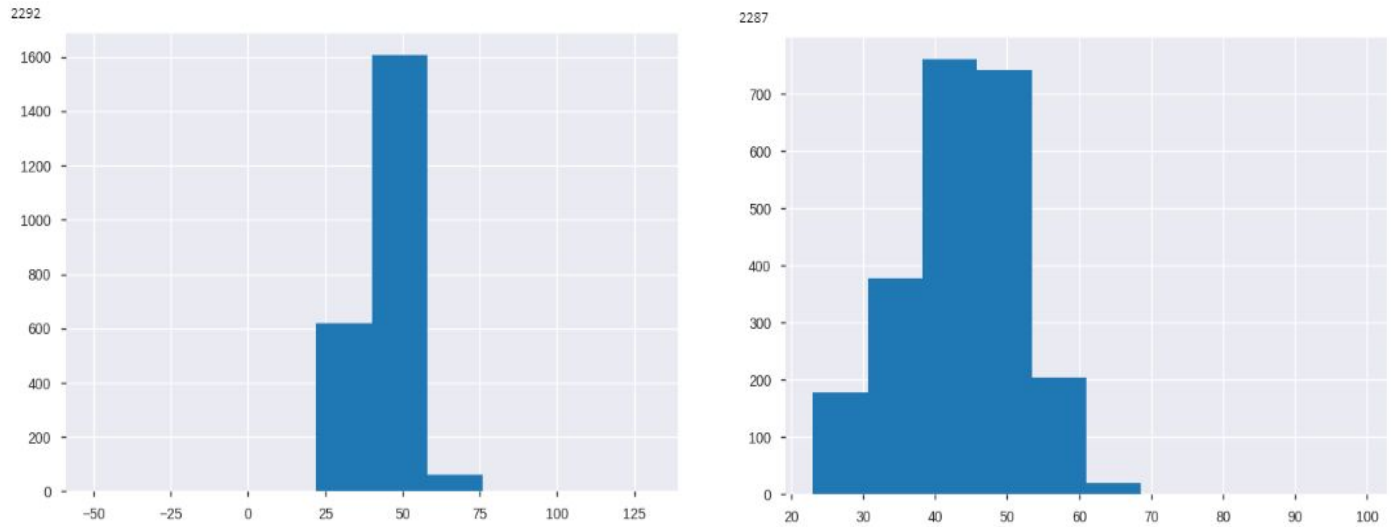


La variable RENTA no presenta un peso significativo en las variable FUGA. Por lo tanto se infiere que no necesariamente a mayor renta implica una mayor tasa de fuga. Al no generar una dependencia de la renta versus la fuga estaríamos frente a un evento no determinista considerando solo esas dos variables.

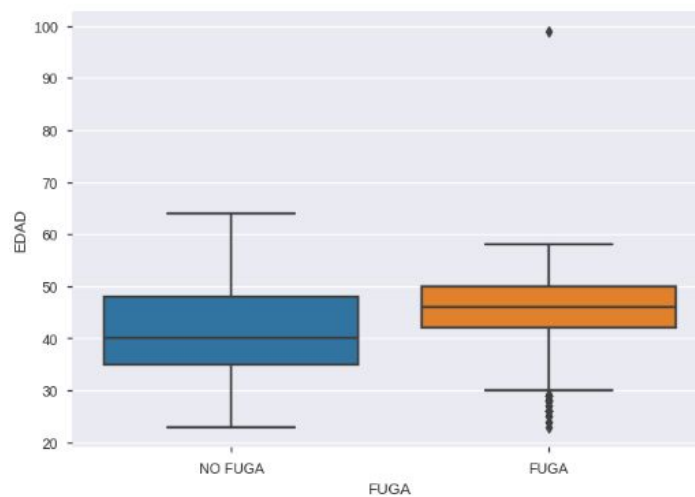


Data analytics certificate MIT - Universidad de Chile

- **EDAD:** Para el histograma de EDAD, encontramos que en la base hay clientes con edades menores a 0, lo cual es incorrecto, además hay clientes con edades sobre los 100 años. Para efectos del análisis, consideraremos que los clientes deben tener más de 18 años y menores a 100 años. El resultado de este filtro es el histograma de la derecha. Ahora podemos apreciar con mayor claridad el rango de edad de los clientes.

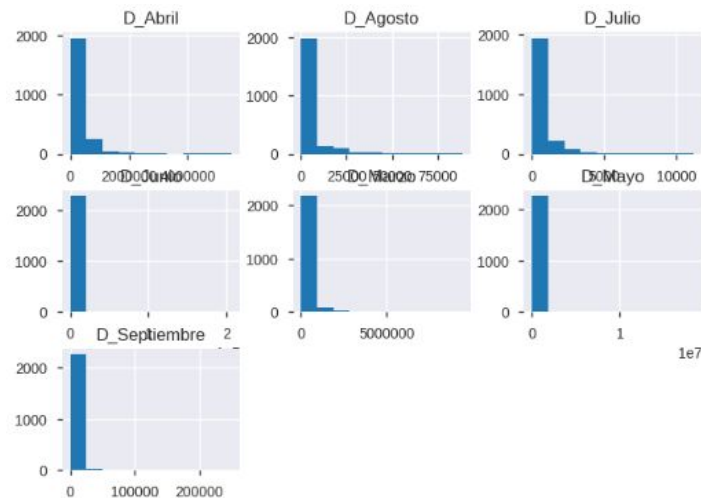


También analizamos a través de Boxplots si hay diferencias para FUGA sobre EDAD, el gráfico a continuación nos indica que para los clientes **FUGA** tienen en promedio una EDAD mayor a los clientes **NO FUGA**.

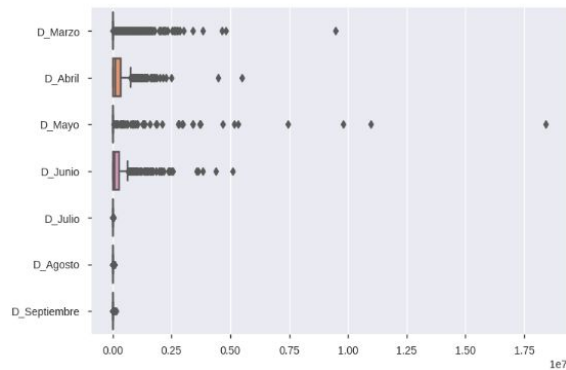


Data analytics certificate MIT - Universidad de Chile

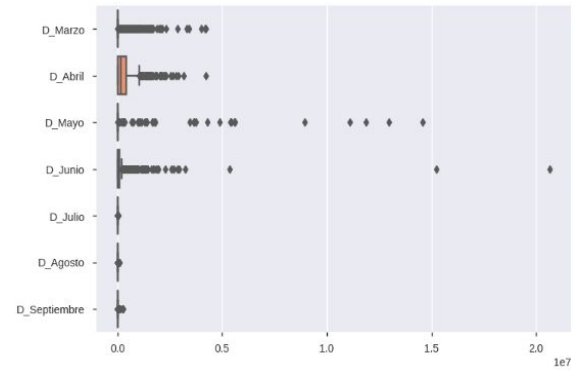
- **DEUDAS:** En el dataframe tenemos 7 columnas, con la información mensual de deuda para cada cliente. Analizando sus histogramas notamos que las escalas no son las mismas para todos los meses, para Marzo, Abril, Mayo, Junio y Julio tenemos una escala entre 0 y  $1e7$ , mientras que para los meses de Agosto, Septiembre y Octubre, la escala va de 0 a  $1e4$ . Además si revisamos las medias sobre la clase FUGA, tampoco vemos diferencias o tendencias notorias.



```
ax = sns.boxplot(data=df_deudas[df_deudas.FUGA == 'FUGA'], orient='h', palette='Set2')
```



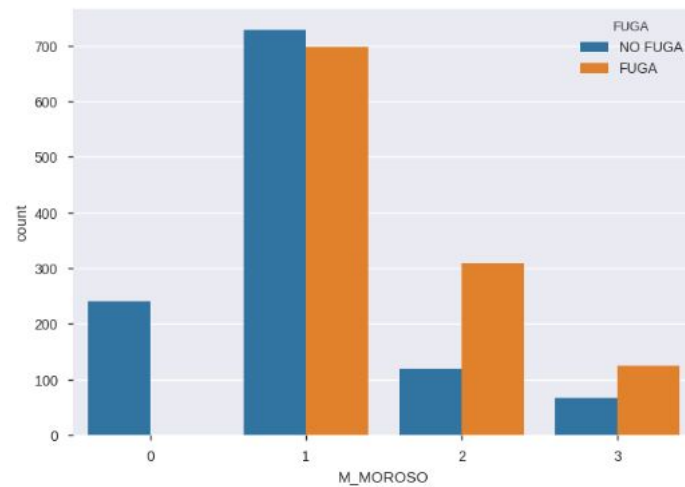
```
ax = sns.boxplot(data=df_deudas[df_deudas.FUGA == 'NO FUGA'], orient='h', palette='Set2')
```



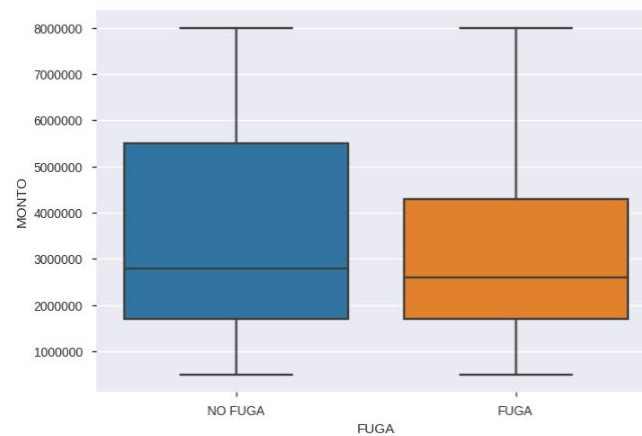
Dado el análisis descriptivo realizado, suponemos que estas variables de forma individual no aportarían valor al modelo, es posible que después de realizar una normalización y combinación de estas para crear nuevas variables ocurra una mejora al modelo.

Data analytics certificate MIT - Universidad de Chile

- **M\_MOROSO:** A continuación revisamos cómo aumentaba la proporción de FUGA a medida que el cliente tiene mayor cantidad de meses morosos. El gráfico nos lleva a concluir que a mayor cantidad de meses moroso, mayor es la probabilidad de fuga del cliente.



- **MONTO:** Respecto al monto, analizando su relación con FUGA, podemos ver que los clientes FUGA tienen en promedio montos menores aprobados que los clientes No Fuga. Podemos suponer que hay clientes que se fugan buscando mejores ofertas en la competencia.



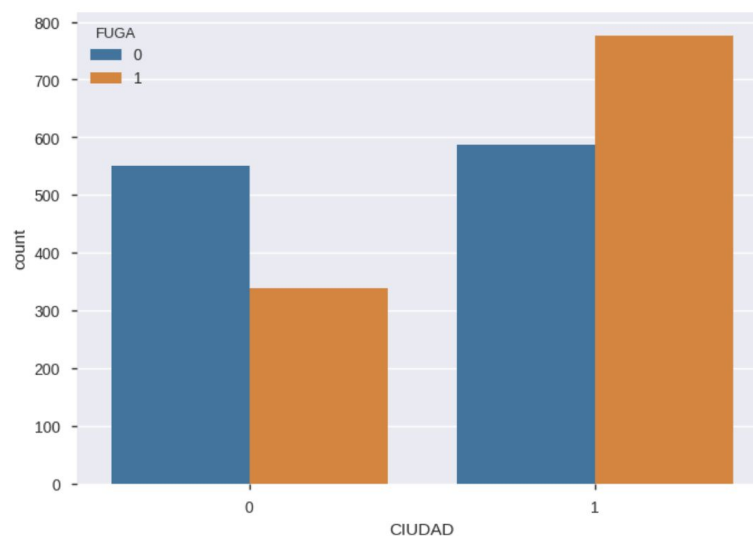
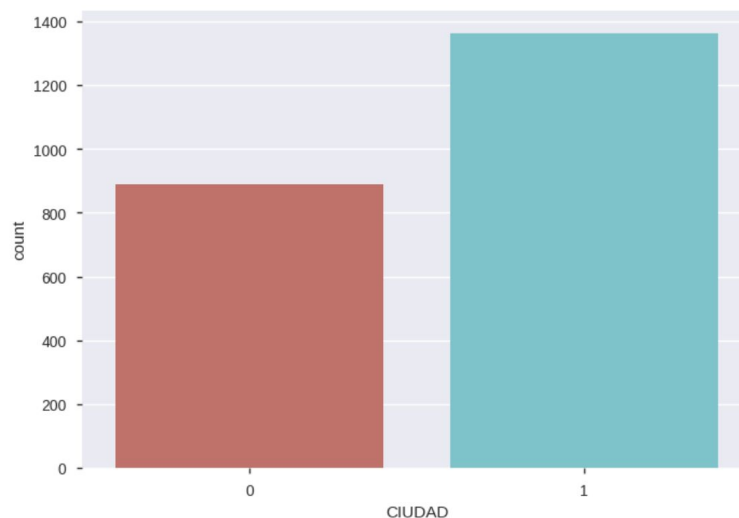
## 2.3. Feature Engineering

**2.3.1. Eliminación de variables:** Eliminaremos las variables COD\_OFI y COD\_COM del data frame, ya que son variables categóricas con una alta dimensionalidad. Para poder usarlas deberíamos crear un diccionario que nos ayude a agruparlas por regiones, por ejemplo. Pero como veremos a continuación, usaremos la variable Ciudad para hacer una discriminación por localización geográfica.

para tener alguna utilidad deberíamos hacer un agrupamiento para ambas con un diccionario que sirva de traductor.

### 2.3.2. Encoding:

- **CIUDAD:** Para la variable Ciudad, notamos que está altamente concentrada para Santiago, más de un 60% de los datos pertenecen a esta categoría, el resto de las clases no tiene un peso significativo, además de estar dispersa en más de 70 otras opciones. Finalmente el encoding hace 1 a los registros que son 'SANTIAGO' y 0 al resto. Luego, revisando las proporciones de FUGA sobre CIUDAD, notamos que la probabilidad de que un cliente se fugue, es mayor si vive en Santiago.



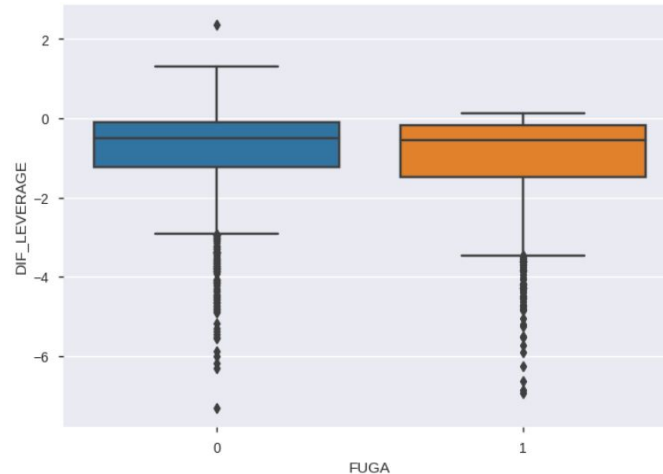
### Data analytics certificate MIT - Universidad de Chile

- Para las variables **E\_CIVIL**, **GENERO**, **SEGURO** y **FUGA** también se realizó un encoding 1 0, como se muestra en el código a continuación:

```
df_training['E_CIVIL'] = np.where(df_training['E_CIVIL'].str.contains('CAS'), 1, 0)
df_training['GENERO'] = np.where(df_training['GENERO'].str.contains('M'), 1, 0)
df_training['SEGURO'] = np.where(df_training['SEGURO'].str.contains('SI'), 1, 0)
df_training['FUGA'] = np.where(df_training['FUGA'].str.contains('NO FUGA'), 0, 1)
```

**2.3.3. Creación de Variables:** Con el objetivo de obtener mayor representatividad de las variables dependientes sobre la predictiva, se han creado variables e indicadores que permitirán agregar mayor peso a los coeficientes que explicarán y le darán robusticidad al modelo de ML.

- **Leverage1** : variable conformada por MONTO\_DEUDA de [Julio, Agosto, Septiembre]
- **Leverage 2:** variable conformada por MONTO\_DEUDA de [Abril, Mayo, Junio]
- **Diferencia leverage:** variable representada por la diferencia entre leverage 1 y leverage 2 sobre la renta de los periodos, mediante la siguiente fórmula:

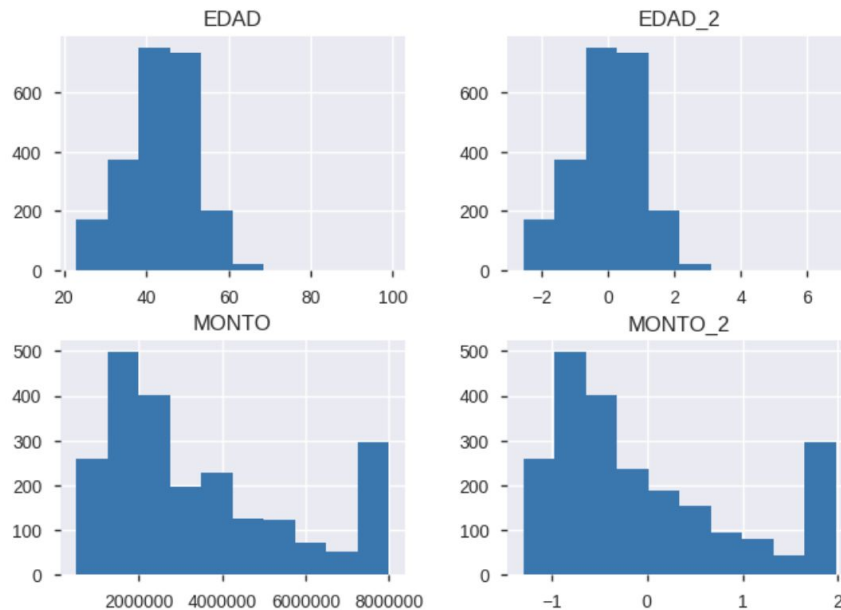


La variable diferencia leverage no indica mayor diferencia para casos fugados y no fugados.

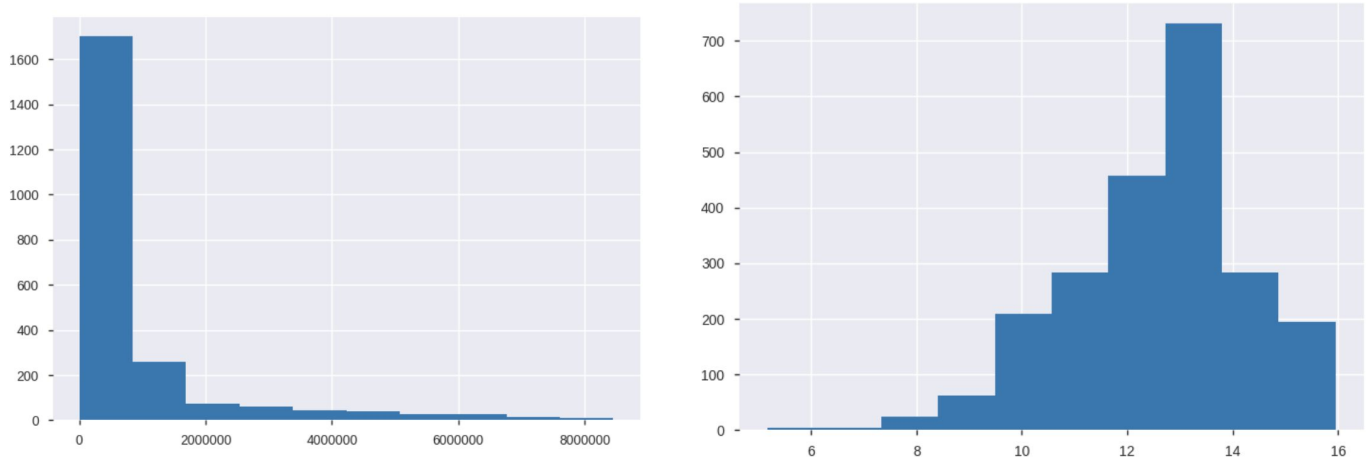
Data analytics certificate MIT - Universidad de Chile

**2.3.4. Escalamiento:** Algunos de los algoritmos usados en la parte de minería de datos requieren que las variables numéricas están escaladas. Es por esto que aplicamos esta transformación a las variables EDAD y MONTO.

La metodología a usar para el escalamiento fue z-score  $((x - u) / s)$ , tanto las medias y las desviaciones estándar de cada variable fueron guardadas en variables independientes, para poder ser aplicadas posteriormente en la transformación de estas mismas variables en la base de validación.



**2.3.5. Transformación Log:** La variable RENTA, como revisamos anteriormente en el EDA, tenía una asimetría pronunciada hacia la derecha, decidimos aplicar una transformación logarítmica para cambiar la distribución de esta variable.



### 3. Minería de datos

Se realizará una comparación del desempeño de distintos modelos de clasificación utilizando k-fold Cross Validation (k = 10). Se revisarán las métricas de accuracy, sensitivity (recall) y AUC. Con ello se definirá un modelo de buen desempeño, el que pasaremos a optimizar, a través de un grid search de sus hiper-parámetros, para elegir el modelo campeón.

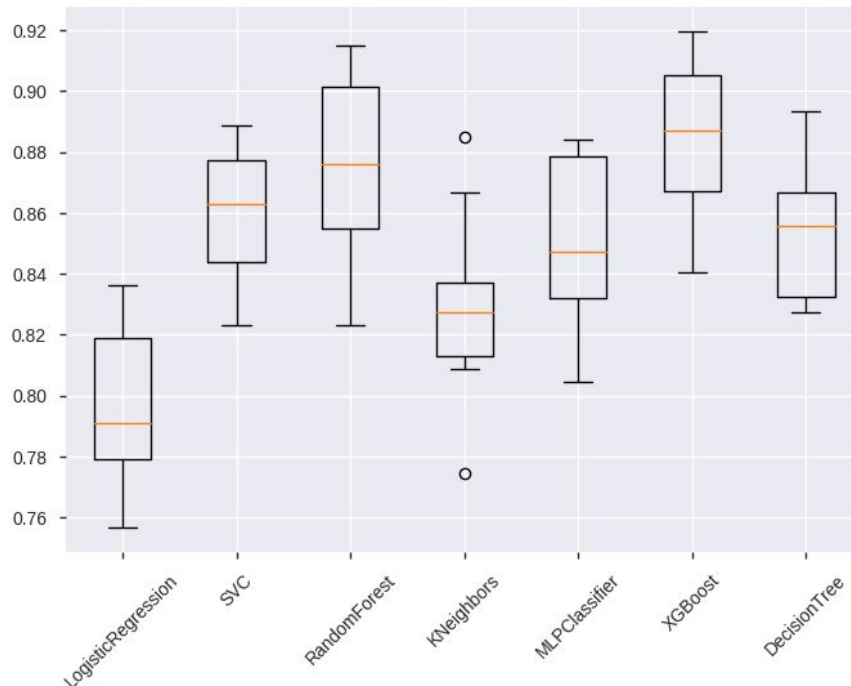
Los modelos a comparar se eligieron entre los más comunes para problemas de clasificación binomial.

- Regresión Logística
- Support Vector Machines - Classification (SVC)
- Random Forest
- K - Nearest Neighbors (KNN)
- Redes Neuronales (MLP)
- Extreme Gradient Boosting (XGBoost)
- Árboles de Decisión

En la img. x se muestra el resultado del desempeño de los modelos respecto del accuracy .

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

donde se observa que en promedio XGBoost se desempeña por sobre los demás, y con un grado de dispersión tolerable. Alcanzando el 88% de exactitud.



Img x. Comparación del desempeño de modelos por accuracy.

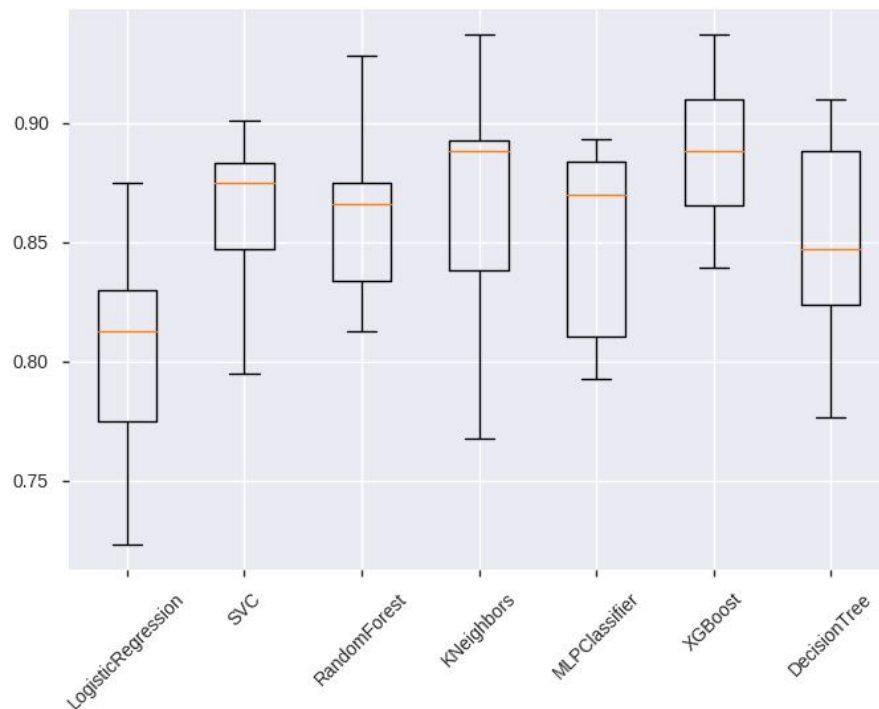
Para revisar cómo los modelos capturan la variable de fuga, se evalúa cómo se comporta la métrica de recall en los distintos modelos. Con ello se espera revisar, qué tan sensible.



Data analytics certificate MIT - Universidad de Chile

$$Recall = \frac{TP}{TP + FN}$$

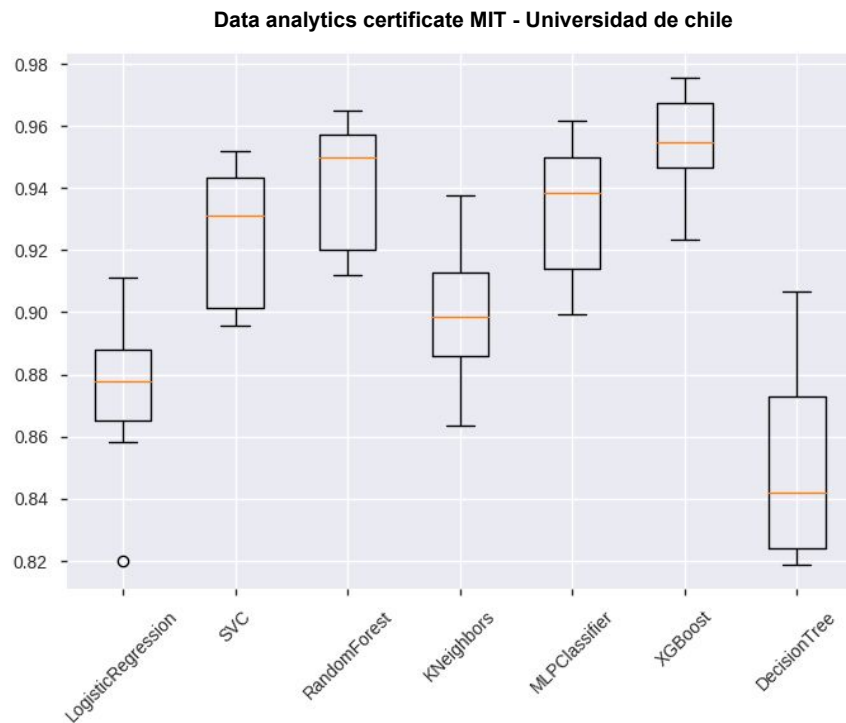
En la imagen X. se observa que XGBoost vuelve a ser el modelo de mejor desempeño con un 88% de recall.



Img X. Desempeño de los modelos según recall

Finalmente la métrica del área bajo la curva ROC (AUC), de manera de poder evaluar el desempeño general de clasificación de los distintos modelos.

En la imagen X. vemos que, consistentemente con los resultados anteriores. XGBoost derrota en su capacidad de clasificación, alcanzando un 0.95 de AUC.



Img. Desempeño de los modelos por AUC

## 7. Elección de modelo.

Dados los resultados anteriores, se elige Extreme Gradient Boosting, como modelo para realizar una búsqueda optimizando el parámetro del AUC. La elección de los hiper-parámetros tiene que ver con la lógica de tener parámetros que mueven bastante los resultados, de manera de encontrar una región sub-óptima, desde la cual si alguien quisiera seguir profundizando pudiese tener como base para empezar desde un buen modelo.

### 7.1. Grid Search.

Dividiremos los parámetros del grid en tres tipos, que reflejan, a grandes rasgos, 3 dimensiones que se pueden optimizar de extreme gradient boosting, notar entre corchetes cuadrados los valores elegidos para la búsqueda.

- A. Hiper-parámetros del Ensamble:
  - a. Learning rate: Velocidad o salto de reducción del error **[0.01, 0.05, 0.1]**
  - b. N Estimators: Número de árboles construidos por XGBoost **[50, 100, 200]**
- B. Hiper-parámetros de los Árboles de Decisión:
  - a. Max depth: Máxima profundidad de los árboles construidos por XGBoost **[4, 5, 6]**
  - b. Colsample by tree: Número de features usados por cada árbol **[0.3, 0.5, 0.7]**
  - c. Min Child Weight: Mínimo peso admitido en una hoja para abrir en ramas, de no cumplir entonces poda. **[1, 5, 10]**
- C. Hiper-parámetros de Regularización: **[0, 0.5, 1]**
  - a. Gamma:
  - b. alpha: Regularización L1

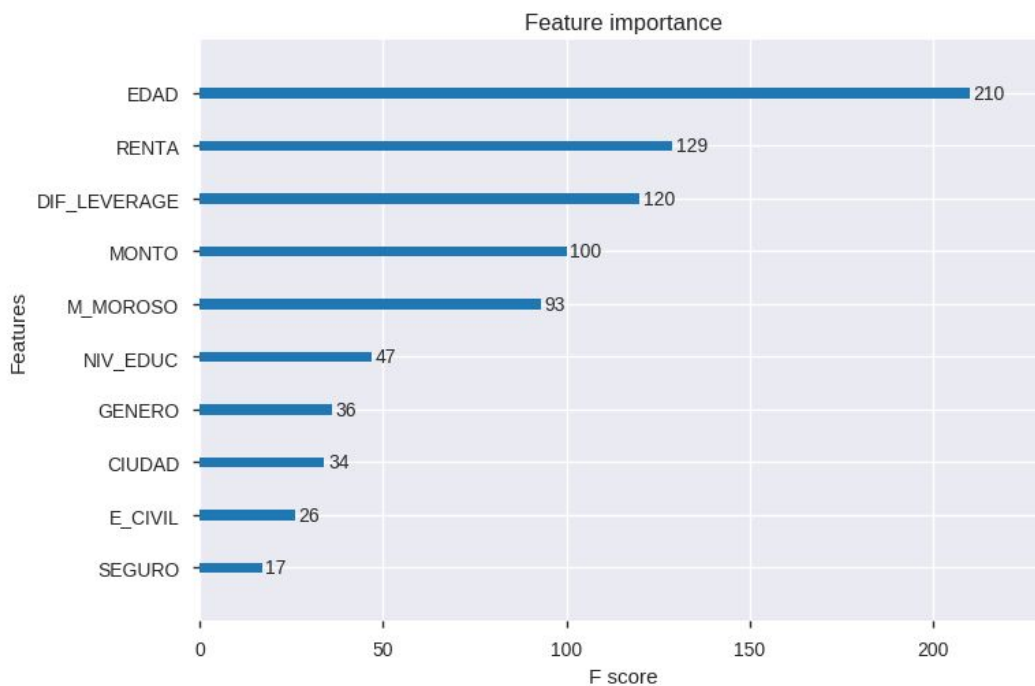
Data analytics certificate MIT - Universidad de Chile

c. lambda: Regularización L2

Obteniéndose el modelo ganador. Con las siguientes métricas de desempeño de clasificación (1 es clase Fuga):

precision	recall	f1-score	support		
	0	0.86	0.92	0.89	321
	1	0.92	0.87	0.89	356
accuracy				0.89	677
macro avg		0.89	0.89	0.89	677
weighted avg		0.89	0.89	0.89	677

Y con la siguiente lista de importancia de atributos.



## 7.2. Punto de Corte.

Con el modelo anterior tenemos una probabilidad de fuga bastante creíble. En esta etapa trataremos de encontrar el punto de corte  $c$  óptimo para la probabilidad de fuga  $P(x)$ , tal que si se tiene una función de utilidad  $U$ : dependa de esa probabilidad y de un punto de corte, y entregue dinero. Notar que para esta tarea se está asumiendo que tenemos una capacidad de retención del 100% de los clientes, cosa que es una simplificación de la realidad. Donde se duele operar en base a presupuestos financieros.

En particular se sabe que la matriz de costo/beneficio  $B$  tiene la siguiente estructura.

Data analytics certificate MIT - Universidad de Chile

		Real	
		Fuga	
Predicción	N	0	-5.000
	S	-100	1.000

Y la matriz de confusión  $M(x, f(x), y', c)$ , que depende de un vector de atributos  $x$ , la predicción  $f(x)$ , los labels de testeo  $y'$  y un cierto punto de corte  $c$  de probabilidad de fuga. En general  $M$  tiene la siguiente estructura:

		Real	
		Fuga	
Predicción	N	TN	FN
	S	FP	TP

Luego para el caso particular que se está estudiando la función de utilidad estaría dada por el producto escalar (punto o interno) de estas dos matrices.

$$U(M, B) = M \cdot B$$

O sea evaluando  $B$

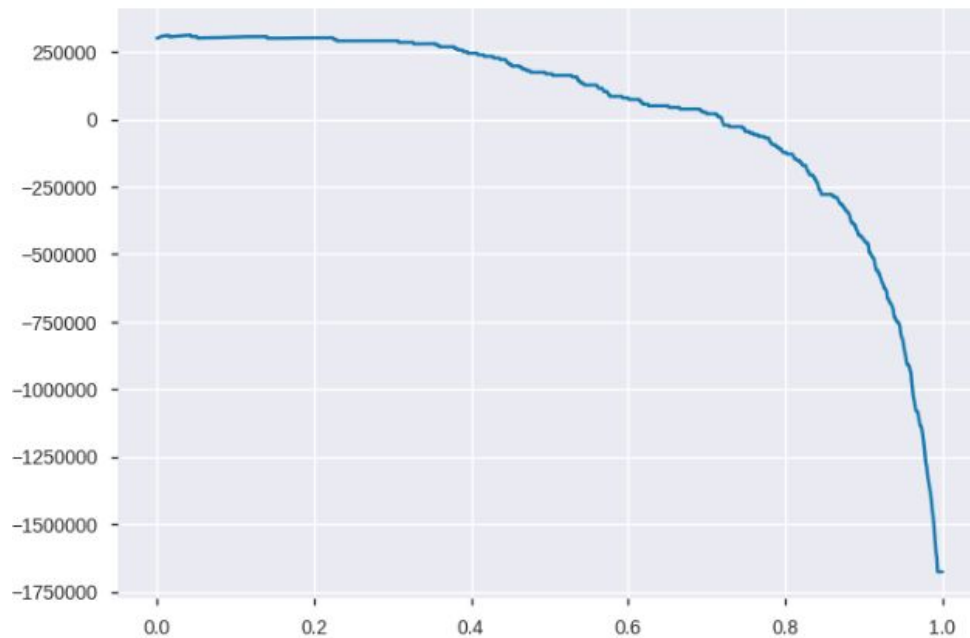
$$U(M, B) = 0 \cdot TN - 100 \cdot FN - 5.000 \cdot FP + 1.000 \cdot TP$$

Luego maximizando la utilidad se obtiene que

$c = 4.29\%$ , el que implica que  $U_{\max} = \$311.900$  para el modelo  $f(x) = y'$

Viendo gráficamente la función de utilidad dado el punto de corte, podemos ver que si no se toman acciones al menos en el 30% de mayores propensiones a fuga, la empresa pierde dinero.

Data analytics certificate MIT - Universidad de Chile



Sabemos que existe otro approach para resolver este problema, y sería realizar el grid search de XGBoost utilizando el recall como métrica objetivo en lugar del AUC. Sin embargo creemos en base a lo señalado sobre todo en el curso de Data Driven Decision Making, que es muy importante separar los objetivos de un proyecto de analytics. Es decir, cuando se quiere buscar el modelo de fuga, uno desea que caracterice bien al fenómeno, cuanto me cuesta a mi una campaña no debería ser un input para ese modelo, es decir se debe encargar de entregar una confiable probabilidad de fuga.

Además, si se mezclan los objetivos, añadiendo a la búsqueda del modelo la información de costo-beneficio, o presupuesto, o capacity de gestión de un contact center (por ejemplo). Y alguno de estas restricciones al problema cambia, entonces el modelo ya no sirve. En cambio si se separa, sólo habría que re-calibrar el punto de corte. Es decir, para el problema particular que nos convoca, sólo habría que modificar B.

## 9. Conclusión y discusión.

Con todo el trabajo anterior, estamos en condiciones de proponer acciones comerciales para revertir la situación de fuga de la compañía. Como se obtuvo que se justifica económicamente realizar acciones proactivas de prevención hasta un punto del corte  $c = 4.29\%$  de probabilidad de fuga. Entonces dividiremos por motivos comerciales en 2 grupos:

- Acciones de Retención para  $P(y = F) > 50\%$ .
  - Acciones de Fidelización para  $P(y = F)$  entre  $c$  y  $50\%$
- **Retención 1: Refinanciamiento/retención de préstamos vigentes con tasa preferencial.**  
Para clientes en segmentos con deuda alta, oferta alta y de buen perfil que tengan una probabilidad de Fuga mayor al  $50\%$ .
  - **Retención 2: Reprogramación de créditos Morosos**  
Para clientes en segmentos con deuda, oferta baja y de mal perfil que tengan una probabilidad de Fuga mayor al  $50\%$ .

**Data analytics certificate MIT - Universidad de Chile**

- **Retención 3: Aceleración de flujo de crédito oferta.**  
Para clientes en segmentos con deuda baja, de buen perfil que tengan una probabilidad de Fuga mayor al 50%.
- **Fidelización 1: Cuenta corriente gratuita con PAC y/o Abono de remuneraciones.**  
Para clientes de edad media y media-alta, deuda media y media baja que tengan una probabilidad de fuga mayor a c.
- **Fidelización 2: Premio en KM FEN\_MIT**  
Para clientes de edad baja, deuda baja que tengan una probabilidad de fuga mayor a c.

Con este framework de acciones podemos

**A. Bibliografía**

- a. [Scikit-learn: Machine Learning in Python](#), Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011
- b. Chen, Tianqi & Guestrin, Carlos. (2016). XGBoost: A Scalable Tree Boosting System. 785-794. 10.1145/2939672.2939785.
- c. Travis E. Oliphant. **Python for Scientific Computing**, Computing in Science & Engineering, **9**, 10-20 (2007), DOI:10.1109/MCSE.2007.58 ([publisher link](#))
- d. K. Jarrod Millman and Michael Aivazis. **Python for Scientists and Engineers**, Computing in Science & Engineering, **13**, 9-12 (2011), DOI:10.1109/MCSE.2011.36 ([publisher link](#))
- e. Travis E. Oliphant. **A guide to NumPy**, USA: Trelgol Publishing, (2006).
- f. Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. **The NumPy Array: A Structure for Efficient Numerical Computation**, Computing in Science & Engineering, **13**, 22-30 (2011), DOI:10.1109/MCSE.2011.37 ([publisher link](#))
- g. John D. Hunter. **Matplotlib: A 2D Graphics Environment**, Computing in Science & Engineering, **9**, 90-95 (2007), DOI:10.1109/MCSE.2007.55 ([publisher link](#))
- h. Wes McKinney. **Data Structures for Statistical Computing in Python**, Proceedings of the 9th Python in Science Conference, 51-56 (2010) ([publisher link](#))
- i. Wright, M.H. (1996), "Direct Search Methods: Once Scorned, Now Respectable", in Numerical Analysis 1995, Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis, D.F. Griffiths and G.A. Watson (Eds.), Addison Wesley Longman, Harlow, UK, pp. 191-208.